Trabajo Fin de Grado
Grado en Ingeniería Aeroespacial

Fundamentos de programación de VBA en Catia V5: Aplicación al diseño de Rodamientos

Autor: José Miguel Peña Rodríguez
Tutor: Cristina Torrecillas Lozano

Dep. de Ingeniería Gráfica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2015

Proyecto Fin de Grado
Grado en Ingeniería Aeroespacial

Fundamentos de programación de VBA en Catia V5: Aplicación al diseño de Rodamientos

Autor:

José Miguel Peña Rodríguez

Tutor:

Cristina Torrecillas Lozano

Profesor titular

Dep. de Ingeniería Gráfica

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2015

Trabajo Fin de Grado: Fundamentos de programación de VBA en Catia V5: Aplicación al diseño de Rodamientos

Autor: José Miguel Peña Rodríguez

Tutor: Cristina Torrecillas Lozano

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2015

El Secretario del Tribunal

A mi familia y amigos

A mis maestros

Resumen general

El contenido del presente proyecto estudia algunos de los posibles campos de aplicación del lenguaje de macros de *Microsoft Visual Basic for Applications* (VBA) al programa de diseño informático asistido por ordenador CATIA V5.

En primer lugar se realiza una introducción a la teoría de los rodamientos donde se explican los fundamentos generales de los tipos de rodamientos que encontramos hoy día así como sus características principales y posibles aplicaciones. Se definirá aquí la geometría de los rodamientos que será empleada para las programaciones en VBA.

En segundo lugar, se define qué es el VBA, un lenguaje de macros de *Microsoft Visual Basic*, además de explicar cómo se accede desde CATIA V5, el entorno de este, el modo en que el usuario puede desarrollar interfaces para interactuar con el programa así como los objetos más importantes del módulo *Mechanical Design (sketcher, Part Desing y Assembly Desing)* y las herramientas que ofrecen para trabajar.

Posteriormente, se ha pasado a desarrollar el núcleo principal de este trabajo que es una aplicación para elaborar rodamientos de distintos tipos en 3D a partir de los parámetros geométricos encontrados en el catálogo general de la empresa SKF. Se han implementado rodamientos rígidos de bolas, rodamientos de bolas con contacto angular, rodamientos de rodillos cilíndricos, rodamientos de agujas y rodamientos axiales de bolas, de rodillos cilíndricos y de agujas.

Para terminar se ofrece el manual para el usuario a partir del cual, la persona que quiera hacer uso de la aplicación, sepa como interactuar con la interfaz para obtener el rodamiento que necesite.

Objetivos y alcance

El objetivo del proyecto es la interacción y manejo del lenguaje de programación de macros VBA para desarrollar una aplicación basada en dicha tecnología con la que poder obtener rápidamente en CATIA V5, rodamientos catalogados que se empleen en conjuntos modelados con tecnología 3D.

La idea de este trabajo nace a partir de la ausencia de herramientas específicas en CATIA para el diseño de elementos mecánicos comunes como pueden ser engranajes, rodamientos o incluso alas de aviones. Por ello, en este proyecto se va a desarrollar un software para que el usuario pueda obtener una representación gráfica automática de distintos tipos de rodamientos a partir de los parámetros de diseño requeridos que se pueden encontrar en el catálogo de rodamientos de la empresa SKF.

Así, el alcance de este trabajo de fin de grado en Ingeniería Aeroespacial, es familiarizarse y controlar los conocimientos de programación necesarios para elaborar esta o cualquier otra aplicación y facilitar a cualquier usuario el acceso al software realizado para cumplir sus necesidades. De este modo, el alcance del proyecto no es un estudio intensivo de la teoría de los rodamientos y tampoco la exploración de las posibles aplicaciones de éste.

Índice general

Resumen general ..... - 9 -
Objetivos y alcance ..... - 10 -
Índice general ..... - 11 -
Índice de figuras ..... - 15 -
Índice de tablas ..... - 18 -
Índice de Códigos ..... - 19 -
1. Introducción a la teoría de los rodamientos ..... - 21 -
1.1. Introducción ..... - 21 -
1.2. Breve descripción de la evolución histórica de los rodamientos ..... - 22 -
1.3. Partes de un rodamiento ..... - 24 -
1.4. Clasificación general de los rodamientos ..... - 25 -
1.5. Tipos de rodamientos ..... - 26 -
1.5.1. Rodamientos rígidos de bolas ..... - 26 -
1.5.2. Rodamientos de Magneto ..... - 28 -
1.5.3. Rodamientos de bolas a rótula ..... - 28 -
1.5.4. Rodamientos de bolas con contacto angular ..... - 29 -
1.5.5. Rodamientos de rodillos cilíndricos ..... - 30 -
1.5.6. Rodamientos de agujas ..... - 31 -
1.5.7. Rodamientos de rodillos a rótula ..... - 33 -
1.5.8. Rodamientos de rodillos cónicos ..... - 33 -
1.5.9. Rodamientos axiales de bolas ..... - 34 -
1.5.10. Rodamientos axiales de rodillos y de agujas ..... - 35 -
1.5.11. Rodamientos axiales de rodillos a rótula ..... - 36 -
1.6. Materiales de los rodamientos ..... - 37 -
1.6.1. Materiales usados para los aros y los elementos rodantes de los rodamientos…. ..... - 37 -
1.6.2. Materiales usados para las jaulas ..... - 38 -
1.7. Selección del tipo de rodamiento ..... - 39 -
1.7.1. Espacio disponible ..... - 40 -
1.7.2. Cargas sobre el rodamiento ..... - 41 -
1.7.3. Desalineación angular ..... - 41 -
1.7.4. Límites de velocidad ..... - 41 -

1.7.5. Precisión ..... - 42 -
1.7.6. Rigidez ..... - 42 -
1.7.7. Desplazamiento axial ..... - 42 -
1.8. Aplicaciones de los rodamientos ..... - 42 -
1.9. Descripción de la geometría de los rodamientos que se van a realizar ..... - 44 -
1.9.1. Rodamientos rígidos de bolas ..... - 45 -
1.9.2. Rodamientos de bolas con contacto angular ..... - 45 -
1.9.3. Rodamientos de rodillos cilíndricos ..... - 46 -
1.9.4. Rodamientos de agujas ..... - 47 -
1.9.5. Rodamientos axiales de bolas ..... - 48 -
1.9.6. Rodamientos axiales de rodillos y agujas ..... - 48 -
1.10. Posibles averías ..... - 49 -
2. El entorno de programación VBA en CATIA V5 ..... - 52 -
2.1. CATIA V5 ..... - 52 -
2.2. Visual Basic for Application ..... - 53 -
2.2.1. Entorno ..... - 53 -
2.2.2. Librerías de las macros ..... - 54 -
2.2.3. Macro Recording ..... - 55 -
2.3. Visual Basic Editor ..... - 56 -
2.4. Iniciación a la programación ..... - 59 -
2.4.1. Declaración de estamentos ..... - 60 -
2.4.2. Estamentos ..... - 60 -
2.4.3. Estamentos ejecutables ..... - 60 -
2.4.4. Funciones y subfunciones ..... - 61 -
2.4.5. Estructuras condicionales e iterativas ..... - 61 -
2.4.6. Objetos orientados a la programación ..... - 62 -
2.4.7. Cómo definir un objeto ..... - 63 -
2.5. Interfaz con el usuario ..... - 63 -
3. Herramientas de programación de rodamientos. ..... - 67 -
3.1. Introducción ..... - 67 -
3.2. Arranque ..... - 67 -
3.3. SKETCHER ..... - 69 -
3.3.1. Crear un punto ..... - 71 -

3.3.2. Crear una recta ..... - 71 -
3.3.3. Crear un círculo o arco de circunferencia ..... - 71 -
3.3.4. Crear un spline ..... - 72 -
3.4. Restricciones ..... - 73 -
3.4.1. CatCstTypeDistance ..... - 75 -
3.4.2. CatCstTypeRadius ..... - 75 -
3.4.3. CatCsTypeHorizontally/Vertically ..... - 75 -
3.4.4. CatCstTypeLength ..... - 76 -
3.4.5. CatCstTypeParallelism ..... - 76 -
3.4.6. CatCstTypeOn ..... - 76 -
4. CATPART ..... - 78 -
4.1. PART DESIGN ..... - 78 -
4.1.1. Pad… ..... - 80 -
4.1.2. Pocket ..... - 81 -
4.1.3. Hole… ..... - 82 -
4.1.4. Slot… ..... - 84 -
4.1.5. Shaft.. ..... - 84 -
4.1.6. Rib… ..... - 85 -
4.1.7. CircularPattern ..... - 86 -
5. CATPRODUCT ..... - 88 -
5.1. Assembly Design ..... - 88 -
5.1.1. AddComponentsFromFiles ..... - 89 -
5.1.2. Fix Component ..... - 90 -
5.1.3. Offset Constraint ..... - 90 -
5.1.4. Coincidence Constraint ..... - 91 -
6. Aplicación: entorno de programación. ..... - 92 -
6.1. Ventana emergente para seleccionar la carpeta de guardado ..... - 93 -
6.2. Rodamientos de bolas ..... - 94 -
6.2.1. Rodamientos rígidos de bolas ..... - 94 -
6.2.2. Rodamientos de bolas con contacto angular ..... - 99 -
6.2.3. Rodamientos axiales de bolas ..... - 102 -
6.3. Rodamientos de rodillos cilíndricos ..... - 103 -
6.3.1. Tipos N, NJ, NU y NUP ..... - 103 -

6.3.2. Rodamientos axiales rodillos cilíndricos ..... - 104 -
6.3.3. Corona de rodillos cilíndricos ..... - 105 -
6.4. Rodamientos de agujas ..... - 106 -
6.4.1. Rodamientos de agujas con aro interno ..... - 106 -
6.4.2. Rodamientos de agujas sin aro interno ..... - 107 -
6.4.3. Corona de agujas ..... - 108 -
6.4.4. Rodamientos axiales de agujas ..... - 109 -
7. Aplicación Rodamiento SKF: manual de usuario ..... - 111 -
8. Conclusiones ..... - 113 -
9. Bibliografía ..... - 115 -
Anexo 1: Códigos de programación ..... - 116 -

Índice de figuras

Figura 1. Representación del antiguo Egipto .....- 22 -
Figura 2. Dibujo de Leonardo .....- 23 -
Figura 3. Rodamiento compuesto de diversos materiales. .....- 24 -
Figura 4. Partes de un rodamiento:1…2…. .....- 24 -
Figura 5. Rodamientos rígidos de bola. .....- 27 -
Figura 6. Montaje de un rodamiento de bolas. .....- 27 -
Figura 7. Rodamientos de bolas a rótula. .....- 28 -
Figura 8. Rodamientos de contacto angular de una hilera. .....- 29 -
Figura 9: Rodamiento de bolas con contacto angular de dos hileras. .....- 30 -
Figura 10. Rodamiento de rodillos cilíndricos. .....- 31 -
Figura 11. Tipos de Rodamientos de rodillos cilíndricos. .....- 31 -
Figura 12. Rodamientos de agujas. .....- 32 -
Figura 13. Rodamientos de rodillos a rótula. .....- 33 -
Figura 14. Rodamientos de rodillos cónicos. .....- 34 -
Figura 15. Rodamientos axiales de bolas. .....- 35 -
Figura 16. Rodamientos axiales de rodillos. .....- 36 -
Figura 17. Rodamiento axial de rodillos a rótula. .....- 36 -
Figura 18. Ejemplo de jaula. .....- 39 -
Figura 19. Corona axial de agujas. .....- 40 -
Figura 20. Geometría rodamientos rígidos de bolas. .....- 45 -
Figura 21. Geometría de los rodamientos de bolas con contacto angular. .....- 46 -
Figura 22. Disposiciones en tándem, en X y en O respectivamente. .....- 46 -
Figura 23. Geometría de los rodamientos de rodillos de tipo NU y N respectivamente…. .....- 47 -
Figura 24. Geometría de los rodamientos de rodillos de tipo NJ y NUP respectivamente. .....- 47 -
Figura 25. Geometría de los rodamientos de agujas con y sin aro interior. .....- 47 -
Figura 26. Geometría de los rodamientos axiales de bolas. .....- 48 -
Figura 28. Geometría de las coronas de rodillos y de agujas respectivamente. .....- 48 -
Figura 29. Geometría de las arandelas de los rodamientos axiales de rodillos y agujas. .....…- 49 -
Figura 30. Desgaste de los rodamientos. .....- 49 -

Figura 31. Avería por adherencia. ..... - 50 -
Figura 32. Avería por desconchado. ..... - 51 -
Figura 33. Acceso a macros. ..... - 54 -
Figura 34. Entorno VBA. ..... - 56 -
Figura 35. Project Explorer y Properties Window. ..... - 57 -
Figura 36. Object browser. ..... - 58 -
Figura 37. Formulario de trabajo. ..... - 63 -
Figura 38. Herramientas de diseño. ..... - 64 -
Figura 39. Editor. ..... - 65 -
Figura 40. Comandos para hacer funcionar la aplicación. ..... - 65 -
Figura 41. Estructura del módulo Sketcher. ..... - 70 -
Figura 42. Estructura del módulo *Part Design*. ..... - 78 -
Figura 43. Estructura interna del *PartDocument*. ..... - 79 -
Figura 44. Estructura interna del *ProductDocument*. ..... - 88 -
Figura 45. Interfaz del programa. ..... - 92 -
Figura 46. Formulario de los rodamientos rígidos de bolas. ..... - 94 -
Figura 47. Creación de los aros (1). ..... - 95 -
Figura 48. Creación de los aros (2). ..... - 96 -
Figura 49. Creación de los aros (3). ..... - 97 -
Figura 50. Bola. ..... - 97 -
Figura 51. Bolas. ..... - 98 -
Figura 52. Creación de la jaula. ..... - 98 -
Figura 53. Montaje final del rodamiento de bolas. ..... - 99 -
Figura 54. Formulario de los rodamientos de bolas con contacto angular. ..... - 100 -
Figura 55. *Sketch* para los rodamientos de bolas con contacto angular. ..... - 100 -
Figura 56. Jaula de los rodamientos de bolas con contacto angular. ..... - 101 -
Figura 57. Montaje en tándem de dos rodamientos con contacto angular. ..... - 101 -
Figura 58. Formulario de los rodamientos axiales de bolas. ..... - 102 -
Figura 59. Montaje de un rodamiento axial de bolas. ..... - 102 -
Figura 62. Formulario de los rodamientos de los rodillos cilíndricos. ..... - 103 -
Figura 63. Montaje de un rodamiento de rodillos cilíndricos tipo NU. ..... - 103 -
Figura 64. Formulario de los rodamientos axiales y coronas de rodillos cilíndricos ..... - 104 -
Figura 65. Aro de un rodamiento axial de rodillos. ..... - 104 -

Figura 66. Montaje de un rodamiento axial de rodillos cilíndricos. ..... - 105 -
Figura 67. Formulario de las coronas de rodillos. ..... - 105 -
Figura 68. Montaje de una corona de rodillos cilíndricos. ..... - 106 -
Figura 69. Formulario de los rodamientos de agujas con aro interno. ..... - 106 -
Figura 70. Montaje de un rodamiento de agujas con aro interno. ..... - 107 -
Figura 71. Formulario de los rodamientos de agujas sin aro interno. ..... - 107 -
Figura 72. Montaje de un rodamiento de agujas sin aro interno. ..... - 108 -
Figura 73. Formulario de las coronas de agujas. ..... - 108 -
Figura 74. Montaje de las coronas de agujas. ..... - 109 -
Figura 75. Formulario de los rodamientos axiales de agujas. ..... - 109 -
Figura 76. Montaje de un rodamiento axial de agujas. ..... - 110 -
Figura 77. Interfaz de la aplicación Rodamientos SKF. ..... - 111 -
Figura 78. Ventana para escoger la carpeta de guardado. ..... - 112 -

Índice de tablas

Tabla 1. Aplicaciones de los rodamientos. .....- 44 -
Tabla 2. Datos generales de los rodamientos. .....- 45 -
Tabla 3. Conceptos de la programación. .....- 59 -
Tabla 4. Diferencias entre clase y objeto. .....- 63 -
Tabla 5. Opciones de la barra de herramientas. .....- 65 -
Tabla 6. Objetos del módulo *Sketcher*. .....- 70 -
Tabla 7. Restricciones. .....- 74 -
Tabla 8. Objetos del *PartDesign*. .....- 80 -
Tabla 9. Herramientas del *Shapefactory*. .....- 80 -
Tabla 10. Herramientas para programar un *hole*. .....- 83 -
Tabla 11. Herramientas del módulo *Assembly Design*. .....- 89 -

Índice de Códigos

Código 1. Arranque de documentos. ..... - 67 -
Código 2. Bodies. ..... - 68 -
Código 3. Sketches. ..... - 68 -
Código 4. Vector de coordenadas. ..... - 68 -
Código 5. Objeto factory 2D. ..... - 69 -
Código 6. Creación de un punto. ..... - 71 -
Código 7. Creación de una recta. ..... - 71 -
Código 8. Creación de un círculo. ..... - 72 -
Código 9. Creación de un arco de circunferencia. ..... - 72 -
Código 10. Creación de un spline. ..... - 73 -
Código 11. *CatCstTypeDistance*. ..... - 75 -
Código 12. *CatCstTypeRadius*. ..... - 75 -
Código 13. *CatCstTypeHorizontally/Vertically*. ..... - 76 -
Código 14. *CatCstTypeLength*. ..... - 76 -
Código 15. *CatCstTypeParallelism*. ..... - 76 -
Código 16. *CatCstTypeOn*. ..... - 77 -
Código 17. Definición del objeto *shapefactory*. ..... - 80 -
Código 18. *Pad*. ..... - 81 -
Código 19. Referencia del *pad* 1. ..... - 81 -
Código 20. Referencia del *pad* 2. ..... - 82 -
Código 21. *Pocket*. ..... - 82 -
Código 22. Referencia para un *hole*. ..... - 83 -
Código 23. *Hole*. ..... - 83 -
Código 24. *Slot*. ..... - 84 -
Código 25. *Shaft*. ..... - 85 -
Código 26. *Rib*. ..... - 86 -
Código 27. *CircularPattern*. ..... - 87 -
Código 28. *AddComponentsFromFiles*. ..... - 89 -
Código 29. *Fix Component*. ..... - 90 -
Código 30. *Offset Constraint*. ..... - 91 -
Código 31. *Coincidence Constraint*. ..... - 91 -
Código 33. Llamada a los *Userforms* con un *CommandButton*. ..... - 93 -

Código 34. Ventana emergente. .....- 94 -
Código 35. Guardado. .....- 94 -
Código 36. Chaflán. .....- 97 -

1. Introducción a la teoría de los rodamientos

1.1. Introducción

Cuando buscamos en la Real Academia Española la palabra rodamiento encontramos la siguiente definición:

"Cojinete formado por dos cilindros concéntricos, entre los que se intercala una corona de bolas o rodillos que pueden girar libremente."

Generalmente, un rodamiento es un elemento mecánico que sirve para soportar las cargas, tanto estáticas como dinámicas de un sistema que esté sometido a movimientos rotativos. Según como actúen las cargas, encontramos rodamientos radiales y axiales para cargas radiales y axiales respectivamente, además de algunos tipos que están diseñados para cargas combinadas.

Hoy en día, los rodamientos son usados en infinidad de aplicaciones ingenieriles que pueden ir desde maquinaria general hasta sofisticados sistemas aeronáuticos pasando por otras opciones como puede ser la energía eólica.

Este proyecto, está enfocado en la recreación en CATIA de los rodamientos del catálogo de la empresa SKF.

SKF es un proveedor de tecnología líder en el mundo desde 1907. Su punto fuerte es la capacidad para crear continuamente nuevas tecnologías y utilizarlas en la fabricación de productos que ofrecen ventajas competitivas a sus clientes. Esto lo consiguen combinando la experiencia práctica en más de 40 industrias con sus conocimientos en todas las plataformas tecnológicas de SKF rodamientos y unidades de rodamientos, obturaciones, mecatrónica, servicios y sistemas de lubricación. Todo su éxito se basa en esos conocimientos, en los empleados y en el gran compromiso con los principios del Cuidado SKF.

El desarrollo tecnológico actual de SKF presta especial atención a la reducción del impacto medioambiental de los activos durante su ciclo de vida, tanto en sus propias operaciones como en las de sus clientes. La cartera de productos SKF BeyondZero es el último ejemplo de lo que SKF puede ofrecer en este aspecto.

1.2. Breve descripción de la evolución histórica de los rodamientos

La idea de usar elementos rodantes para mover grandes pesos se remonta al antiguo Egipto. Los egipcios usaban los troncos de los árboles para hacer rodar sus grandes piezas de piedra hasta las áreas de construcción cuando construían las pirámides.

Figura 1. Representación del antiguo Egipto

Al principio los rodamientos fueron fabricados en *Lignum Vitae*, en latín "Madera de la vida", una madera muy pesada, dura y naturalmente aceitosa, nativa de Centro América y de los Indios del Este. Los aceites naturales en la madera ayudaron en los procesos de fabricación actuando como un fluido de corte. Estos rodamientos son mejor conocidos por sus aplicaciones en húmedo como en las hélices de navíos, ruedas de agua y bombas. Los rodamientos de madera fueron conocidos por ser muy resistentes (un promedio de aproximadamente 7 años de vida), fuertes, disponibles de inmediato y fáciles de reemplazar. Fueron lubricados con sebo o cualquier otra grasa animal. En muchas aplicaciones de turbinas, aún se puede encontrar rodamientos de madera aunque *Lignum Vitae* ya no está disponible como antes.

Leonardo Da Vinci, famoso por sus pinturas y dibujos, también tuvo muchas ideas sobre proyectos de ingeniería mecánica. Muchos de sus dibujos fueron bombas, montacargas, grúas y algunas armas de guerra. Durante sus trabajos como ingeniero hidráulico sirviendo al Duque de Milán, pasó mucho de su tiempo analizando los rodamientos, conexiones, ruedas y otros mecanismos de transmisión. Muchas de las ideas de Da Vinci están presentes en el mundo de la ingeniería de hoy día.

Con el siglo XVIII, los cambios en los procesos de producción fueron cambiando la forma de vida y de trabajo de las personas. El hierro fue volviéndose más y más popular y fue reemplazando a la madera en muchas factorías. Con los nuevos

progresos en fabricación había también una necesidad de más herramientas de precisión. El torno de madera es conocido por ser la máquina más antigua y a mediados del 1700, las innovaciones en el hierro permitieron la producción de máquinas de herramientas más precisas aún. Con los nuevos inventos, vino la necesidad de más fuentes de energía para la potencia de esas máquinas. La máquina de vapor se volvió una práctica fuente de energía. La innovación trajo consigo formas más sencillas de producir con buenas calidades y en mayor escala que antes. Esto permitió el crecimiento de la industria y la mayor necesidad de construir maquinaria que permitiera un nuevo estilo de rodamientos requeridos en la construcción de esas nuevas máquinas. Con el desarrollo de los nuevos rodamientos, vino la necesidad de nuevos materiales para su producción.

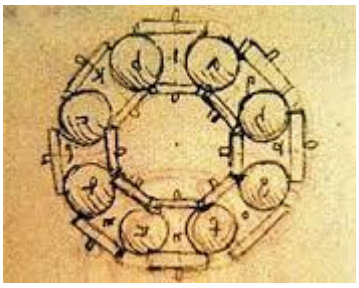

Figura 2. Dibujo de Leonardo

En 1839 Isaac Babbitt inventó una aleación antifricción con un bajo punto de fusión. Esta aleación pudo ser moldeada para producir una superficie ideal para los rodamientos. Con la introducción de este metal el uso de los rodamientos de madera disminuyo ligeramente. En los posteriores años del 1800, Henry Bessemer (1813-1898) ideó nuevos métodos de fabricación del acero. Su nuevo proceso permitió al acero, ser producido más económicamente. Esto permitió el uso del acero en la construcción de nuevos edificios y se volvió más ampliamente usado en rodamientos y máquinas de producción.

Con los inventos del Siglo XX como los coches, la robótica, ordenadores y las nuevas y rápidas máquinas de herramientas, los rodamientos se han vuelto mucho más significativos en las líneas de producción. Los materiales más nuevos nos han permitido producir rodamientos con un menor coste para el consumidor.

Hoy en día, los materiales usados para fabricar los rodamientos son muy variados y los podemos encontrar de distintas aleaciones de metales, mezclas de plásticos y en algunos casos aún se sigue usando la madera. *Referencia [1]*.

Figura 3. Rodamiento compuesto de diversos materiales.

1.3. Partes de un rodamiento

Excepto en diseños especiales, los rodamientos de bolas, de rodillos y de agujas se componen de las siguientes partes:

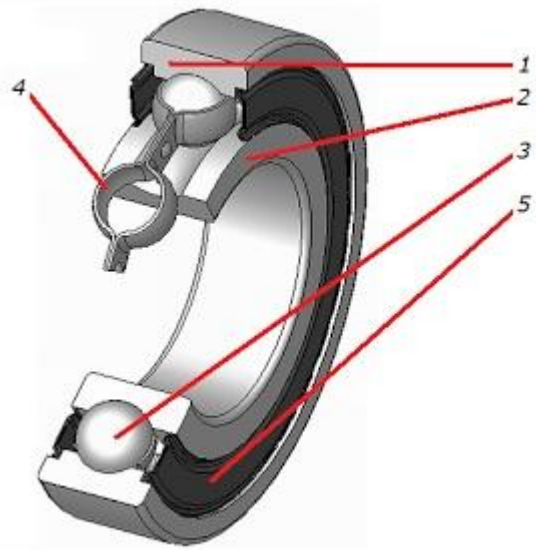

Figura 4. Partes de un rodamiento:1…2….

1. Un anillo o aro externo que integra una pista de rodadura (superficie en la cual "ruedan" los elementos rodantes)
2. Un anillo o aro interno con la misma función que el externo.
3. Elementos rodantes ya sean bolas, rodillos o agujas que permiten el movimiento de ambos anillos con una fricción mínima.
4. Un separador o jaula para separar uniformemente los elementos rodantes alrededor de la periferia.

5. Una protección que asegure la estanqueidad para que las partes activas del rodamiento estén siempre perfectamente limpias y bien lubricadas. Esta parte no se incluye siempre, sólo en casos necesarios.

1.4. Clasificación general de los rodamientos

Los rodamientos se clasifican principalmente, según el elemento rodante, en rodamientos de bolas o de rodillos y a su vez podemos distinguirlos de acuerdo al tipo de carga que soportan: radial, axial o combinada. Una primera clasificación *(referencia [5])* puede ser la siguiente:

- Rodamientos Radiales.
  - Rodamientos de bolas.
    - De ranura profunda.
    - De Magneto.
    - De contacto angular.
    - De tres-cuatro puntos de contacto.
    - A rótula.
    - Para soportes.
  - Rodamientos de rodillos.
    - Cilíndricos.
    - De rodillos largos.
    - De agujas.
    - Cónicos.
    - Esféricos.
- Rodamientos de Empuje.
  - Rodamientos de bolas.
    - Axiales.
    - De contacto angular.
  - Rodamientos de rodillos.
    - Axiales.
    - Axiales de agujas.
    - Cónicos.
    - Esféricos.

- Rodamientos para aplicaciones específicas.
    - De embrague para automoción.
    - De bomba de agua para automoción.
    - Para ejes de ferrocarril.
    - Para grúas roldanas.
    - Pivotantes.
    - Para transportadoras de cadena.
    - Otros.

1.5. Tipos de rodamientos

1.5.1. Rodamientos rígidos de bolas

El rodamiento rígido de bolas tiene gargantas profundas sin orificio para la entrada de las bolas. Debido a la profundidad de las gargantas, al tamaño de las bolas y al íntimo contacto entre bolas y las gargantas, este tipo de rodamiento tiene gran capacidad de carga, incluso en sentido axial; por consiguiente este rodamiento es muy adecuado para resistir cargas en todas direcciones. Su diseño le permite soportar un empuje axial considerable, aún funcionando a muy altas velocidades.

La figura 5 representa la ejecución normal de los rodamientos rígidos de bolas. También se fabrican en pequeños tamaños, rodamientos con placas de protección de chapa o arandelas de obturación de goma por uno o por los dos lados. Los rodamientos así protegidos, son apropiados para casos especiales en los que la obturación normal del soporte fuera insuficiente o no puede ser empleada.

Los rodamientos con dos placas de protección o dos arandelas de obturación, se entregan provistos de grasa; no deben por tanto ser lavados ni calentados al montarlos. Van provistos de una grasa de alta calidad y en una cantidad apropiada según el tamaño del rodamiento.

Las arandelas de obturación de goma aseguran una buena obturación y los rodamientos con dos arandelas, aunque por lo general no son relubricados, tienen una gran duración incluso a velocidad elevada. *Referencia [2]*.

Figura 5. Rodamientos rígidos de bola.

Para realizar el montaje de las bolas entre los anillos primero se rellena el espacio libre creado por la posición excéntrica de los anillos. El tamaño y el número de las bolas se calcularán de forma que mediante la utilización de la elasticidad de los anillos, el anillo interno pueda ser desplazado a una posición concéntrica con el anillo externo. Las bolas estarán entonces distribuidas uniformemente sobre la circunferencia y la jaula podrá ser insertada. *Referencia [3]*.

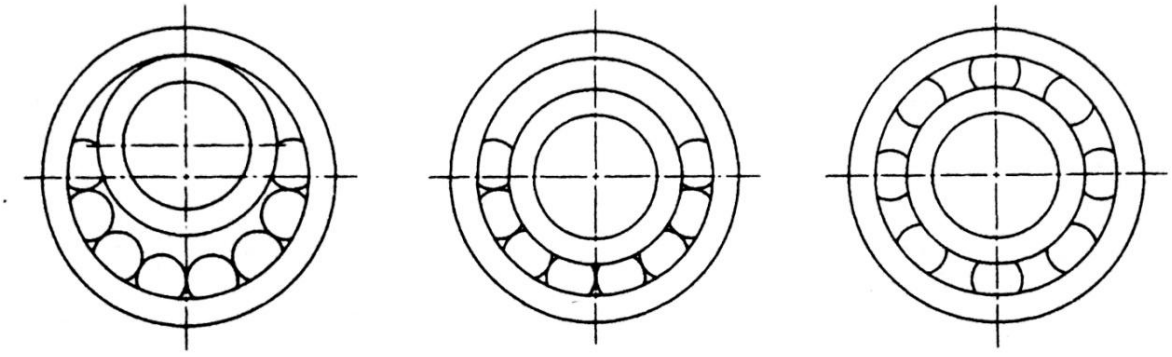

Figura 6. Montaje de un rodamiento de bolas.

Los rodamientos rígidos de bolas SKF pueden ser de una o de dos hileras. Los de dos hileras tienen escotes de llenado, que permiten introducir un mayor número de bolas en cada hilera. Los rodamientos tienen así una gran capacidad de carga radial,

aunque su capacidad para soportar carga axial queda limitada. También se suministran los rodamientos rígidos de dos hileras de bolas, en un diseño alternativo, sin escotes de llenado. *Referencia [4]*.

1.5.2. Rodamientos de Magneto

La ranura interior de los rodamientos de magneto esligeramente menos pronunciada que la de los anteriores. Puesto que el aro exterior sólo tiene un resalte a un lado, este puede ser desmontado por lo que este rodamiento resulta ventajoso a la hora del montaje. Suelen ser rodamientos con un diámetro interior pequeño que va de 4 a 30 mm y se utilizan principalmente en pequeños magnetos, giróscopos, instrumentos, etc. *Referencia [5]*.

1.5.3. Rodamientos de bolas a rótula

Los rodamientos de bolas a rótula tienen dos hileras de bolas y un camino de rodadura esférico común en el aro exterior. Debido a la esfericidad del camino de rodadura, el rodamiento es autoalineable, lo que permite pequeñas desviaciones angulares del eje con relación al soporte del rodamiento. Son, por tanto, especialmente adecuados para aplicaciones en las cuales se puedan producir desalineaciones por errores de montaje o por flexión del eje. Los rodamientos de bolas a rótula se fabrican tanto con agujero cilíndrico como con agujero cónico existiendo soportes apropiados para estos últimos.

Figura 7. Rodamientos de bolas a rótula.

Los rodamientos con aro interior ancho, se usan en general en aplicaciones con ejes que tengan las tolerancias comerciales normales. El agujero de estos rodamientos

tiene una tolerancia especial, lo cual permite un fácil montaje y desmontaje. *Referencia [4]*.

1.5.4. Rodamientos de bolas con contacto angular

En los rodamientos de bolas con contacto angular, la línea de acción de la carga, en los contactos entre las bolas y los caminos de rodadura, forma un ángulo con el eje del rodamiento. Los caminos de rodadura de los aros interior y exterior están axialmente desplazados entre sí y los rodamientos pueden soportar cargas combinadas (radiales y axiales).

En la actualidad existen rodamientos tanto de una hilera de bolas como de dos hileras.

El rodamiento de una hilera es de diseño no desmontable y resulta adecuado para altas velocidades. Una carga radial aplicada sobre el rodamiento da lugar a una carga axial inducida que debe ser contrarrestada y normalmente se disponen los rodamientos de modo que puedan ajustarse contra un segundo rodamiento. Los rodamientos pueden soportar cargas axiales en un sentido solamente.

También es posible encontrar rodamientos preparados para el montaje por parejas. Los rodamientos apareados pueden usarse si la carga que se ha de soportar es excesiva para un solo rodamiento, o bien si se han de soportar cargas axiales en ambos sentidos y se requiere tener controlado el juego axial.

Figura 8. Rodamientos de contacto angular de una hilera.

Un rodamiento de dos hileras de bolas tiene características similares a las de dos rodamientos de una hilera pero su anchura es menor que la de los dos rodamientos juntos. Es adecuado para soportar cargas axiales en uno u otro sentido, y puede absorber también pares de fuerza que actúen en un plano axial. Es de diseño desmontable, lo cual permite montar independientemente el conjunto exterior (aro exterior y corona de bolas) y el aro interior.

Los rodamientos de cuatro puntos de contacto son rodamientos de una hilera de bolas con los caminos de rodadura dispuestos de tal forma que pueden soportar cargas axiales en uno u otro sentido. *Referencia [4]*.

Figura 9: Rodamiento de bolas con contacto angular de dos hileras.

1.5.5. Rodamientos de rodillos cilíndricos

En los rodamientos de rodillos cilíndricos, los rodillos están guiados por pestañas existentes en uno de los aros mientras que el otro aro puede tener pestañas o no. El aro con pestañas y la corona de rodillos, forman un conjunto que puede separase del otro aro. Esta característica facilita su montaje y desmontaje en la aplicación, especialmente cuando se precisa ajuste de apriete para ambos aros, debido a las condiciones de carga. Estos rodamientos son adecuados para soportar elevadas cargas radiales y para funcionar a elevadas velocidades.

Figura 10. Rodamiento de rodillos cilíndricos.

Según sea la disposición de las pestañas, hay varios tipos de rodamientos de rodillos cilíndricos:

- Tipo NU: con dos pestañas en el aro exterior y sin pestañas en el aro interior.
- Tipo N: con dos pestañas en el aro interior y sin pestañas en el aro exterior.
- Tipo NJ: tienen tres pestañas, una de ellas en el aro interior, por lo que pueden usarse para la fijación axial del eje en un sentido.
- Tipo NUP: llevan un anillo apoyado sobre el aro interior, y pueden usarse para fijar al eje en ambos sentidos.

Los rodamientos de rodillos cilíndricos tipo NJ y NUP son capaces de soportar cierta carga axial además de la radial. *Referencia [4]*.

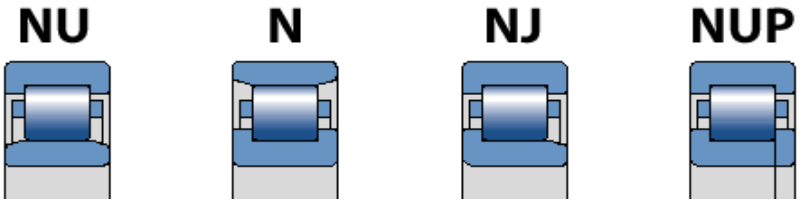

Figura 11. Tipos de Rodamientos de rodillos cilíndricos.

1.5.6. Rodamientos de agujas

Los elementos rodantes de los rodamientos de agujas están constituidos por rodillos cilíndricos de diámetro muy pequeño con respecto a su longitud, siendo esta la

principal característica de estos rodamientos. Debido a su pequeña altura de sección son particularmente adecuados para aplicaciones donde el espacio disponible radialmente sea limitado. Tienen con respecto a su reducida sección, una elevada capacidad de carga. Para asegurar una distribución uniforme de la carga, los rodillos tienen los extremos levemente bombeados.

Figura 12. Rodamientos de agujas.

Se pueden encontrar rodamientos de agujas de una o de dos hileras, con pestañas o sin ellas y con aro interior o sin este.

Los más comúnmente empleados son los rodamientos con pestañas. El aro exterior y la corona de agujas forman un conjunto no desmontable. En ciertos casos se facilita el montaje considerablemente al emplear rodamientos sin pestañas. No obstantes debe ser prevista, para estos casos, una fijación axial a través de las piezas adyacentes de la máquina.

Los rodamientos con aro interior permiten un desplazamiento axial del eje con relación al alojamiento, dentro de unos límites especificados. Puesto que son desmontables, el aro interior puede ser montado independientemente.

Donde sea posible templar y rectificar el eje para formar un camino de rodadura, se pueden montar rodamientos sin aro interior. Al no requerirse aro interior, puede aumentarse el diámetro del eje y con ello su rigidez. El emplazamiento axial del eje con relación al alojamiento está limitado solamente por el ancho del camino de rodadura mecanizado en el eje. *Referencia [4]*.

1.5.7. Rodamientos de rodillos a rótula

Los rodamientos de rodillos a rótula tienen dos hileras de rodillos que ruedan sobre un camino de rodadura esférico común en el aro exterior y cada uno de los caminos de rodadura del aro interior está inclinado formando un ángulo con el eje del rodamiento. Estos rodamientos son autoalineables y permiten pequeños desplazamientos angulares del eje con relación al alojamiento, como los que se pueden producir al efectuar el montaje o por flexión del eje bajo carga. Los podemos encontrar con agujero cilíndrico o cónico.

Figura 13. Rodamientos de rodillos a rótula.

El guiado de los rodillos lo efectúan la jaula y un anillo guía de gran sección radial, centrado sobre el aro interior y que puede alinearse por sí mismo entre las dos hileras de rodillos.

La disposición angular de los rodillos con relación al eje del rodamiento, permite aplicar cargas axiales de uno u otro sentido además de las cargas radiales. No obstante, los límites de velocidad admisibles se reducen a medida que aumenta la carga axial por lo que generalmente estos rodamientos no se recomiendan para cargas axiales puras. *Referencia [4]*.

1.5.8. Rodamientos de rodillos cónicos

En un rodamiento de rodillos cónicos, la línea de acción de la carga resultante a través de los rodillos forma un ángulo con el eje del rodamiento, por lo que estos rodamientos son particularmente adecuados para soportar cargas combinadas (radiales y axiales). Estos rodamientos son desmontables, es decir, que el aro exterior y el aro

interior con la corona de rodillos pueden montarse por separado.

Figura 14. Rodamientos de rodillos cónicos.

Los rodamientos de una hilera pueden soportar cargas axiales solamente en un sentido. Una carga radial aplicada sobre el rodamiento da lugar a una carga axial inducida, que debe ser contrarrestada, por lo que el rodamiento es ajustado generalmente contra un segundo rodamiento.

Los rodamientos están diseñados con contacto lineal modificado entre los rodillos cónicos y los caminos de rodadura, evitándose con ello los perjudiciales efectos de las cargas sobre los bordes. Esta característica junto con el gran número de rodillos largos, hace que estos rodamientos tengan alta capacidad de carga. *Referencia [4]*.

1.5.9. Rodamientos axiales de bolas

Estos rodamientos están diseñados para soportar cargas axiles no siendo adecuados para trabajos con cargas radiales. Al fin de impedir el deslizamiento en los puntos de contacto de las bolas con los caminos de rodadura, a causa de la fuerza centrífuga y pares giroscópicos, es necesario que los rodamientos estén sometidos a cierta carga axial mínima.

Los rodamientos axiales de bolas los podemos encontrar en dos diseños: de simple y de doble efecto. Los primeros pueden soportar cargas axiales actuando solamente en un sentido. Consisten en una arandela de soporte, una arandela de eje y una corona de bolas.

Figura 15. Rodamientos axiales de bolas.

Los rodamientos de doble efecto pueden soportar cargas axiales en ambos sentidos. Constan de una arandela de eje, dos arandelas de soporte y dos coronas de bolas. Las arandelas de soporte y las coronas de bolas son las mismas que para el correspondiente rodamiento de simple efecto.

Las arandelas de soporte pueden ser de asiento esférico o plano. Los rodamientos con asiento plano no admiten desalineaciones mientras que los que tienen asientos esféricos se montan generalmente sobre contraplacas esféricas, pudiendo de este modo admitir pequeños errores de desalineación entre la superficie de apoyo del alojamiento y el eje. *Referencia [4]*.

1.5.10. Rodamientos axiales de rodillos y de agujas

Los rodamientos axiales de agujas y los de rodillos cilíndricos están diseñados para proporcionar disposiciones de rodamientos rígidas, que puedan soportar grandes cargas, sean insensibles a las cargas de choque y requieran un espacio axial mínimo. Como regla general están constituidos por dos arandelas planas y una corona axial que consiste en una jaula con rodillos o agujas. Las agujas y los rodillos cilíndricos tienen sus extremos ligeramente bombeados para asegurar una distribución uniforme de la carga en el rodamiento.

Las combinaciones de coronas axiales de agujas o de rodillos cilíndricos con los diversos tipos de arandelas disponibles permiten diseñar disposiciones de rodamientos óptimas para las más diversas aplicaciones. De igual modo se consiguen disposiciones de rodamientos particularmente estrechas cuando las piezas adyacentes de la máquina

pueden adaptarse como caminos de rodadura, sustituyendo a las arandelas y de esta manera se puede incluso obtener disposiciones de rodamientos de mayor precisión, mecanizando los caminos de rodadura con la correspondiente precisión.

Figura 16. Rodamientos axiales de rodillos.

A pesar del movimiento combinado de deslizamientos y rodadura de los elementos rodantes, estos rodamientos pueden alcanzar elevadas velocidades de funcionamiento.

Con el fin de evitar un movimiento de deslizamiento entre los elementos rodantes y los caminos de rodadura a causa de la fuerza centrífuga y de los pares giroscópicos, los rodamientos deben estar siempre sometidos a una carga axial mínima. *Referencia [4]*.

1.5.11. Rodamientos axiales de rodillos a rótula

Estos rodamientos tienen una hilera de rodillos situados oblicuamente, los cuales guiados por una pestaña del aro fijo al eje, giran sobre la superficie esférica del aro apoyado en el soporte. En consecuencia, el rodamiento posee una gran capacidad de carga y es autoalineable.

Figura 17. Rodamiento axial de rodillos a rótula.

Los rodillos están lubricados por una fina capa de aceite lo que los permite girar a gran velocidad, aun soportando una elevada carga.

Contrariamente a los otros rodamientos axiales, este también es capaz de resistir cargas radiales. *Referencia [2]*.

1.6. Materiales de los rodamientos

El rendimiento y la fiabilidad de los rodamientos vienen determinados en gran medida por los materiales de los cuales se fabrican los componentes de los mismos. Entre las consideraciones típicas para los aros y los elementos rodantes de los rodamientos se encuentran la dureza para la capacidad de carga, la resistencia a la fatiga bajo condiciones de contacto de rodadura, bajo condiciones de lubricación limpias o contaminadas, y la estabilidad dimensional de los componentes del rodamiento. Entre las consideraciones para la jaula se incluyen la fricción, la deformación, las fuerzas de inercia y, en algunos casos, el efecto químico de determinados lubricantes, disolventes, enfriadores y refrigerantes. También hay que tener en cuenta otras consideraciones como la corrosión, las temperaturas elevadas, las cargas de choque o la combinación de estas y otras condiciones.

1.6.1. Materiales usados para los aros y los elementos rodantes de los rodamientos

- Aceros de temple total.

El acero de temple total más comúnmente usado para los rodamientos es un acero al cromo rico en carbono, que contiene aproximadamente un 1% de carbono y un 1.5% de cromo. Normalmente, este acero recibe un tratamiento térmico martensítico o bainítico para aumentar su dureza.

- Aceros templados por corrientes de inducción.

El templado de la superficie por corrientes de inducción ofrece la posibilidad de templar de forma selectiva el camino de rodadura de un componente sin que el resto del componente se vea afectado por este proceso.

- Aceros de cementación.

Los aceros aleados al cromo-níquel y al cromo-manganeso con un contenido de

carbono de aproximadamente el 0.15% son los aceros más utilizados para cementar los rodamientos.

- Aceros inoxidables.

Los más usados son aquellos con un alto contenido en cromo. De igual manera se pueden usar recubrimientos resistentes a la corrosión que pueden ser una alternativa al acero inoxidable.

- Cerámica.

La cerámica normalmente utilizada para fabricar los aros y los elementos rodantes de los rodamientos es un material de nitruro de silicio con calidad para rodamientos. Está formado por granos alargados de nitruro de silicio beta en una matriz vítrea. Ofrece una combinación de propiedades favorables para los rodamientos, como una gran dureza, baja densidad, baja dilatación térmica, gran resistencia a la electricidad, baja constante dieléctrica y no se ve afectada por los campos magnéticos

- Materiales poliméricos.

Los rodamientos de bolas de material polimérico suelen utilizarse en aplicaciones en las que la resistencia a la humedad o a las sustancias químicas es un factor esencial y en las que no pueden utilizarse rodamientos de acero, o bien estos pueden utilizarse con ciertas restricciones.

1.6.2. Materiales usados para las jaulas

- De chapa de acero.

La mayoría de las jaulas embutidas en chapa de acero están hechas con un bajo contenido de carbono laminado en caliente. Tienen una resistencia relativamente alta, y su superficie puede ser tratada para reducir en mayor medida la fricción y el desgaste.

Las que normalmente se usan en rodamientos de acero inoxidable están hechas de acero inoxidable.

- Jaulas mecanizadas de acero.

Las jaulas mecanizadas de acero se usan para los rodamientos de gran tamaño, o en aplicaciones en las que existe el riesgo de que se produzca una rotura por corrosión

intergranular, causada por una reacción química sí se usara una jaula de latón.

- De chapa de latón.

Las jaulas de chapa de latón se usan en algunos rodamientos pequeños y medianos. La mayoría están mecanizadas de latón moldeado o forjado.

- Jaulas de polímero.

Para la mayoría de las jaulas moldeadas por inyección, se utiliza la poliamida 6,6. Este material, con o sin refuerzo de fibra de vidrio, se caracteriza por una combinación favorable de resistencia y elasticidad. Las propiedades mecánicas de los materiales poliméricos dependen de la temperatura y están sometidas a cambios permanentes bajo las condiciones de funcionamiento, a lo que se denomina envejecimiento.

- Resina fenólica.

Las jaulas de resina fenólica ligeras, reforzadas con tejido, pueden resistir grandes fuerzas centrífugas y de aceleración, pero no pueden soportar altas temperaturas de funcionamiento. *Referencia [6]*.

Figura 18. Ejemplo de jaula.

1.7. Selección del tipo de rodamiento

Cada tipo de rodamiento tiene propiedades características que lo hacen particularmente adecuado para ciertas aplicaciones. Sin embargo, no es posible

establecer reglas rígidas para la selección del tipo de rodamiento, pues para ello se han de considerar diversos factores. Aquí se dan una serie de indicaciones para poder hacer una elección más adecuada. *Referencia [4]*.

1.7.1. Espacio disponible

Hay muchos casos en que al menos una de las dimensiones principales del rodamiento, generalmente el diámetro del agujero, viene determinada por las características de diseño de la máquina a la que va destinado.

Normalmente se seleccionan rodamientos rígidos de bolas para ejes de pequeño diámetro, mientras que para ejes de grandes diámetros se pueden considerar los mismos, los de rodillos cilíndricos y los de rodillos a rótula.

Cuando el espacio radial es limitado, deberán seleccionarse rodamientos de pequeña sección como por ejemplo las coronas de agujas, rodamientos de agujas con o sin aro interior, ciertas series de rodamientos rígidos de bolas y de rodamientos de rodillos a rótula.

Cuando la limitación es en sentido axial y se requieren rodamientos particularmente estrechos, pueden usarse algunas series de rodamientos de una hilera de rodillos cilíndricos o rígidos de bolas para cargas radiales y combinadas, y para cargas axiales, coronas axiales de agujas, rodamientos axiales de agujas y algunas series de rodamientos axiales de bolas.

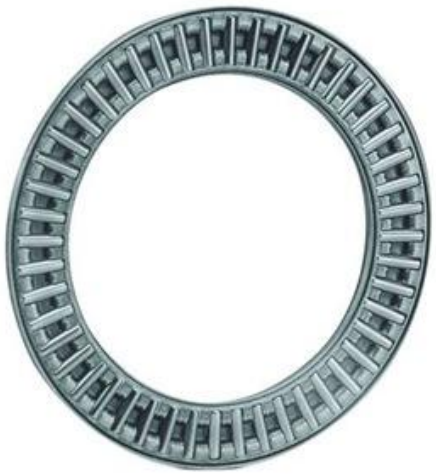

Figura 19. Corona axial de agujas.

1.7.2. Cargas sobre el rodamiento

- Magnitud de la carga.

Este es normalmente el factor más importante para determinar el tamaño del rodamiento. En general, para unas mismas dimensiones principales los rodamientos de rodillos pueden soportar mayores cargas que los rodamientos de bolas.

- Dirección de la carga.

Para una carga radial, los rodamientos de rodillos cilíndricos con un aro sin pestañas y los rodamientos de agujas, solamente pueden soportar cargas radiales. Todos los demás tipos pueden soportar cargas tanto radiales como axiales.

Los rodamientos axiales de bolas pueden soportar sólo moderadas cargas axiales puras. Los rodamientos axiales de bolas pueden soportar cargas axiales en un sentido y los de doble efecto, cargas axiales en ambos sentidos. Los rodamientos axiales de rodillos cilíndricos y los axiales de agujas pueden soportar elevadas cargas axiales en un sentido mientras que los axiales de rodillos a rótula son capaces de soportar una cierta magnitud actuando simultáneamente en dirección radial.

Para soportar cargas combinadas se usan principalmente los rodamientos de bolas con contacto angular de una o de dos hileras y los rodamientos de rodillos cónicos.

1.7.3. Desalineación angular

Cuando existe la posibilidad de desalineación del eje con respecto al soporte, se necesitan rodamientos capaces de absorber tal desalineación, es decir, rodamientos de bolas a rótula, rodamientos de rodillos a rótula y rodamientos axiales de rodillos a rótula.

1.7.4. Límites de velocidad

La velocidad de rotación de un rodamiento viene limitada por la temperatura máxima de funcionamiento permisible. Los rodamientos de bajo rozamiento dan lugar a una escasa generación interna de calor y son los más adecuados para altas velocidades de rotación. Con cargas radiales se pueden obtener las máximas velocidades de rotación empleando rodamientos rígidos de bolas o de rodillos cilíndricos y para cargas

combinadas emplearemos rodamientos de bolas con contacto angular.

1.7.5. Precisión

Se requieren rodamientos de grado de precisión mayor que el normal para ejes que tengan que funcionar con rigurosas exigencias de exactitud, por ejemplo, para husillos de máquinas-herramientas y generalmente también para ejes que giren a velocidades muy elevadas.

1.7.6. Rigidez

La deformación elástica de un rodamiento cargado es muy pequeña y en la mayoría de los casos, despreciable. No obstante, en algunos casos la rigidez del rodamiento es un factor importante como por ejemplo para husillos de máquinas-herramientas.

Debido a la mayor superficie de contacto entre los elementos rodantes y los caminos de rodadura, los rodamientos de rodillos se deforman menos que los de bolas. Puede aumentarse la rigidez de los rodamientos aplicando una precarga adecuada.

1.7.7. Desplazamiento axial

La disposición normal de los rodamientos en un eje u otro elemento de maquinaria consiste en un rodamiento posicionador o fijo y uno o más rodamientos libres.

Un rodamiento libre puede desplazarse axialmente evitando así un contraste recíproco entre los rodamientos, por ejemplo por dilatación o contracción del eje. Como rodamientos libres, son particularmente adecuados los rodamientos de rodillos cilíndricos que tienen uno de los aros sin pestañas o los rodamientos de agujas. *Referencia [4]*.

1.8. Aplicaciones de los rodamientos

Los rodamientos forman parte de muchos tipos de máquinas de todas las industrias, desde maquinarias pesadas hasta dispositivos de simulación y entretenimiento. En la siguiente tabla se recogen algunas de la infinidad de aplicaciones que tienen los rodamientos.

Industria	Aplicación
Aeroespacial y defensa	Aeronaves (jet de combate, helicópteros, UAVs (Unmanned Aerial Vehicle), comerciales, ligeros…): sistema de aterrizaje, propulsión y sistemas de control, bombas de refuel, APUs (Unidad Auxiliar de Potencia), elevador de rescate, armas…
	Barcos: sonar, radar y comunicaciones, defensa de misiles, periscopio submarino…
	Vehículos de seguridad y vehículos de combate en tierra: suspensión, transmisión, sistemas de observación y enfoque…
	Armas: sistemas de guía y propulsión, trípodes, cargadores de munición, lanzadores de misiles y morteros, etc.
	Espacio: satélites, despliegue de paneles solares, estación espacial, telescopios…
Transporte de materiales industriales	Barrenas y recuperadores.
	Excavadoras de cubeta.
	Esparcidor.
	Apiladoras y recuperadoras.
Maquinaria industrial	Máquinas herramientas.
	Sistemas de automoción y robots.
	Equipos de fotografía y óptica.
	Equipos de procesamiento de comida.
	Elevadora de pallets y máquina para envolver pallets.
	Máquinas de hilar y maquinaria textil.
	Máquinas de impresión.
	Máquinas de fabricación de pañales de papel.
Transporte	Dispositivo antihielo.
	Escáneres de seguridad de equipajes.
	Mecanismos para direccionar los raíles de tren.
	Pasarela de las aerolíneas para los pasajeros.
Equipamientos municipales	Ascensores de inspección de puentes.
	Puentes rotantes.
	Grúas en general.
	Escaleras de los camiones de bomberos.
	Tratamiento de aguas: filtros, aglutinantes y distribuidores rotatorios.
Sistemas médicos	Escáneres.
	Robótica de asistencia a humanos.
	Máquinas de tratamiento oncológico.
	Rayos X.
	Equipos de diagnóstico.
Energías renovables	Turbinas de viento.
	Mecanismo para orientar los paneles solares.
	Despliegue de paneles solares.
Entretenimiento	Atracciones de feria.
	Simuladores de vuelo.
	Edificios y habitaciones rotatorias.
	Plataformas y pantallas rotatorias.

Maquinaria pesada	Equipos para la construcción: camiones basculantes, grúas, camiones de cemento, bombas, excavadoras, equipamiento de pavimentos, tuneladoras…
	Forestal: trituradoras, apiladoras, cargadores.
	Minería: cubetas, minadores continuos, palas hidráulicas y sistemas de transporte, tamizadores del suelo, aglutinantes…
	La Marina: propulsores, cabestrantes, pescantes, ROVs (Remotely Operated Vehicle), cargadores de barcos, grúas de carga…

Tabla 1. Aplicaciones de los rodamientos.

1.9. Descripción de la geometría de los rodamientos que se van a realizar

Para definir la geometría de cada rodamiento, se ha empleado el catálogo general de la empresa SKF el cual ofrece casi todos los tipos de rodamientos en una serie de diámetros y anchuras definidos en unas tablas. Por tanto, para obtener un rodamiento con unas especificaciones concretas, se recomienda usar dicho catálogo para que los resultados sean razonables.

La razón por la que se han implementado los códigos de las macros para los rodamientos aquí descritos es la de contar con todos los parámetros necesarios para poder definir correctamente la geometría. En el caso de los rodamientos de bolas a rótula y de rodillos a rótula, carecemos del radio de curvatura de los canales de rodaduras, el cual es indispensable para limitar la posible desalineación del aro interno y para los rodamientos de rodillos cónicos se necesita conocer el ángulo de contacto que define la inclinación de los rodillos. Como se comentó en la descripción de cada tipo de rodamiento, esta inclinación es la responsable de producir una carga axial inducida a partir de una carga radial. Para solucionar estos problemas y obtener dichas especificaciones, se ha contactado con la empresa productora SKF pero los correos enviados no han tenido respuesta por parte del departamento que trabaja este tema.

Por lo tanto, se ha desarrollado una aplicación en VBA para CATIA V5 que nos permite hacer una representación en 3D de los rodamientos rígidos de bolas, de bolas con contacto angular en varias configuraciones, de los 4 tipos de rodillos cilíndricos, de agujas, axiales de bolas y axiales de rodillos cilíndricos y de agujas.

En la siguiente tabla se recogen los datos que son necesarios para definir los rodamientos:

Datos generales	Definición
d	Diámetro nominal del agujero.
d1	Diámetro de la superficie cilíndrica exterior del aro interior.
D	Diámetro nominal exterior.
D1	Diámetro de la superficie cilíndrica interior del aro externo.
B, C	Anchura de los aros.
r, r1	Ángulos de chaflán de los bordes de los aros.
a	Distancia sobre el eje del rodamiento donde se define el ángulo de contacto.
E, F	Definen el diámetro de los rodillos.
Ea, Eb	Definen la longitud de los rodillos en los rodamientos axiales.
H	Anchura total de los rodamientos axiales.
Dw	Diámetro del elemento rodante.

Tabla 2. Datos generales de los rodamientos.

1.9.1. Rodamientos rígidos de bolas

Para definir estos rodamientos necesitaremos los siguientes datos: d, d1, D, D1, B y r.

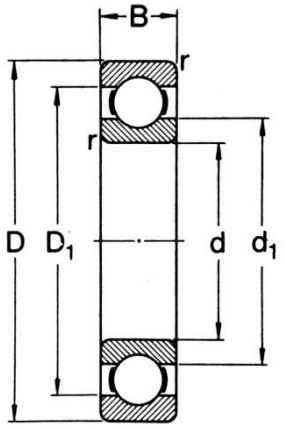

Figura 20. Geometría rodamientos rígidos de bolas.

1.9.2. Rodamientos de bolas con contacto angular

SKF diseña este tipo de rodamientos con diferentes ángulos de contacto según la configuración requerida: si deseamos un rodamiento de bola de contacto angular simple, el ángulo de contacto será de 40° al igual que si necesitamos dos rodamientos de contacto angular montados en tándem, en X o en O, el ángulo sigue siendo 40°. Los datos que son necesarios son: d, d1, D, D1, B, r, r1 y a.

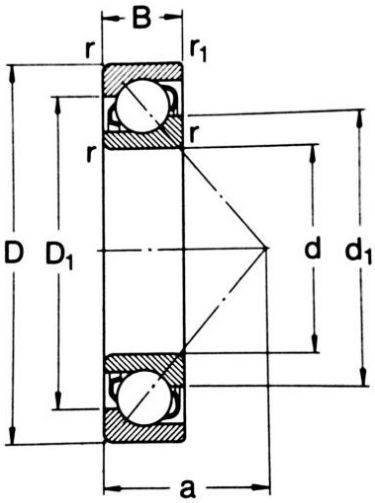

Figura 21. Geometría de los rodamientos de bolas con contacto angular.

Una carga radial aplicada sobre el rodamiento da lugar a una carga axial inducida que debe ser contrarrestada y normalmente se disponen rodamientos de modo que puedan ajustarse contra un segundo rodamiento. Los rodamientos apareados pueden montarse en tándem cuando la carga que se ha de soportar es excesiva o bien se montan en O ó en X si se han de soportar cargas axiales en ambos sentidos y se requiere tener contralado el juego axial.

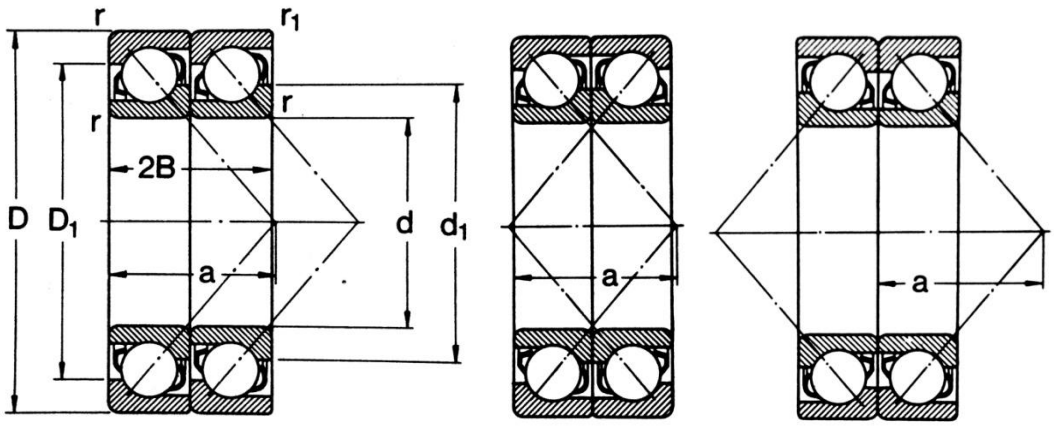

Figura 22. Disposiciones en tándem, en X y en O respectivamente.

1.9.3. Rodamientos de rodillos cilíndricos

Se puede elegir entre los 4 tipos de rodamientos de rodillos cilíndricos disponibles y los datos necesarios son los mismos en las 4 configuraciones: d, d1, D, D1, F, E, B, r y r1 .

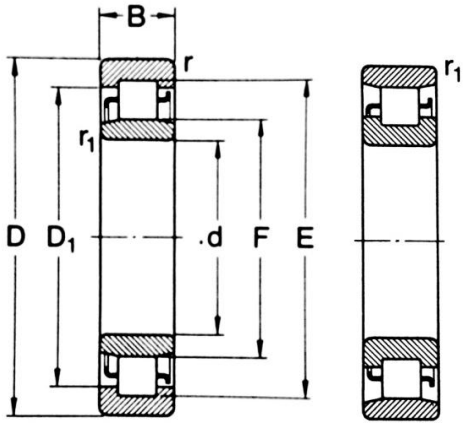

Figura 23. Geometría de los rodamientos de rodillos de tipo NU y N respectivamente.

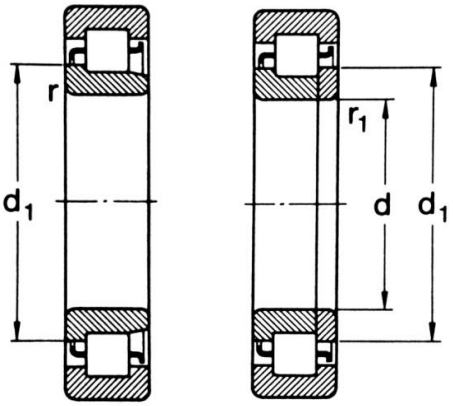

Figura 24. Geometría de los rodamientos de rodillos de tipo NJ y NUP respectivamente.

1.9.4. Rodamientos de agujas

Las configuraciones disponibles son las de rodamientos de agujas con pestañas y aro interior y el mismo pero sin aro interior. Los datos geométricos necesarios son: d, D, D1, F, C y r.

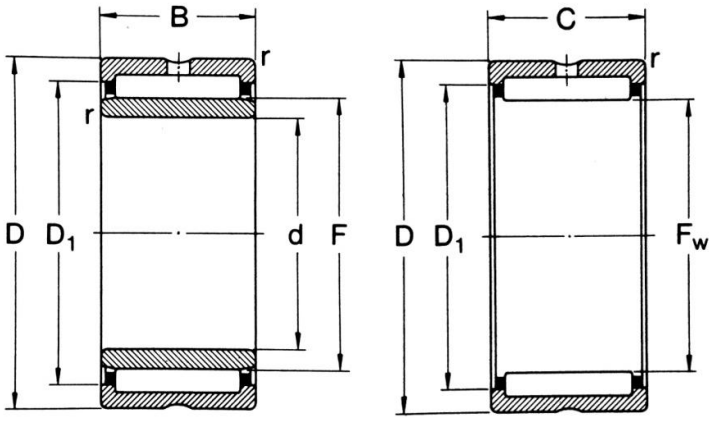

Figura 25. Geometría de los rodamientos de agujas con y sin aro interior.

1.9.5. Rodamientos axiales de bolas

Para que el programa procese correctamente un rodamiento axial de bolas de simple efecto hay que facilitarle los siguientes datos: d, d1, D, D1, H y r.

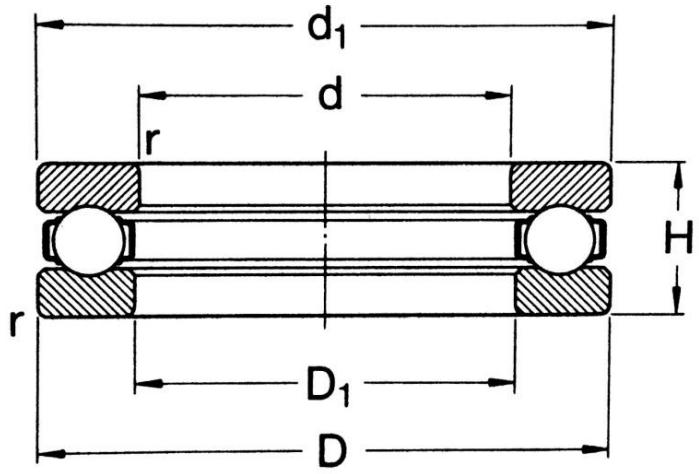

Figura 26. Geometría de los rodamientos axiales de bolas.

1.9.6. Rodamientos axiales de rodillos y agujas

Para estos tipos de rodamientos podremos elegir entre una simple corona de rodillos o de agujas cuando las limitaciones de espacio axial así lo exijan y el rodamiento completo con sus dos aros y su corona en el medio.

Los datos necesarios son: d, d1, D, D1, Dc, Ea, Eb, Dw, B y r.

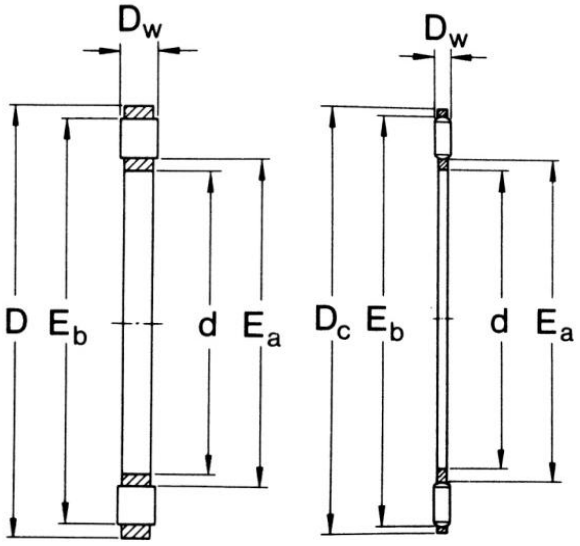

Figura 27. Geometría de las coronas de rodillos y de agujas respectivamente.

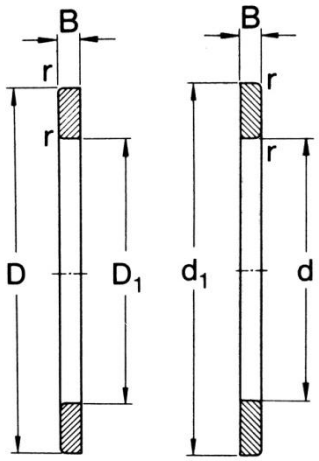

Figura 28. Geometría de las arandelas de los rodamientos axiales de rodillos y agujas.

Referencia [4].

1.10. Posibles averías

Algunas veces sucede que un rodamiento no alcanza su duración de vida calculada. Puede haber varias razones para esto, por ejemplo, cargas más pesadas que las previstas, lubricación inadecuada o insuficiente, manipulación negligente, obturaciones ineficaces, etc. Cada uno de estos factores origina su propio tipo de avería e imprime su particular huella en el rodamiento. Consecuentemente, examinando un rodamiento averiado, en la mayoría de los casos es posible determinar las causas del fallo y adoptar las medidas precisas para evitar su repetición. Las averías más frecuentes se comentan a continuación:

- Desgaste: en casos normales, no existe desgaste apreciable en rodamientos rotativos. Sin embargo, el desgaste puede ocurrir como resultado de la presencia de partículas extrañas dentro del rodamiento o cuando la lubricación es insatisfactoria. Las vibraciones en rodamientos que permanezcan estacionarios también producen desgaste.

Figura 29. Desgaste de los rodamientos.

- Indentaciones: los caminos de rodadura y los elementos rodantes pueden llegar a presentar indentaciones si la presión de montaje que se transmite a través de los elementos rodantes esta aplicada al aro equivocado o si el rodamiento está sujeto a cargas anormales mientras no gira. Las partículas extrañas en el rodamiento también pueden causar indentaciones.
- Adherencias: cuando dos superficies inadecuadamente lubricadas deslizan entre sí bajo cargas, el material se transfiere de una a otra y las superficies correspondientes llegan a desgarrarse y presentar mal aspecto. Cuando se produce adherencia, el material alcanza generalmente tales temperaturas que da lugar al recalentado. Esto produce concentraciones de fatiga localizadas que pueden originar agrietamiento o desconchado.

Figura 30. Avería por adherencia.

- Fatiga superficial: si la película lubricante entre los caminos de rodadura y los elementos rodantes llega a ser demasiado fina, las crestas de asperezas superficiales entraran momentáneamente en contacto entre sí. Esto producirá pequeñas grietas en la superficie que van creciendo gradualmente hasta un tamaño tal que interfieren el suave rodaje del rodamiento.
- Corrosión: se formará óxido si se introducen agua o agentes corrosivos en el rodamiento en tal cantidad que el lubricante no puede ofrecer protección para la superficie del acero. Este proceso conducirá pronto a zonas de óxido profundo. Otro tipo de corrosión es la oxidación de contacto.
- Desconchado: se presenta como resultado de la fatiga normal, es decir, el rodamiento ha alcanzado el fin de su duración normal esperada. Cuando el

descascarillado ha proseguido hasta cierto nivel, hace notar su presencia en forma de ruido y vibración, lo que sirve como advertencia de que es el momento para sustituir el rodamiento. Las causas de un desconchado prematuro pueden ser unas cargas externas muy pesadas, precargas a causa de ajustes incorrectos, excesivo calado en asiento cónico, por indentacion, corrosión profunda o adherencia.

- Grietas: pueden formarse en los aros del rodamiento por varias razones. La causa más común es un trato brusco cuando el rodamiento está siendo montando o desmontado. Golpes de mazo directamente aplicados contra el aro o mediante un botador templado, pueden originar la formación de finas grietas, dando lugar a que trozos de dicho aro salten cuando el rodamiento se pone en funcionamiento. Otra causa del agrietamiento es calar excesivamente sobre un asiento cónico. El límite de elasticidad se sobrepasa en el aro como resultado de la excesiva penetración y origina la fractura cuando el rodamiento se pone en servicio. *Referencia [8]*.

Figura 31. Avería por desconchado.

2. El entorno de programación VBA en CATIA V5

En este capítulo se va a presentar el programa CATIA V5 y se va a explicar a grandes rasgos el lenguaje de programación "Visual Basic for applications" (VBA). Una vez introducido el entorno en el que se va a trabajar, se procederá a explicar el entorno general de trabajo de VB6, así como nociones generales de programación de dicho lenguaje.

2.1. CATIA V5

CATIA (computer-Aided Three dimensional Interactive Application) es un programa informático de diseño, fabricación e ingeniería asistida por computadora comercial, realizado por Dassault Systèmes. CATIA es la solución líder en todo el mundo para la experiencia y el diseño de productos. Organizaciones líderes de distintos sectores la utilizan para desarrollar los productos que vemos y usamos en nuestra vida cotidiana.

Esta herramienta ofrece la posibilidad única no solo de modelar cualquier producto, sino de hacerlo en el contexto de su comportamiento en la vida real: *diseño en la era de la experiencia*. Los arquitectos de sistemas, los ingenieros, los diseñadores y todos sus colaboradores pueden definir el mundo que nos conecta, imaginarlo y darle forma.

CATIA, que se basa en la plataforma 3DEXPERIENCE de Dassault Systèmes, ofrece lo siguiente:

- Entorno de diseño social basado en una fuente única de autenticidad, al que se accede mediante potentes paneles en 3D que impulsan la inteligencia empresarial, el diseño simultáneo en tiempo real y la colaboración de todas las partes interesadas, incluidos los trabajadores.
- 3DEXPERIENCE ofrece una experiencia intuitiva con funcionalidades de modelado y simulación en 3D de primer nivel que optimizan la eficacia de todos los usuarios.
- Se trata de una plataforma inclusiva de desarrollo de productos, que resulta fácil de integrar con los procesos y las herramientas existentes. Esto permite que varias disciplinas aprovechen las eficaces e integradas

aplicaciones especializadas en todas las fases del proceso de desarrollo de los productos.

Referencia [9].

2.2. Visual Basic for Application

Como su nombre indica, el lenguaje de programación *Visual Basic* procede del denominado *BASIC* (*Beginner´s All-purpose Symbolic Instruction Code*) que fue creado en 1964 en el *Dartmouth College*, como un medio para iniciarse en el mundo de la programación. Tras varias modificaciones, en 1978 se estableció el *BASIC standard*. Primero fue nombrado como *GW-BASIC*, luego quedó en *QuickBASIC* y actualmente se le conoce como *Visual Basic* tras adaptarse al entorno de ventanas "*Windows*" e incorporar herramientas de tipo visual como botones, listas de texto o cuadros de texto asociadas a eventos. La primera versión de *Visual Basic* fue presentada en 1991, siendo la última la versión 6, liberada en 1998.

Su utilidad principal es automatizar tareas cotidianas, así como crear aplicaciones y servicios de bases de datos para el escritorio. Permite acceder a las funcionalidades de un lenguaje orientado a eventos con acceso a la API de *Windows*.

Visual Basic for Applications (VBA) es el lenguaje de macros de *Visual Basic v6*, incorporado en muchas aplicaciones de *Microsoft* y posteriormente en otras aplicaciones para ampliar la funcionalidad de las mismas. VBA incorpora las librerías y herramientas de *Visual Basic,* a las que añade librerías de objetos propias de cada software donde está incluido. La debilidad de este lenguaje radica en que la compilación de la macro no puede realizarse si no se dispone del entorno en el que se ha desarrollado. Otra debilidad es su falta de versatilidad para trabajar en otros sistemas operativos.

CATIA en 1998 con la versión V5 incorporó VBA a su entorno, pudiendo realizar macros en VB y en lenguaje C++, siendo aún los lenguajes de macros que se han dispuesto para su versión V6. *Referencia [10]*.

2.2.1. Entorno

En primer lugar se va a explicar que es una Macro. Una macro consiste en una serie de funciones escritas en un lenguaje de programación que agrupa una serie de

comandos, los cuales permiten realizar las operaciones requeridas automáticamente. Son usadas para ahorrar tiempo y reducir la posibilidad de errores humanos a la hora de realizar operaciones que se lleven a cabo de forma repetitiva.

El uso de Macros para la automatización en el proceso de diseño es prácticamente ilimitado, siendo algunos ejemplos la importación de puntos desde Excel a un modelo CAD 3D, la generación de geometrías de manera automática, la creación de planos de modelos 3D, etc.

Para poder acceder al entorno de trabajo VBA no hay más que pinchar en la pestaña *Tools* desde cualquier módulo de CATIA, donde encontramos la opción Macros y dentro de ésta se podrá o bien comenzar a grabar una macro, acceder a las macros ya realizadas y librerías o al editor de *Visual Basic*. *Referencia [11]*.

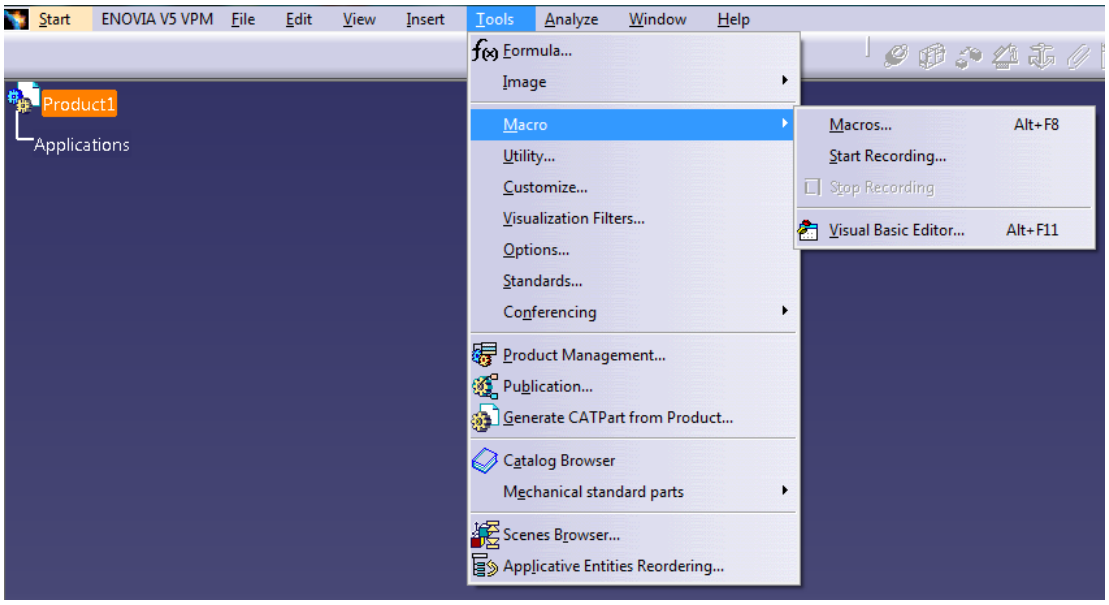

Figura 32. Acceso a macros.

2.2.2. Librerías de las macros

Las macros de CATIA son almacenadas en las librerías de macros de tres formas posibles: *Folders* (vbscript y CATScript), *Project files* (catvba) o CATParts/CATProducts. Solo una de estas tres librerías de macros puede ser usada a la vez. Para crear una nueva librería el procedimiento a usar es el siguiente:

1. Ir a *Tools* → *Macro* → *Macros*.
2. Abrir "*Macro libraries*".

3. Hay que asegurarse de que el tipo de librería esté cargada en "*directories*" y luego se hace clic en "*Add existing library"*.
4. Seleccionar la carpeta donde se van a guardar los CATScripts a lo largo del proyecto.
5. Cerrar la librería de macros. En dicha librería creada deberían aparecer la lista de CATScripts que se realicen. *Referencia [11]*.

2.2.3. Macro Recording

Un método para crear macros es grabando las acciones que se realicen con el ratón. Para macros grabadas en un fichero o en un CATpart o CATproduct, los estamentos declarados se grabaran para CATScript pero no para MSVBscritp. Para macros grabadas en una librería, "MS VBA" es la única opción. A la hora de grabar una macro mediante este procedimiento hay que tener en cuenta algunos aspectos:

- No seleccionar *Workbenches* (entornos de trabajo) mientras se está grabando una macro.
- No grabar más de lo que sea absolutamente necesario.
- No usar la opción "deshacer" mientras se está grabando.
- Ser consciente y darse cuenta de la configuración de CATIA cuando se está grabando.
- Salir de los *sketches* (dibujos) antes de parar de grabar.
- Verificar cada macro una vez se haya grabado.

Una vez se haya finalizado la grabación, se deshará todo lo realizado y se reproducirá la macro, de manera que se podrá comprobar si es correcta la macro y si reproduce la operación que se quería realizar.

También hay que tener en cuenta que mediante dicho procedimiento aparecerán numerosas líneas de código que no son realmente necesarias, por lo que pueden eliminarse.

Por otro lado, tampoco aparecerán comentarios acerca de lo que se está realizando o explicando los parámetros de entrada, por lo que se deberán añadir manualmente. *Referencia [11]*.

2.3. Visual Basic Editor

Este será el entorno sobre el que desarrollaremos nuestro proyecto. Para acceder al mismo, como ya se ha comentado, entramos en *Tools* → *Macro* → *Visual Basic Editor*, obteniéndose la siguiente ventana:

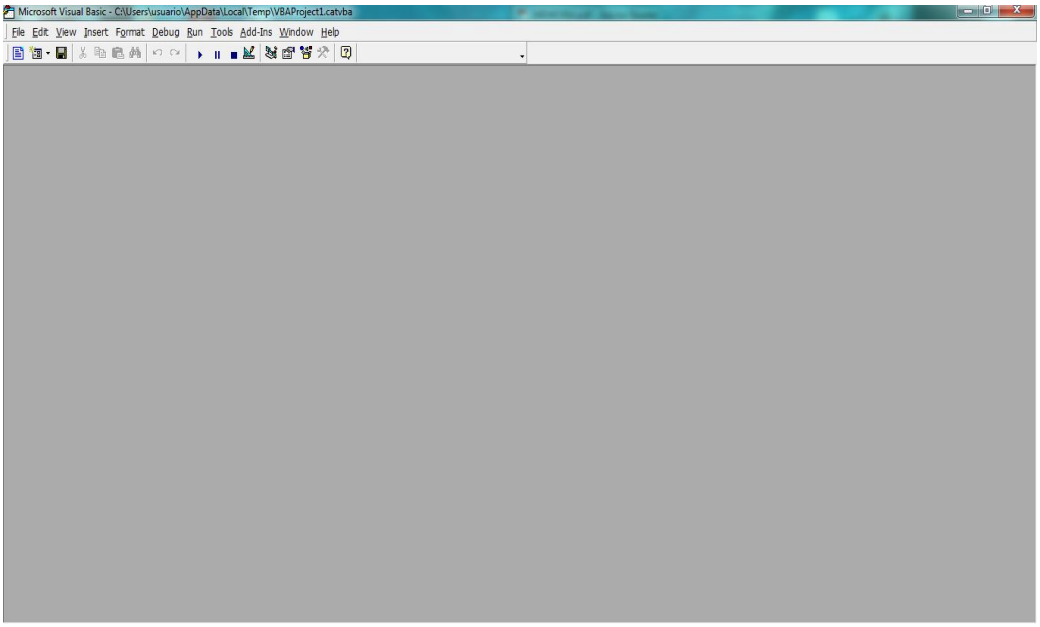

Figura 33. Entorno VBA.

Todo lo que se programe y se realice dentro de esta ventana puede interactuar directamente con CATIA si empleamos los objetos del programa. Se verá a continuación cada una de las partes o bloques que constituyen el entorno donde se realiza toda la programación.

Primero conviene tener a disposición del programador tanto el menú denominado *Project Explorer*, así como el *Properties Windows*, ya que son los dos menús fundamentales en los que se trabaja y los cuales facilitan mucho el trabajo. Para acceder a ellos, hay que clicar sobre la pestaña *View* y adjuntar ambas aplicaciones a la pantalla principal tal como se muestra en la Figura 35.

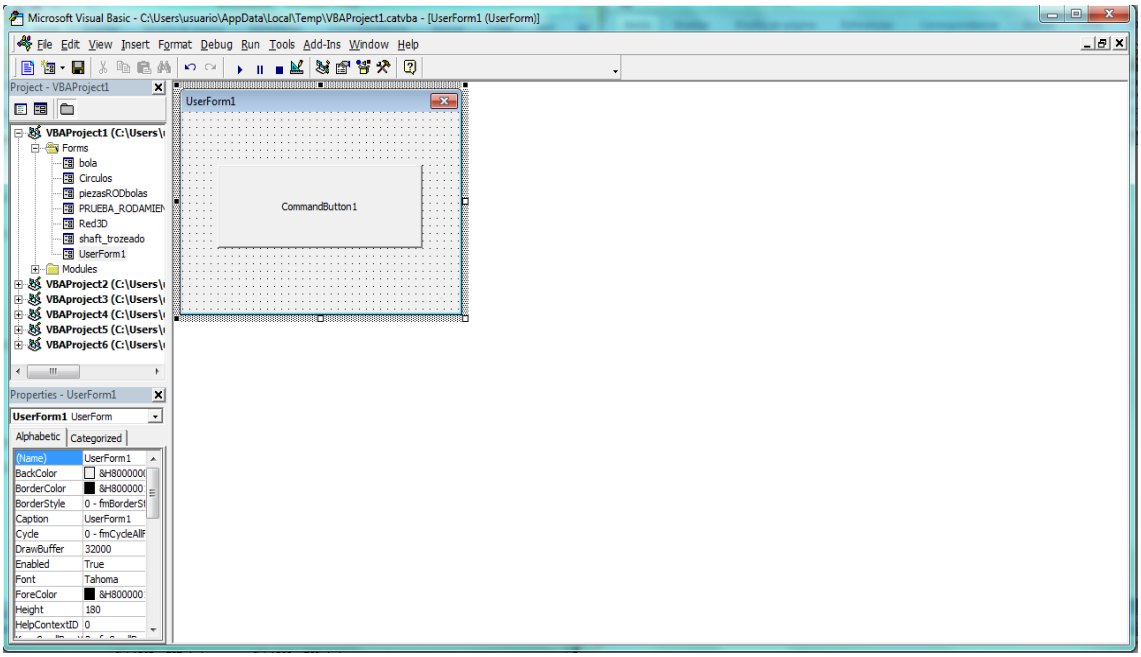

Figura 34. Project Explorer y Properties Window.

La ventana *Project Explorer* recoge todos los datos y archivos que conforman el *Project* (proyecto de programación) en el que se esté trabajando. Puede albergar formularios (*Userforms*), módulos (*Modele*) y clases (C*lass module*), los cuales se explican a continuación:

- Módulos: son rutinas independientes. Dado que el código contenido en un módulo estándar de *Visual Basic* es accesible desde distintos formularios del programa, será ventajoso colocar en este módulo todo lo que queramos disponer como "código compartido", es decir, que pueda ser utilizado en cualquier formulario del programa. Se guardan para su exportación en ficheros con extensión *.bas.
- Formularios: son rutinas asociadas a ventanas gráficas donde se incorporan objetos y eventos. Los formularios son el elemento básico que permite la interacción del programa con el usuario, demandando variables, opciones, etc. Se almacenan con extensión *.frm.
- Clases: son definiciones de nuevos objetos de tipo plantilla donde se definen las propiedades y eventos del mismo, son almacenados como *.cls.

En cuanto a la ventana *Properties Windows*, refleja en una columna todas las propiedades de cada formulario o control que se tenga seleccionado en la ventada

Project Explorer. Para acceder a dicha información solo tenemos que pinchar sobre el formulario o control deseado. *Referencia [10]*.

Por último, se va a explicar otra ventana que ha sido de utilidad, la ventana *object browser*, a la cual se accede de la misma forma que las anteriores.

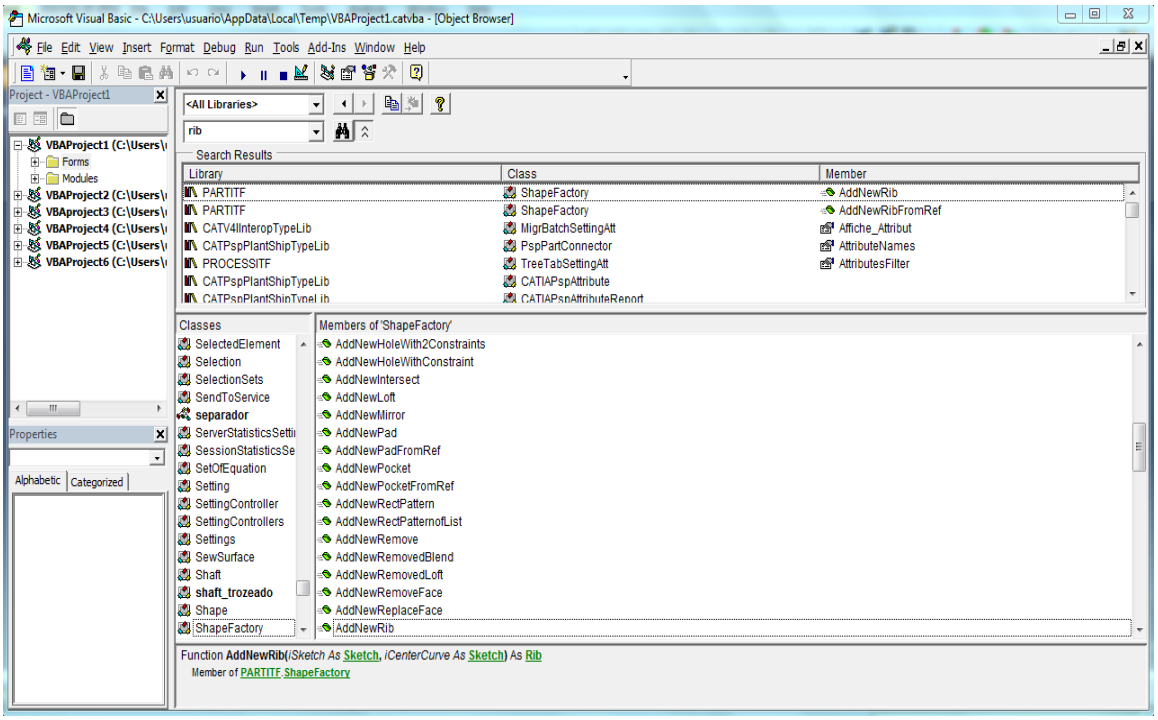

Figura 35. Object browser.

Dicha herramienta nos es de gran ayuda cuando se está en un punto de la programación en la que no se sabe bien qué hacer, cómo seguir o cómo funciona cualquier tipo de objeto. Si escribimos cualquier objeto en la barra de búsqueda, se nos muestran tres columnas que nos ayudan a controlar dicho objeto. La tercera de ellas, la columna *members*, muestra distintas formas para realizar la misma operación, eligiéndose la más conveniente. La segunda columna, muestra a qué tipo de objeto pertenece la operación que se quiere realizar, por lo que ya se sabrá que objeto hay que definir para poder acceder a la herramienta deseada. *Referencia [5]*.

Se puede observar que en la parte inferior de la imagen se muestra una descripción que nos sirve de ayuda ya que desglosa uno a uno todos los argumentos de entrada que son necesarios para realizar la operación. Para volver de nuevo a la ventana de trabajo no hay más que cerrar esta ventana.

2.4. Iniciación a la programación

Visual Basic es un lenguaje de programación por eventos, es decir, la ejecución del programa se produce en diferentes secciones debido a respuestas a eventos. Estos se producen por la interacción con el usuario mediante una interfaz gráfica que solicita al usuario los parámetros necesarios para la ejecución de cualquier acción o también debido a la existencia de otras aplicaciones que desencadenan dichos eventos.

Por eso, a la hora de programar, habrá que prestar especial atención a cómo se está escribiendo el código para que responda de forma adecuada a los eventos para los que se diseñe la aplicación.

A continuación se definen una serie de conceptos que es necesario conocer ya que en ellos se basa la programación controlada por eventos.

Concepto	Definición
Tiempo de diseño	Instante en que se crea la aplicación.
Tiempo de ejecución	Instante en el cual se ejecuta y se interactúa con la aplicación.
Formulario	Ventana sobre la que es posible personalizar la interfaz de la aplicación o cuadro de diálogo para obtener información del usuario.
Objetos	Formularios y controles.
Controles	Representación gráfica de objetos, con lo que el usuario interactúa y aporta la información que se le pide.
Propiedades	Los valores de un objeto (*properties window* antes mencionada). Son características de un objeto y definen el estado del mismo en un momento específico.
Métodos	Las acciones que un objeto puede realizar sobre sí mismo. Se suelen usar verbos para dar nombre a los métodos.
Eventos	Acciones que son reconocidas por un formulario o control. Los eventos ocurren a medida que el usuario interactúa con los objetos de la aplicación.
Colección	Grupo o lista de objetos similares que se ponen juntos por una razón específica. Las colecciones son objetos que agregan un conjunto de otros objetos.
Clases	Definen un tipo de objeto. Se suele usar la herencia para crear jerarquía entre clases y subclases.
Herencia	Todas aquellas clases que sean herencia de la misma clase tienen todas las propiedades y métodos en común de la herencia de la que provienen, pero también tienen sus propios métodos y propiedades que las diferencian entre ellas.

Tabla 3. Conceptos de la programación.

Referencia [10] y [11].

Otros conceptos que convienen reseñar antes de programar son los siguientes:

2.4.1. Declaración de estamentos

Antes de empezar a trabajar con una variable, constante o herramienta, es necesario nombrarla, obteniendo de esta forma una primera información acerca del estamento que posteriormente estableceremos.

Si no definimos el tipo de variable, VBA declarará la variable de tipo *variant*, la cual puede aceptar cualquier tipo de variable. Cabe destacar que en raras ocasiones se tiene una buena razón para usar una variable tipo *variant*. En la mayoría de los casos, tendremos que usar un determinado tipo de variable, haciendo de esta manera que el código se ejecute más rápido y reduciendo errores, lo cual se debe a que:

1. CATIA ejecutará el tipo de variable que se especifique.
2. A la hora de revisar el código, siempre se sabrá de qué tipo es cada variable y la intención por la cual se creó.

El comando que se usará para definir el tipo de variable con el que trabajaremos es *Dim*, como se muestra en el siguiente ejemplo:

Dim documents1 As documents.

Referencia[11].

2.4.2. Estamentos

Son una instrucción completa que puede contener *keywords* (*And, if, for, while, sub, function*…), operadores (+, - ,*, /,…), variables, constantes y expresiones. Un ejemplo de estamento sería:

Set partDocument1 = CATIA.ActiveDocument

Donde el comando *Set* se usa para asociar la variable definida con un objeto. *Referencia [10]*.

2.4.3. Estamentos ejecutables

Se trata de acciones iniciales como:

Select1.Search("name='Optimization.MinimumValue',all"). Referencia [10].

2.4.4. Funciones y subfunciones

Son una secuencia de estados que conforman la operación deseada. Esta operación viene especificada en una función.

Sub mySubwithParameter (myParameter)

MsgBox myParameter

End Sub

La diferencia entre *function* y sub es que mientras que la primera te devuelve un valor la segunda no. *Referencia [10]*.

2.4.5. Estructuras condicionales e iterativas

1. Condicionales: Visto que durante la programación, muchas veces se llega a situaciones en que es conveniente utilizar condicionales para comprobar, por ejemplo, que un valor se encuentra en un intervalo aceptable, es conveniente hacer mención de la ejecución: *if…then…*

La estructura es la siguiente:

If [condición] Then
[Estamento]
ElseIf [Condición] Then
[Estamento de ElseIf]
Else
[Estamento de Else]
End If

2. Iterativas: Otra de las estructuras que se usan con mucha frecuencia en la programación son aquellas en las que se realiza una misma operación varias veces consecutivas con el fin de encontrar un valor justo o que haga funcionar un programa hasta que cierta condición se cumpla. Se puede hacer esto de distintas formas:

- *For… Next:* cuando queremos iterar un número de cosas dado.

Contador

For[contador] = [inicio] To [end] {paso a paso}

[Estamentos]

Next

- *While… Wend:* cuando iteramos hasta que el contador cumple cierta condición.

Contador

While [{Contador} Condición]

[Estamentos]

Wend

- *Do…Loop:* cuando iteramos siempre que se cumpla cierta condición.

Do [{While|Until} condition]

[Estamentos]

[Exit Do]

[Estamentos]

Loop

Do [Estamentos]

[Exit Do]

[Estamentos]

Loop [{While|Unitl} condition]

- For *each… Next:* cuando se quiere iterar sobre los objetos en una determinada colección. *Referencia [10]*.

2.4.6. Objetos orientados a la programación

Los objetos son una parte de memoria en la que está contenida cierta información y metodología que permite operar con dicha información almacenada.

Requiere de un cierto código especial para hacer que funcione, ya que dispone de comportamientos definidos por métodos específicos tales como:

1) Pasar información a través de parámetros.
2) Operaciones de cálculo que pueden:
    a. Cambiar cierta parte de los datos iniciales.
    b. Diseñar ciertas operaciones que no están por defecto.
    c. Devolver valores de los datos iniciales.
    d. Devolver resultados de los cálculos usando tanto la información introducida externamente como de la contenida en el objeto.

2.4.7. Cómo definir un objeto

Para cada objeto en particular se define una clase (*classe*) que sirve como plantilla en la que se recoge cómo operan los objetos y los datos que contienen. Es conveniente resaltar que una clase puede ser utilizada para hacer funcionar uno o más objetos. *Referencia [10]*.

Clase	Objeto
Describe la estructura del objeto.	Es el resultado de la clase.
Es una plantilla	Tiene una copia de toda variable no-estática pero no de las variables tipo clase (*static*).
Especifica la representación de la información, el comportamiento, la interrelación (vía variables, métodos y *parents*-estructura lógica)	

Tabla 4. Diferencias entre clase y objeto.

2.5. Interfaz con el usuario

Se explica a continuación cómo podemos crear nuestra propia interfaz para interactuar con el programa y así llegar a la programación por eventos que vimos anteriormente.

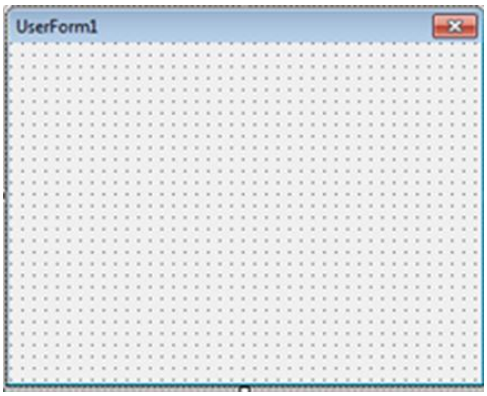

Figura 36. Formulario de trabajo.

Para ello tenemos que crear un nuevo formulario sobre el que se diseñará la interfaz añadiendo los distintos controles necesarios para que el usuario interactúe con la aplicación. Para insertar un formulario abrimos la pestaña *insert* → *Userform.*

Sólo falta añadir los elementos visuales que permitan al usuario controlar el programa. Para ello activaremos la ventana *"Toolbox"*, lo cual haremos desde la pestaña *view* → *Toolbox*.

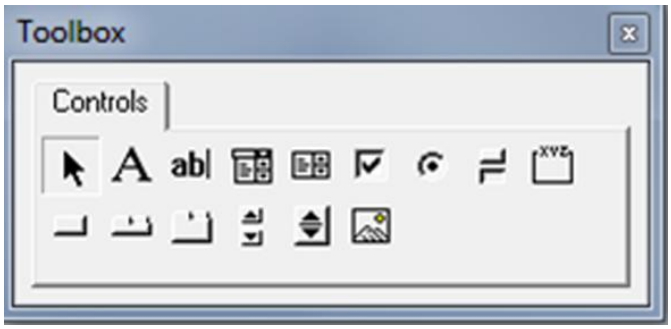

Figura 37. Herramientas de diseño.

A continuación se van a explicar los controles de los que se disponen para diseñar el formulario. *Referencia [10]*.

Símbolo	Etiqueta	Descripción
	Label	Permite escribir títulos o comentarios.
	TextBox	Permite al usuario introducir texto.
	ListBox	Control en el que se muestran varios registros, pudiendo seleccionar uno o más de uno.
	ComboBox	Control parecido al ListBox con una propiedad llamada *Style*, que permite 3 formas distintas de presentar una lista.
	CheckBox	Permite seleccionar una opción al usuario.
	OptionButton	Permite seleccionar una opción al usuario.
	ToggleButton	Botón para selección de opciones.
	Frame	Agrupar diferentes objetos referidos a un mismo tema.
	CommandButton	Permite ejecutar un evento.
	TabStrip	Separadores o etiquetas.
	MultiPage	Contenedor para una colección de objetos.

	*ScrollBar*	Permite tener una barra para desplazamientos.
	*SpinButton*	Permite aumentar o disminuir la cifra conforme se presionan las flechas del control.
	*Image*	Insertar una imagen en el formulario.

Tabla 5. Opciones de la barra de herramientas.

La interfaz por sí sola no realiza ninguna operación por lo que habrá que introducir un código y programar cada uno de ellos. Para añadirlos sólo hay que hacer doble clic sobre el mismo formulario y se abrirá una ventana como la de la figura 39, la cual se corresponde con un *CommandButton*.

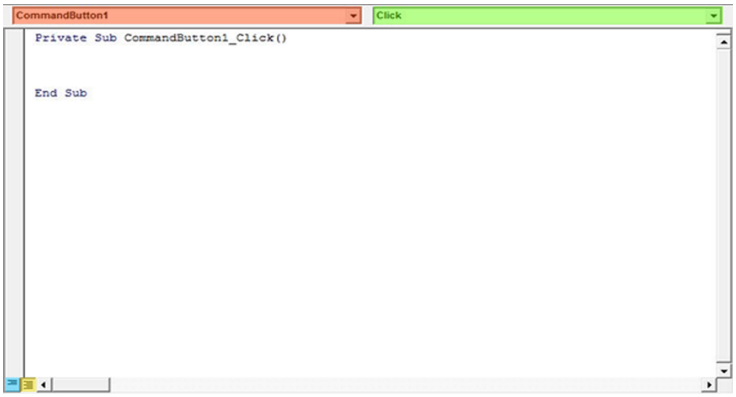

Figura 38. Editor.

Antes que nada, conviene describir la ventana de código adjunta. En ella se distinguen dos pestañas, una roja que recoge la lista de objetos disponibles para programar y una verde que indica para el objeto seleccionado, los eventos disponibles. Y finalmente en la parte inferior se tienen los botones para ver el procedimiento y ver el módulo.

En los siguientes capítulos se explicará cómo funcionan algunos de los controles que se han mostrado en esta sección.

Finalmente, cuando tengamos la rutina diseñada, el siguiente paso será probarla. Para ello se dispone de una serie de opciones correspondientes con el menú *Run*, también disponibles en el menú horizontal.

Figura 39. Comandos para hacer funcionar la aplicación.

Cuando seleccionamos el botón *Play*, el programa se traslada al entorno de CATIA y aparece en pantalla la aplicación creada.

3. Herramientas de programación de rodamientos.

3.1. Introducción

A continuación se van a presentar todos los procedimientos que se han llevado a cabo para realizar los rodamientos que se describieron en el apartado "6.9. Descripción de la geometría de los rodamientos que se van a realizar".

Se trabajará con tres módulos, el módulo *"sketcher",* para realizar los diferentes *sketches* necesarios, el módulo *"Part Design",* para dar dimensión en 3D a los *sketches* y finalmente el módulo *"Assembly Design"* para unir las piezas o *Parts* que se han creado con los módulos anteriores.

3.2. Arranque

Se comienza explicando cómo se define y establece el entorno de trabajo sin necesidad de tener abierto el *part*, lo cual se hará activando algunos parámetros que permitan al programador responder a las acciones que se le indican.

```
Dim documents1 As Documents
Dim partDocument1 As PartDocument
Dim part1 As Part
Set documents1 = CATIA.Documents
Set partDocument1 = documents1.Add("Part")
Set part1 = partDocument1.Part
```

Código 1. Arranque de documentos.

El *PartDocument* activa el módulo *Part Design*, donde se definen el resto de elementos que componen el proyecto en el que se está trabajando. También se define el archivo *part* donde se creará la pieza, el cual se establece en el *PartDocument* definido anteriormente.

Una vez definido el *part*, es necesario crear el árbol de trabajo dónde se reflejarán todas las operaciones que se vayan realizando durante el proyecto, estableciendo el *body* de trabajo. Para ello, establecemos los *bodies*, que son como los hijos del *part*, por lo que primero habrá que definir el *bodies1* y dentro de estos uno específico *body1*.

Se puede observar la jerarquización de la estructura donde cada archivo está activado dentro de otro de manera que si alguno falla, otros no podrán realizar sus

funciones ya que se produce el fallo global del programa.

```
Dim bodies1 As Bodies
Dim body1 As Body
Set bodies1 = part1.Bodies
Set body1 = bodies1.Item("PartBody")
```

Código 2. Bodies.

Al igual que hacemos cuando diseñamos manualmente cualquier pieza, lo primero que se va a realizar es un *sketch* donde poder dibujar. Dicho *sketch* se genera dentro del *body* definido anteriormente. A continuación hay que establecer el sistema de referencia 3D con el que se va a trabajar y además definir en qué plano de este sistema se va a hacer. Finalmente, colocamos el *sketch* en el que se dibujará dentro del plano de referencia definido anteriormente.

```
Dim sketches1 As Sketches
Set sketches1 = body1.Sketches
Dim originElements1 As OriginElements
Set originElements1 = part1.OriginElements
Dim reference1 As Reference
Set reference1 = originElements1.PlaneYZ
Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)
```

Código 3. Sketches.

Para este código se ha cogido al azar el plano YZ y se ha guardado dentro de una referencia que hemos llamado *reference1*.

Con el *sketch* y el plano de trabajo creados, lo siguiente es establecer una matriz que defina las direcciones dentro de cada plano, de forma que el programa tenga claro en qué plano de trabajo se encuentra.

```
Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0# 'Vector unidad x en el plano X
arrayOfVariantOfDouble1(1) = 0# 'Vector unidad y en el plano X
arrayOfVariantOfDouble1(2) = 0# 'Vector unidad z en el plano X
arrayOfVariantOfDouble1(3) = 0# 'Vector unidad x en el plano Y
arrayOfVariantOfDouble1(4) = 1# 'Vector unidad y en el plano Y
arrayOfVariantOfDouble1(5) = 0# 'Vector unidad z en el plano Y
arrayOfVariantOfDouble1(6) = 0# 'Vector unidad x en el plano Z
arrayOfVariantOfDouble1(7) = 0# 'Vector unidad y en el plano Z
arrayOfVariantOfDouble1(8) = 1# 'Vector unidad z en el plano Z
Set sketch1Variant = sketch1
sketch1Variant.SetAbsoluteAxisData arrayOfVariantOfDouble1
'Establece el sistema de ejes absoluto del sketch en 3D
```

Código 4. Vector de coordenadas.

Finalmente se escribe esta línea de código informando que el *sketch1* es el lugar de trabajo a partir de ese punto y las operaciones que se realicen serán sobre el mismo.

```
part1.InWorkObject = sketch1
```

A continuación se definen y establecen el conjunto de herramientas para realizar el dibujo 2D y todos los elementos geométricos necesarios .

```
' Establecimiento del conjunto de herramientas
' 2D y asignarlo al sketch de trabajo
Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()

'Se establecen los elementos geométricos
Dim geometricElements1 As GeometricElements
Set geometricElements1 = sketch1.GeometricElements

'Se define el sistema de ejes dentro del sketch
Dim axis2D1 As Axis2D
Set axis2D1 = geometricElements1.Item("AbsoluteAxis")

'Establecimiento de las direcciones horizontal y vertical
Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2

'Se cierra el sketch y se establece como objeto de trabajo
sketch1.CloseEdition
part1.InWorkObject = sketch1
part1.Update
```

Código 5. Objeto factory 2D.

Las últimas líneas del código 5, cierran el *sketch* en cuestión, lo establece como objeto de trabajo, es decir, sobre el que se realizarán las operaciones para el modelado de un sólido, y lo carga en el *part*. *Referencia [10]*.

3.3. SKETCHER

Dentro de este módulo se puede encontrar una serie de objetos y colecciones que resultarán de gran utilidad en este proyecto.

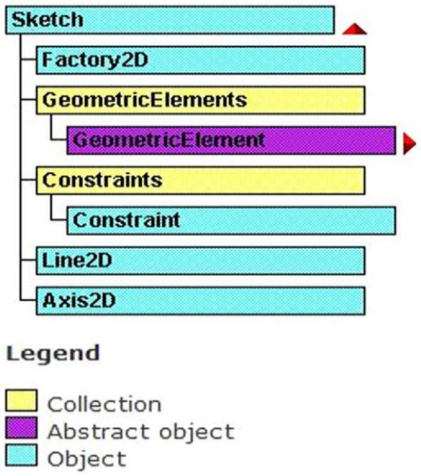

Figura 40. Estructura del módulo Sketcher.

Como se puede observar, los objetos que se encuentran en dicho módulo son *"Factory2D", "line2D"* y *"Axis2D",* dentro de los cuales se encuentran una serie de métodos que se resumen a continuación:

Objeto	Comentario	Contenido
Factory2D	Incluye todos los métodos necesarios para poder trabajar en el módulo Sketcher.	CreateCircle
		CreateClosedCircle
		CreateClosedEllipse
		CreateControlPoint
		CreateEllipse
		CreateHyperbola
		CreateIntersection
		CreateIntersections
		CreateLine
		CreateLineFromVector
		CreateParabola
		CreatePoint
		CreateProjection
		CreateProjections
		CreateSpline
Line2D	Engloba tres métodos que dan el univector de la dirección de la recta, un punto sobre la línea y un último que permite modificar las características de la línea infinita.	GetDirection
		GetOrigine
		SetData
Axis2D	Vienen detalladas las propiedades del sistema de coordenadas	HorizontalReference
		Origin
		VerticalReference

Tabla 6. Objetos del módulo *Sketcher*.

Una vez vista la estructura del módulo *sketcher*, se va a describir cómo usar los diferentes métodos que componen los principales objetos y colecciones de dicho módulo para diseñar la geometría de los diferentes perfiles en cuestión. *Referencia [10]*.

3.3.1. Crear un punto

Como se observa en la tabla 6, el objeto *factory2D* del módulo *sketcher* posee herramientas para crear una amplia gama de geometrías. Se va a describir a continuación cómo crear un punto. Para ello, se usará la siguiente línea de código:

```
Dim point2D1 As Point2D
Set point2D1 = factory2D1.CreatePoint(coord_x, coord_y)
point2D1.ReportName = 3
point2D1.Construction = False
```

Código 6. Creación de un punto.

Como se puede apreciar, solo es necesario introducir las coordenadas del punto que se desea crear. A este punto se le puede asignar un nombre para que sea más fácil de localizar en el *sketch*, así como imponer que dicho punto no sea de construcción, que es como CATIA lo crea por defecto.

3.3.2. Crear una recta

Otra herramienta que se encuentra también en el objeto *factory2D* es la que permite crear una recta a partir de dos puntos, el de origen y el punto final, previamente creados. El código necesario para esto es el que sigue:

```
Dim line2D1 As Line2D
Set line2D1 = factory2D1.CreateLine(x1, y1, x2, y2)
line2D1.ReportName = 3
line2D1.StartPoint = point2D1
line2D1.EndPoint = point2D2
```

Código 7. Creación de una recta.

Los puntos de origen y final *(Point2D1, Point2D2)* se crean cómo se ha mostrado en el apartado anterior, y para que la recta quede bien definida, es necesario reseñar que dichos puntos sean el de origen y fin, de forma que el programa entienda que se trata de hacer una sola recta y no continúe con otra después.

3.3.3. Crear un círculo o arco de circunferencia

Para definir un círculo cerrado necesitamos tres variables: la coordenada x e y

del punto que será el centro de la circunferencia, y el radio de la misma.

Para ello es necesario crear primero el punto de origen, *point2D1*, con la orden *factory2D1.CreatePoint()* en las coordenadas que corresponda. Se usará la orden *factory2D1.CreateClasedCircle()* para generar el círculo. Como se puede apreciar, el definir el punto sirve para que a través de *circle2D1.CenterPoint* el programa centre la circunferencia en dichas coordenadas. A continuación se muestran las líneas de código con las que se crea un círculo cerrado.

```
Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateClosedCircle(x1, y1, r1)
circle2D1.CenterPoint = point2D1
circle2D1.ReportName = 4
```

Código 8. Creación de un círculo.

De igual manera, es muy útil saber crear arcos de circunferencias ya que es uno de los elementos que más se usan a la hora de realizar piezas con zonas curvas. Para ello se usa el comando *factory2D1.CreateCircle(x, y, r, start, finish),* donde las variables que hay que introducir son:

- x: coordenada horizontal del centro del arco de la circunferencia.
- y: coordenada vertical del centro del arco de la circunferencia.
- r: radio del arco de la circunferencia.
- *start*: ángulo en radianes del punto donde comienza el arco.
- *finish*: ángulo en radianes del punto donde finaliza el arco.

El siguiente código muestra la estructura para crear un arco de circunferencia.

```
Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateCircle(x, y, r, ang_inicio [radianes], ang_fin [radianes])
```

Código 9. Creación de un arco de circunferencia.

3.3.4. Crear un spline

Es posible crear curvas a partir de puntos de la misma, es decir, mediante una aproximación polinómica a partir de distintos puntos que las conforman. Para ello el objeto *factory2D* presenta una herramienta llamada *"control Point"*.

Para originar dichos puntos de control se usa el comando *createcontrolpoint(x,y)* y es necesario crear un vector que incluya todos los *controlpoints* que conforman la curva que queremos desarrollar. Se usará el siguiente código para programar un *spline*.

```
Dim controlPoint2D1 As ControlPoint2D
Set controlPoint2D1 = factory2D1.CreateControPoint(x1, y1)
controlPoint2D1.ReportName=3
Dim controlPoint2D2 As ControlPoint2D
Set controlPoint2D2 = factory2D1.CreateControPoint(x2, y2)
controlPoint2D2.ReportName=4

Dim controlPoint2D3 As ControlPoint2D
Set controlPoint2D3 = factory2D1.CreateControPoint(x3, y3)
controlPoint2D3.ReportName=5

Dim controlPoint2D4 As ControlPoint2D
Set controlPoint2D4 = factory2D1.CreateControPoint(x4, y4)
controlPoint2D4.ReportName=6

Dim arrayOfObject1(3)
Set arrayOfObject1(0) = controlPoint2D1
Set arrayOfObject1(1) = controlPoint2D2
Set arrayOfObject1(2) = controlPoint2D3
Set arrayOfObject1(3) = controlPoint2D4

Dim spline2D1 As Spline2D
Set factory2D1temp=factory2D1
Set spline2D1 = factory2D1temp.CreateSpline(arrayOfObject1)
spline2D1.ReportName = 7
```

Código 10. Creación de un spline.

Como se puede apreciar, el único argumento de entrada que precisa para el correcto funcionamiento de dicha herramienta es el vector objetos que en este caso son los *controlpoints*.

3.4. Restricciones

Para que el *sketch* quede bien definido y fijo, es decir, que sea inalterable por algún error cuando se esté creando o manipulando, es necesario establecer las restricciones o *constraints*.

Los *constraints* trabajan con referencias, es decir, hay que establecer la referencia de cada uno de los objetos del *sketch*. Una vez definidas éstas, es necesario una instrucción que permita relacionarlas entre sí, para lo cual se usa la siguiente línea de código: *constraintsX.AddBiEltCs(CatCsTypeDistance, reference1, reference2)*. En dicho comando se observan dos propiedades de gran interés.

La instrucción *AddBiEltCs* nos indica que la restricción va a usar dos referencias, es decir, se van a relacionar dos objetos. Se pueden relacionar uno, dos o tres objetos. Para ello solo se ha de cambiar el prefijo "Bi-" por el correspondiente. Es decir, una restricción tal que *AddMonoEltCs* solo necesitará una referencia y otra como *AddTriEltCs*, tres. En este proyecto se trabajará con restricciones de uno y dos objetos.

CatCsTypeDistance refleja el tipo de restricción que se quiere imponer. Existen numerosos tipos de restricciones como se muestra en la tabla 7.

Con estas órdenes, el programa sabe que las referencias en cuestión se encuentran relacionadas según el tipo de restricción utilizada. Sin embargo no conoce la cantidad exacta del parámetro (el cual dependerá del tipo de restricción en cuestión) que las relaciona (distancia, ángulo…), por lo que hay que señalar que *constraint* tiene dos modos de trabajo:

1. Modo *constraint:* El valor asignado restringe la geometría en dicha posición >>*constraint1.Mode = catCstMode**Driving**Dimension*
2. Modo *Measurement:* el valor solo refleja aquello que puede ser observado desde dicha posición
   >> *constraint1.Mode = catCstMode**Driven**Dimension*

Número de referencias	Tipo
BiEltCs	*CatCstTypeAnnulContact*
	CatCstTypeParallelims
	CatCstTypePerpendicularity
	CatCstTypeChamfer
	CatCstTypeConcentry
	CatCstTypeDistance
	CatCstTypeHorizontaly/Vertically
	CatCstTypeLength
	CatCstTypeLineContact
	CatCstTypeMajor/MinorRadius
	CatCstTypeMidpoint
	CatCstTypeOn
	CatCstTypePlanarangle
	CatCstTypeDistance
	CatCstTypeTangency
MonoEltCs	*CatCstTypeRadius*
TriEltCs	*CatCstTypeSimetry*

Tabla 7. Restricciones.

A continuación se van a explicar las restricciones más empleadas en el

desarrollo de este proyecto. *Referencia [10]*.

3.4.1. CatCstTypeDistance

Esta restricción se usa para fijar la posición respecto al origen de coordenadas o la distancia entre dos objetos. También es posible utilizarlo para determinar la longitud de algunos objetos.

Además de los comandos introducidos al inicio de la subsección, es necesario declarar el valor de una longitud que será definida como la dimensión de la restricción.

```
'Establecimiento de las constraints
Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints
'constraint de la distancia horizontal de un punto al origen de coordenadas.
Dim reference7 As Reference
Set reference7 = part1.CreateReferenceFromObject(point2D3)
Dim reference8 As Reference
Set reference8 = part1.CreateReferenceFromObject(line2D2)
Dim constraint4 As Constraint
Set constraint4 = constraints1.AddBiEltCst(catCstTypeDistance, reference7, reference8)
constraint4.Mode = catCstModeDrivingDimension
Dim length2 As Length
Set length2 = constraint4.Dimension
length2.Value = x1
```

Código 11. *CatCstTypeDistance*.

3.4.2. CatCstTypeRadius

La utilidad de esta restricción es poder fijar el radio de circunferencias y arcos de circunferencias. De nuevo habrá que definir la longitud y darle el valor del radio.

```
'establecimiento de los constraints
Dim reference25 As Reference
Set reference25 = part1.CreateReferenceFromObject(circle2D1)
Dim constraint13 As Constraint
Set constraint13 = constraints1.AddMonoEltCst(catCstTypeRadius, reference25)
constraint13.Mode = catCstModeDrivingDimension
Dim length8 As Length
Set length8 = constraint13.Dimension
length8.Value = r1
```

Código 12. *CatCstTypeRadius*.

3.4.3. CatCsTypeHorizontally/Vertically

Cuando queremos crear una recta sea horizontal o vertical y no conocemos los puntos de inicio y final ni la longitud de esta, es muy útil definir una recta arbitraria e

imponer esta restricción de horizontalidad o verticalidad.

```
Dim reference22 As Reference
Set reference22 = part1.CreateReferenceFromObject(line2D8)
Dim reference23 As Reference
Set reference23 = part1.CreateReferenceFromObject(line2D2)
Dim constraint12 As Constraint
Set constraint12 = constraints1.AddBiEltCst(catCstTypeVerticality, reference22, reference23)
constraint12.Mode = catCstModeDrivingDimension
```

Código 13. *CatCstTypeHorizontally/Vertically.*

3.4.4. CatCstTypeLength

Por el contrario, si lo que queremos fijar no es la orientación de la recta sino la longitud de esta, se puede emplear este *constraint*. Para ello habrá que crear una longitud y darle valor a esta.

```
Dim reference19 As Reference
Set reference19 = part1.CreateReferenceFromObject(line2D6)
Dim constraint10 As Constraint
Set constraint10 = constraints1.AddMonoEltCst(catCstTypeLength, reference19)
constraint10.Mode = catCstModeDrivingDimension
Dim length7 As Length
Set length7 = constraint10.Dimension
length7.Value = d
```

Código 14. *CatCstTypeLength.*

3.4.5. CatCstTypeParallelism

Para fijar que una recta sea paralela a otra, se definen las dos rectas como referencias y se aplica la restricción de paralelismo a las dos referencias creadas.

```
Dim reference20 As Reference
Set reference20 = part1.CreateReferenceFromObject(line2D7)
Dim reference21 As Reference
Set reference21 = part1.CreateReferenceFromObject(line2D5)
Dim constraint11 As Constraint
Set constraint11 = constraints1.AddBiEltCst(catCstTypeParallelism, reference20, reference21)
constraint11.Mode = catCstModeDrivingDimension
```

Código 15. *CatCstTypeParallelism.*

3.4.6. CatCstTypeOn

La restricción *CatCstTypeon* es una de las más interesantes de las restricciones existentes.

Dicha restricción permite imponer la coincidencia de dos puntos o curvas cualesquiera, por lo que permite introducir imposiciones geométricas muy importantes. En el código que se muestra a continuación, se está creando una coincidencia entre los ejes de dos piezas cilíndricas. Como se puede apreciar, primero se hacen referencias a los ejes de las piezas y luego se realiza la restricción de coincidencia.

```
        Dim reference2 As Reference
        Set reference2 =
        product1.CreateReferenceFromName("Product1/Part1.2/!Axis:(Selection_
RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;6)));None:();Cf11:());E
dgeFillet.1_ResultOUT;Z0;G3055))")

        Dim reference3 As Reference
        Set reference3 =
        product1.CreateReferenceFromName("Product1/Part1.1/!Axis:(Selection_
RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;18)));None:();Cf11:());
EdgeFillet.2_ResultOUT;Z0;G3055))")

        Dim constraint2 As Constraint
        Set constraint2 = constraints1.AddBiEltCst(catCstTypeOn, reference2,
reference3)
```

Código 16. *CatCstTypeOn*.

4. CATPART

En esta sección de CATIA, denominada CATPART, vienen englobados tres módulos:

1. *PartDesign.*
2. *Wireframe and SurfaceDesign.*
3. *GenerativeShapeDesign.*

De estos módulos solo se usará en el presente proyecto el *PartDesign* para generar modelos sólidos. El *Wireframe and SurfaceDesign* sirve para generar superficies. Se va a estudiar a continuación el primer módulo.

4.1. PART DESIGN

Este módulo se centra en otra sección de CATIA en el que se dispone de una gran variedad de objetos que permiten tener a mano toda la información que es necesaria para generar un modelo sólido.

Figura 41. Estructura del módulo *Part Design.*

La estructura del *Part Design* es la que se aprecia en la figura 42, donde los colores significan lo mismo que se ha explicado en el módulo previo.

Como ya se ha comentado anteriormente, la estructura de CATIA está muy jerarquizada, de manera que por ejemplo, para acceder a los planos de referencia, habrá que definir en primer lugar los objetos y propiedades referentes a *Plane* que contienen dichos planos *(PartDocument → Part → Originelements).*

Figura 42. Estructura interna del *PartDocument.*

En la tabla siguiente se recogen las propiedades de los objetos que se utilizan en el módulo *PartDesign*, explicando brevemente qué función tiene cada una de ellas.

Propiedad	Comando	Comentario
OriginElements	Dim origin1 As OriginElements Set origin1 = part1.OriginElements	Constituye el sistema de referencia absoluto del documento de planos XY, XZ e YZ.
AxisSystems	Dim refsist1 As AxisSystems Set refsist1 = part1. AxisSystems	Es la colección de sistemas de referencia que pueden existir en el documento *part.*
GeometricElements	Dim geometric1 As GeometricElements Set geometric1 = Part1.GeometricElements	Colección de elementos geométricos 3D del *partdocument* generadas directamente en 3D, es decir, sin la mediación del módulo *Sketcher*
Bodies	Dim Bodies1 As Bodies Set Bodies1 = Part1.Bodies	Colección de cuerpos sólidos dentro del *partdocument*
HybridBodies	Dim HybBod1 As HybridBodies Set HybBod1 = Part1.HybridBodies	Colección de *OpenBodies* (elementos de referencia)

Constraints	Dim Constraints1 As Constraints Set Constraints1 = Part1.Constraints	Colección de restricciones geométricas y dimensionales del *partdocument*.
Relations	DimRelations1 AsRelations SetRelations 1 = Part1.Relations	Colección de relaciones del *partdocument*.
Parameters	DimParameters1 AsParameters SetParameters1 = Part1.Parameters	Colección completa de todos los parámetros del *partdocument.*

Tabla 8. Objetos del *PartDesign*.

Referencia [10].

Se va a explicar ahora el objeto *"shapefactory"*, que contiene todas las operaciones que permiten pasar de un dibujo 2D a un sólido 3D, o que una vez obtenido este, realizan transformaciones sobre el mismo. Este objeto presenta una amplia gama de operaciones que podemos encontrar en la ventana *object browser* pero solo se van a incluir las más usuales y que se usaran durante el proyecto.

Objeto	Comentario	Contenido
Shapefactory	Incluye todos los métodos necesarios para poder trabajar en el módulo *Part Design*.	*AddnewCircPattern*
		AddnewHole
		AddnewPad
		AddnewPocket
		AddnewSlot
		AddnewRib
		AddnewEdgeFilletWith ConstantRadius
		AddnewShaft

Tabla 9. Herramientas del *Shapefactory*.

Ahora se van a mostrar en mayor detalle estas herramientas que conforman el *shapefactory* y que hemos usado para realizar los rodamientos. Lo primero que hay que hacer es definir y cargar el objeto en cuestión y una vez cargado ya se podrá hacer uso de las herramientas que incluye. *Reference [10]*.

```
Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory
```

Código 17. Definición del objeto *shapefactory*.

4.1.1. Pad

La herramienta *pad* es la comúnmente usada para hacer extrusiones a partir de

un *sketch*. Es muy potente ya que con dicha herramienta y con alguna más que explicaremos posteriormente, se pueden definir infinidad de sólidos. Usaremos las siguientes líneas de código para hacer un *pad*:

```
Dim pad1 As Pad
Set pad1 = shapeFactory1.AddNewPad(sketch1, espesor)
pad1.Name = "nombre1" ' Darle un nombre al pad
Dim limit1 As Limit
Set limit1 = pad1.FirstLimit
Dim length1 As Length
Set length1 = limit1.Dimension
length1.Value = espesor
```

Código 18. *Pad*.

Los argumentos que nos pide la herramienta son el *sketch* que se desea extruir y el espesor que se quiere. Otra reseña de importancia es la posibilidad de llamar al *pad* con el nombre que elijamos para luego no confundirlo con otro posible *pad*.

Es necesario restringir la altura del *pad*. Para ello hay que definir tanto el límite inferior del que tiene que partir el *pad* como la longitud de la extrusión. Habrá que darle un valor a la dimensión de la longitud. *Reference [10]*.

4.1.2. Pocket

La estructura del *pocket* es prácticamente igual a la del *pad* pero en este caso, lo que se quiere hacer es un hueco sobre un sólido ya creado, de manera que, tendremos que hacer una referencia a la superficie sobre la que se quiere hacer dicho hueco.

La referencia se puede hacer de dos formas:

1. Creando una referencia directa de la superficie del sólido usando las líneas mostradas en el código 18. Se observa que el *pocket* se hará sobre una superficie *RSur:Face*, especificando que se realiza sobre el *pad* ya creado.

```
Dim reference10 As Reference
Set reference10 =
 part1.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();~
~ Cf11:());Pad.1_ResultOUT;Z0;G3055)")
```

Código 19. Referencia del *pad* 1.

2. O bien creando un plano con un offset que coincida con la distancia a la superficie del sólido sobre la que se quiere generar el *pocket*. Para ello se

crea un plano, herramienta que se encuentra dentro del objeto *hybridshapefactory*, en el cual se encuentran las principales herramientas del módulo *wireframeandSurfaceDesign*.

```
Dim hybridShapePlaneExplicit1 As HybridShapePlaneExplicit
Set hybridShapePlaneExplicit1 = originElements1.PlaneXY
Dim reference10 As Reference
Set reference10 = part1.CreateReferenceFromObject(hybridShapePlaneExplicit1)
Dim hybridShapePlaneOffset1 As HybridShapePlaneOffset
Set hybridShapePlaneOffset1 = hshapefactory1.AddNewOffset(reference10, altura,
False)
```

Código 20. Referencia del *pad* 2.

El procedimiento seguido consiste en crear un plano XY, que servirá de referencia a nuestro plano, indicando que son paralelos.

Cuando ya tenemos la referencia para hacer el *pocket*, el resto del proceso se realiza de manera análoga al caso del *pad*, usando el primero de los casos explicados anteriormente:

```
Dim reference10 As Reference
Set reference10 =
 part1.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();~
~ Cf11:());Pad.1_ResultOUT;Z0;G3055)")

Dim pocket1 As Pocket
Set pocket1 = shapeFactory1.AddNewPocket(sketch2, espesor)
pocket1.Name = "hueco interno"' Dar nombre al agujero

Dim limit2 As Limit
Set limit2 = pocket1.FirstLimit
limit2.LimitMode = catUpToSurfaceLimit ' Se define su profundidad a la
' superficie más próxima
```

Código 21. *Pocket.*

Reference [10].

4.1.3. Hole

A diferencia del *pocket*, cuando hacemos un *hole*, estamos haciendo un orificio circular mientras que con el *pocket* se puede hacer un hueco con la forma del *sketch* que demos como entrada. Al igual que hay que hacer para el *pocket*, es necesario crear una referencia de la superficie del sólido sobre la que se realizará el *hole* para lo que habrá que usar las siguientes líneas de código.

```
Dim reference3 As Reference
```

```
Set reference3 =part1.CreateReferenceFromBRepName("FSur:~
~Face:(Brp:(cilindro;2);None:();Cf11:());WithTemporaryBody;~
~WithoutBuildError;WithInitialFeatureSupport;MonoFond;~
~MFBRepVersion_CXR15)", pad1)
```

Código 22. Referencia para un *hole*.

Se dispone de varias herramientas para programar un *hole*, cada una de las cuales nos piden distintos argumentos, y se usarán unas u otras según el caso. Algunas de las distintas formas de programarlo se muestran a continuación:

Orden	Parámetros
AddNewHole	X, Y,Z (coordenadas absolutas), profundidad
AddNewHoleFromPoint	X,Y,Z (coordenadas absolutas), plano de referencia, profundidad
AddNewHoleFromRefPoint	Punto origen, plano de referencia, profundidad
AddNewHoleFromSketch	Sketch, profundidad
AddNewHoleWithConstraints	X,Y,Z (coordenadas absolutas), contorno, plano de referencia, profundidad. (si el contorno es circular, el *hole* será concéntrico a dicho contorno)

Tabla 10. Herramientas para programar un *hole*.

A continuación se explica cómo programar un *hole* a partir de un punto, donde como se aprecia, se nos piden como argumentos de entrada las coordenadas absolutas del punto, un plano de referencia y la profundidad de mismo. Destacar que *hole* trabaja con diámetros y no con radios por lo que si como parámetro tenemos el radio, hay que multiplicarlo por dos para que el programa realice correctamente el agujero.

```
Dim hole1 As Hole
Set hole1 = shapeFactory1.AddNewHoleFromPoint(X, Y, Z,~~reference3, longitud)
hole1.Type = catSimpleHole
hole1.AnchorMode = catExtremPointHoleAnchor
hole1.BottomType = catFlatHoleBottom
hole1.Name = "hueco interno"

Dim limit2 As Limit
Set limit2 = hole1.BottomLimit
limit2.LimitMode = catOffsetLimit
Dim length3 As Length
Set length3 = hole1.Diameter
length3.Value = 2 * radio_int 'Hole trabaja con diámetro no con radio
```

Código 23. *Hole*.

Además, para que la herramienta funcione correctamente, hay que especificar una serie de elementos:

- El ***Type***: indica el tipo de *hole* que se quiere realizar, en este caso un *catSimpleHole.*
- El ***AnchorMode*:** indica donde está anclado el *hole*. En el caso del ejemplo se indica *catExtremPointHoleAnchor*, indicando que se trata de un *hole* que parte de un punto específico en el extremo del sólido.
- El ***BottomType***: se precisa dónde finaliza el agujero que se desea realizar, que en *catFlatHoleBottom* se está especificando que es hasta el fondo del sólido.

Reference [10].

4.1.4. Slot

Otra herramienta para eliminar material es la conocida como *slot*. A diferencia de las dos anteriores, podemos dar como argumento de entrada la forma o *sketch* que queremos desechar y además una guía que recorrerá dicho *sketch*, de modo que, finalmente, se elimina material a lo largo de una trayectoria. Esta herramienta nos servirá para hacer los canales de rodadura de los elementos rodantes.

Los códigos necesarios para programar un *slot* son los siguientes:

```
Dim slot1 As Slot
Set slot1 = shapeFactory1.AddNewSlotFromRef(Nothing, Nothing)

Dim reference24 As Reference
Set reference24 = part1.CreateReferenceFromObject(sketch3)
slot1.SetProfileElement reference24
Dim reference25 As Reference
Set reference25 = part1.CreateReferenceFromObject(sketch4)
slot1.CenterCurveElement = reference25
```

Código 24. *Slot.*

4.1.5. Shaft

La herramienta *shaft* al igual que ocurría con el *pad* o el *pocket* requiere crear previamente un *sketch*, que sirva de base para poder revolucionarlo luego en torno a un eje para conformar el sólido de revolución deseado.

Los dos parámetros en los que se basa la herramienta son:

- **El ángulo de revolución:** es muy importante destacar que se debe programar en grados sexagesimales y no en radianes como ocurría en el círculo.
- **El eje de revolución:** es necesario indicar la dirección alrededor de la cual se genera la pieza.

Las siguientes líneas de código recogen la programación de un *shaft*:

```
Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory
Dim shaft1 As Shaft
Set shaft1 = shapeFactory1.AddNewShaft(sketch1)

' Se define el ángulo de revolución

Dim angle1 As Angle
Set angle1 = shaft1.FirstAngle
angle1.Value = TextBox1' Valor del ángulo introducido en el menú

Dim parameters1 As Parameters
Set parameters1 = part1.Parameters
Dim length1 As Length
Set length1 = parameters1.Item("Part1\PartBody\Shaft.1\ThickThin1")
length1.Value = 0
shaft1.Name = "Revolución"

' Se establece el eje de revolución

Dim reference12 As Reference
Set reference12 = part1.CreateReferenceFromObject(line2D2)
shaft1.RevoluteAxis = reference12
```

Código 25. *Shaft*.

Como se puede observar, es necesario definir un ángulo y establecer que este sea el ángulo de revolución del sólido, así como crear una referencia para definirla como el eje de revolución del *sketch*. *Reference [10]*.

4.1.6. Rib

Otra de las herramientas más potentes y útiles del objeto *shapefactory* es la llamada *Rib*, la cual permite hacer una extrusión de un *sketch* sobre una guía que se le marque. Por tanto habrá que generar dos *sketch* que habrá que aportar a la herramienta como argumentos:

- El perfil: da forma al sólido que queremos generar y podrá ser un *sketch* abierto en el caso en el que el *rib* se realice tangente a una superficie y

queremos un acabado suave sobre la misma.

- Guía: indica el camino que queremos que siga el perfil.

Es importante que a la hora de programar, el perfil y la guía tengan un punto en común, es decir, que estén en contacto en algún punto de su geometría ya que podrían surgir algunos errores o problemas en el sólido extruido en caso de que no compartan algún punto. Las líneas de código para programar un *rib* son muy sencillas y se muestran a continuación:

```
Dim rib1 As Rib
Set rib1 = shapeFactory1.AddNewRibFromRef(refrib1, refrib2)
Part1.update
```

Código 26. *Rib*.

4.1.7. CircularPattern

Esta herramienta es una de las más difíciles de programar debido a la cantidad de argumentos de entrada que pide dicha función.

Dicha herramienta resulta de gran utilidad ya que permite hacer copias de un sólido o de otra herramienta como un *pocket* o un *hole*, en las direcciones radial y circunferencial sin necesidad de crear un origen de coordenadas cilíndrico.

La principal diferencia con respecto a las anteriores herramientas que se han usado para generar sólidos es que no pide un *sketch* como argumento de entrada de la función, sino que se necesita como partida un sólido, el cual se puede realizar con cualquiera de las operaciones descritas.

Se muestra primero el código para programar un *circularPattern*, y a continuación se analizarán todos los argumentos que necesitamos como entrada, así como los diferentes parámetros que se deben definir para el correcto funcionamiento de dicha herramienta.

```
Dim reference8 As Reference
Set reference8 = part1.CreateReferenceFromName("")

Dim reference9 As Reference
Set reference9 = part1.CreateReferenceFromName("")

Dim circPattern1 As CircPattern
Set circPattern1 = shapeFactory1.AddNewCircPattern(Nothing, 1, 2, 20#, 45#, 1, 1,
reference8, reference9, True, 0#, True)
```

```
circPattern1.CircularPatternParameters = catInstancesandAngularSpacing

Dim angularRepartition1 As AngularRepartition
Set angularRepartition1 = circPattern1.AngularRepartition

Dim intParam1 As IntParam
Set intParam1 = angularRepartition1.AngularSpacing
intParam1.Value = 25#

Dim intParam2 As IntParam
Set intParam2 = angularRepartition1.InstancesCount
intParam2.Value = 5#

Dim hybridShapes1 As HybridShapes
Set hybridShapes1 = body1.HybridShapes

Dim hybridShapeLineExplicit1 As HybridShapeLineExplicit
Set hybridShapeLineExplicit1 = hybridShapes1.Item("Z Axis")

Dim reference10 As Reference
Set reference10 = part1.CreateReferenceFromObject(hybridShapeLineExplicit1)

circPattern1.SetRotationAxis reference10

part1.UpdateObject circPattern1
```

Código 27. *CircularPattern.*

De las distintas opciones que permite escoger dicha herramienta, se ha explicado la que se ha usado en la realización del proyecto, la cual crea las copias que se desean basándose en los siguientes parámetros:

- Número de copias que se desean realizar.
- Espacio entre instancias (en grados entre las mismas).

5. CATPRODUCT

Una vez que se han creado las piezas que formarán los rodamientos, hay que ensamblarlas para que en conjunto formen el producto que se desea. Para esta tarea es necesario emplear la sección de CATIA llamada *CatProduct*, y dentro de esta se encuentra un módulo muy potente que nos permitirá unir y ensamblar las piezas de los rodamientos. Este módulo es conocido como *Assembly Design*.

5.1. Assembly Design

Las piezas que se van creando tienen que ser guardadas en una carpeta común para que cuando se vayan a unir, el módulo haga una llamada a las piezas y estas se alojen en el árbol de trabajo.

El módulo *Assembly Design* dispone de una serie de objetos que resultan fundamentales para conseguir el objetivo de este proyecto. La estructura que presenta el *ProductDocument* es la que se muestra en la figura 44.

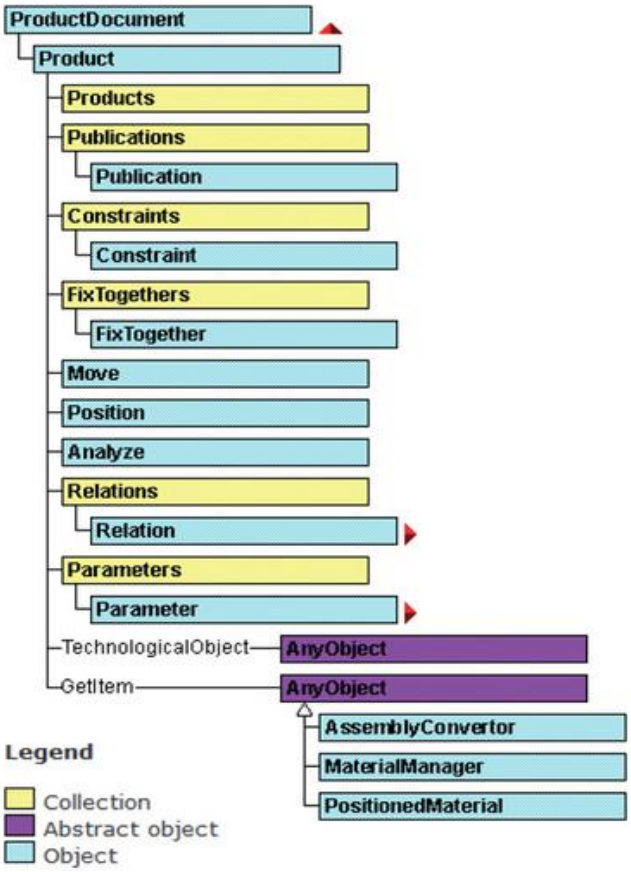

Figura 43. Estructura interna del *ProductDocument*.

Los parámetros, las fórmulas, las restricciones y publicaciones de los *CatProducts* están almacenados en los objetos llamados *Parameters, Relations, Constraints, FixTogethers,* y *Publications*.

Mediante este módulo, se van a imponer las restricciones necesarias entre los distintos *CatParts* creados para que el conjunto sea reconocido como un rodamiento. A continuación se muestran las herramientas usadas en este proyecto. *Referencia [12]*.

Nombre	Icono	Función
AddComponentsFromFiles		Cargar *CatParts,* creados previamente, en el árbol de trabajo del *CatProduct.*
Fix Component		Fijar una pieza en la posición en la que se encuentra.
Offset Constraint		Establecer una distancia fija entre dos puntos, superficies, rectas, etc.
Coincidence Constraint		Establece una relación de coincidencia entre los ejes de circunferencias, cilindros...

Tabla 11. Herramientas del módulo *Assembly Design*.

A continuación se muestran en más detalle las herramientas usadas así como unos códigos de cada restricción que sirvan de ejemplos aclarativos.

5.1.1. AddComponentsFromFiles

Lo primero que hay que hacer es seleccionar y añadir las piezas que se desean al *ProductDocument*. Todas estas piezas han sido guardadas en una carpeta, de la cual conocemos su directorio.

Para añadir dichas piezas al producto, solo habrá que decirle al programa que las busque en el directorio en el que se han guardado. En el código que se muestra, hemos guardado el aro externo de un rodamiento de contacto angular en una carpeta cuyo directorio se ha guardado con el nombre *objPath.*

```
‘Ejemplo: añadir piezas a un product.
Dim arrayOfVariantOfBSTR1(0)
arrayOfVariantOfBSTR1(0) = objPath & "\anillo externo.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR1, "All"
```

Código 28. *AddComponentsFromFiles*.

5.1.2. Fix Component

Es normal que cuando se monta un producto, se tome una pieza como referencia sobre la que se montarán el resto. Esta herramienta es la que permite fijar una pieza para tomarla como base sobre la que se ensambla el producto.

Como ya hemos comentado, esta herramienta añade una restricción a la pieza. Esta *constraint* es la que mencionamos anteriormente en el capítulo 5: *catCstTypeReference*. Para usar dicha herramienta, solo hay que crear una referencia de la pieza y darla como argumento.

```
Dim reference1 As Reference
Set reference1 =
 product1.CreateReferenceFromName("Product1/Part1.1/!Product1/Part1.1/")

Dim constraint1 As Constraint
Set constraint1 = constraints1.AddMonoEltCst(catCstTypeReference, reference1)
```

Código 29. *Fix Component.*

5.1.3. Offset Constraint

Esta es una de las herramientas más usadas de este módulo ya que nos permite establecer y fijar las distancias que se requieran entre dos puntos, rectas, superficies, planos de las piezas, etc.

Para fijar distancias, hay que usar la *constraint*, *catCstTypeDistance*. En este caso se definirán dos referencias, una por cada sólido, y habrá que definir una longitud a la que se dará el valor de la distancia de separación.

Como se puede apreciar en el siguiente código, se está estableciendo una relación entre dos superficies *(RSur:(Face:)* pertenecientes a dos piezas distintas (*Part1.1; Part1.2*).

```
Dim reference1 As Reference
Set reference1 =
product1.CreateReferenceFromName("Product1/Part1.1/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;9)));None:();Cf11:());EdgeFillet.2_ResultOUT;Z0;G3055)")

Dim reference2 As Reference
Set reference2 =
product1.CreateReferenceFromName("Product1/Part1.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;18)));None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055)")
```

```
Dim constraint1 As Constraint
Set constraint1 = constraints1.AddBiEltCst(catCstTypeDistance, reference1, reference2)

Dim length1 As Length
Set length1 = constraint1.Dimension

length1.Value = 0#
```

Código 30. *Offset Constraint.*

5.1.4. Coincidence Constraint

Esta es otra de las herramientas de mayor utilidad. Con ella se impone la coincidencia de los ejes de piezas cilíndricas o esféricas, agujeros, etc.

La restricción que habrá que usar ahora es *catCstTypeOn*. De nuevo creamos las referencias de los sólidos que queremos hacer coincidir y luego las introducimos como argumentos. En el siguiente código, se muestra cómo se usa esta herramienta.

```
Dim reference2 As Reference
Set reference2 =
product1.CreateReferenceFromName("Product1/Part1.2/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;6)));None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055))")

Dim reference3 As Reference
Set reference3 =
product1.CreateReferenceFromName("Product1/Part1.1/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;18)));None:();Cf11:());EdgeFillet.2_ResultOUT;Z0;G3055))")

Dim constraint2 As Constraint
Set constraint2 = constraints1.AddBiEltCst(catCstTypeOn, reference2, reference3)
```

Código 31. *Coincidence Constraint*.

6. Aplicación: entorno de programación.

En esta sección se van a desarrollar los procedimientos que se han llevado a cabo para realizar el diseño de los rodamientos.

En primero lugar se va a crear un formulario, en el cual se podrá elegir entre tres tipos de rodamientos, los cuales serán de bolas, de rodillos o de agujas. Y dentro de estos tres grandes tipos, se podrá escoger el que mejor se ajuste a nuestras exigencias. Según el rodamiento en cuestión, se requerirá luego que se introduzcan los datos de partida necesarios.

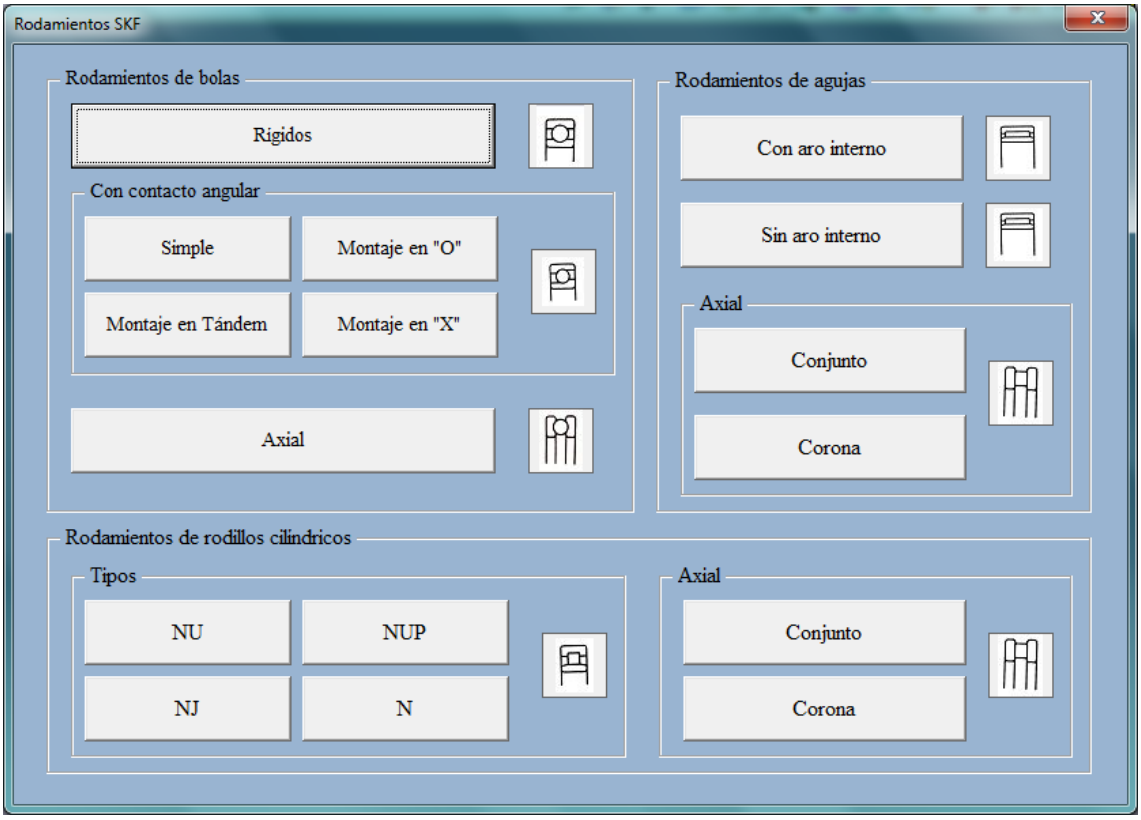

Figura 44. Interfaz del programa.

En el formulario se distinguen tres controles: *Frames* para distinguir los distintos tipos de rodamientos, *CommandButton* para acceder a cada uno de los rodamientos y *Images* para que nos facilite la búsqueda en el catálogo general SKF.

Cada uno de los *CommandButton* de este formulario está programado para que cuando se haga clic sobre él, el programa abra automáticamente el formulario correspondiente con el tipo de rodamiento deseado.

Para ello, se han usado las siguientes líneas de código:

```
Private Sub CommandButton1_Click()
Load NombreUserform
NombreUserform.Show
End Sub
```

Código 32. Llamada a los *Userforms* con un *CommandButton*.

Habrá un *CommandButton* y por lo tanto un formulario asociado a cada tipo de rodamiento que se va a realizar:

- Rodamientos rígidos de bolas.
- Rodamientos de bolas con contacto angular.
- Rodamientos de rodillos cilíndricos tipo N, NJ, NUP y NU.
- Rodamientos de agujas.
- Rodamientos axiales de bolas.
- Rodamientos axiales de rodillos o agujas.

Que hacen un total de dieciséis tipos de rodamientos.

Una vez definida la interfaz de trabajo, se van a explicar los procedimientos que se han llevado a cabo para que dicho programa funcione correctamente.

6.1. Ventana emergente para seleccionar la carpeta de guardado

Como ya se ha mencionado anteriormente, las piezas de cada rodamiento que se van creando, tienen que ir siendo almacenadas en una carpeta para que cuando sean llamadas por el *Product*, se puedan añadir al trabajo. Es obligatorio que todos los *Parts* sean guardados en la misma carpeta ya que de otra forma, el *Product* no funciona correctamente.

Los código que permiten desplegar una ventana emergente son los siguientes:

```
'====================================
'codigo para seleccionar la carpeta donde guardar los archivos
'====================================
Const WINDOW_HANDLE = 0
Const NO_OPTIONS = &H1
Dim objShellApp
Dim objFolder
Dim objFldrItem
Dim objPath
```

```
        Set objShellApp = CreateObject("Shell.Application")
        Set objFolder = objShellApp.BrowseForFolder(WINDOW_HANDLE, strTitle,
NO_OPTIONS)

        Set objFldrItem = objFolder.Self
        objPath = objFldrItem.Path
        BrowseForFolderDialogBox = objPath
        Set objShellApp = Nothing
        Set objFolder = Nothing
        Set objFldrItem = Nothing
```

Código 33. Ventana emergente.

En este código podemos ver que la herramienta *BrowseFolderDialogBox* es la que despliega la ventana y que cuando seleccionamos cualquier carpeta, el directorio de dicha carpeta o *Folder*, se almacena en lo que en el código se llama *objPath*. También tenemos la opción de crear una nueva carpeta en dicha ventana.

Cuando ya se tiene la carpeta donde se irán almacenando las piezas, solo falta ir guardándolas. Para ello se usa la línea de código siguiente.

```
        partDocument1.SaveAs objPath & "\NombreDelPart.CATPart"
```

Código 34. Guardado.

6.2. Rodamientos de bolas

6.2.1. Rodamientos rígidos de bolas

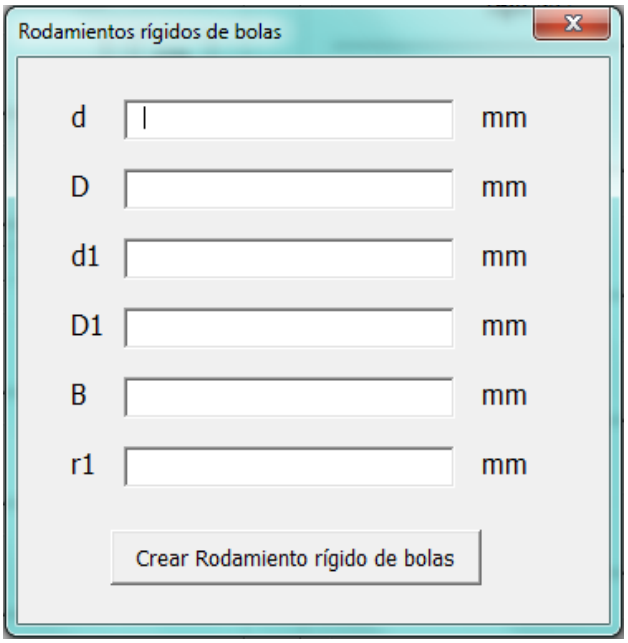

Figura 45. Formulario de los rodamientos rígidos de bolas.

Para desarrollar un rodamiento de bolas se necesitan como datos de partida los parámetros que ya mencionamos en el capítulo 3 pero que aquí recordamos: d, d1, D, D1, B y r. Todas estas medidas se suministrarán en milímetros.

La interfaz asociada al rodamiento rígido de bolas es la que se muestra en la figura 46.

Cada pieza se puede hacer por muchos caminos distintos. En este proyecto, se ha optado por el siguiente:

1. Empezar con un *sketch* de una circunferencia a la que, mediante el objeto *shapefactory*, se le hace un extrusionado con la herramienta *pad*, de un ancho B. El diámetro de la circunferencia dependerá si estamos creando el aro interior o el exterior del rodamiento (d1 ó D, respectivamente).

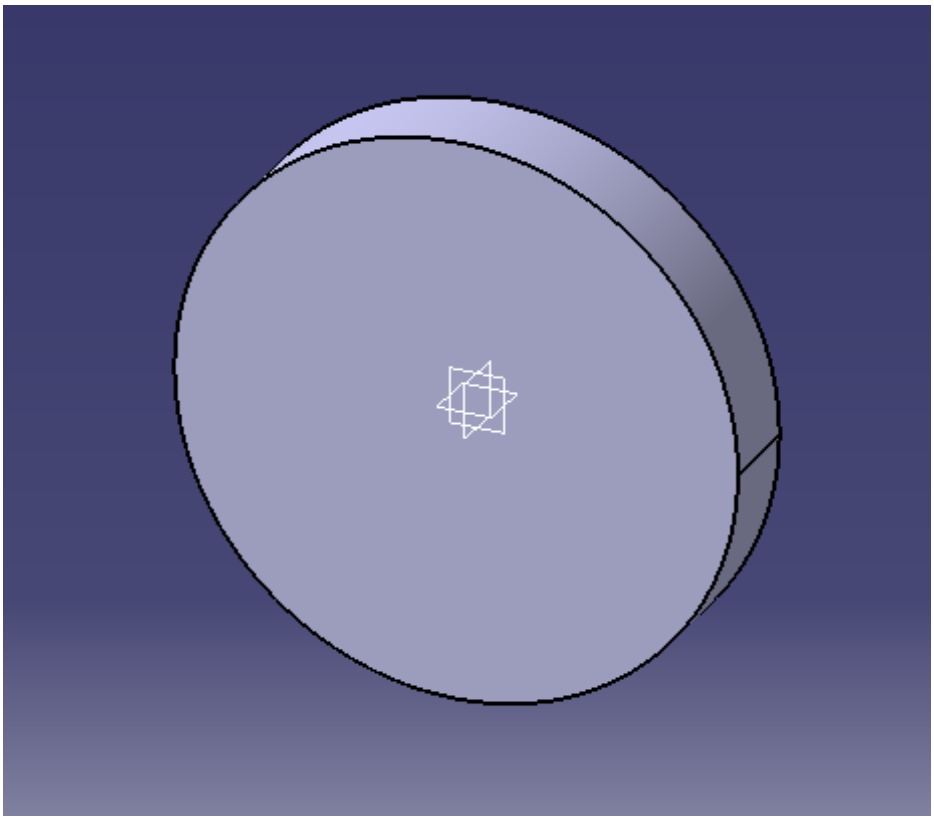

Figura 46. Creación de los aros (1).

2. Sobre una de las dos caras del cilindro resultante, se dibuja otra circunferencia que servirá como perfil del *pocket*. Así obtenemos un anillo o aro. De nuevo, el diámetro de la circunferencia dependerá de si estamos modelando el aro interno o el externo (d ó D1, respectivamente).

3. Para definir la guía por donde rodarán las bolas se ha usado la herramienta *Slot*. El *sketch* que trabaje como perfil para eliminar materia será una circunferencia y el *sketch* que sirva como guía del *slot*, tendrá un diámetro igual a la distancia a la que se situarán las bolas respecto del eje de los aros.

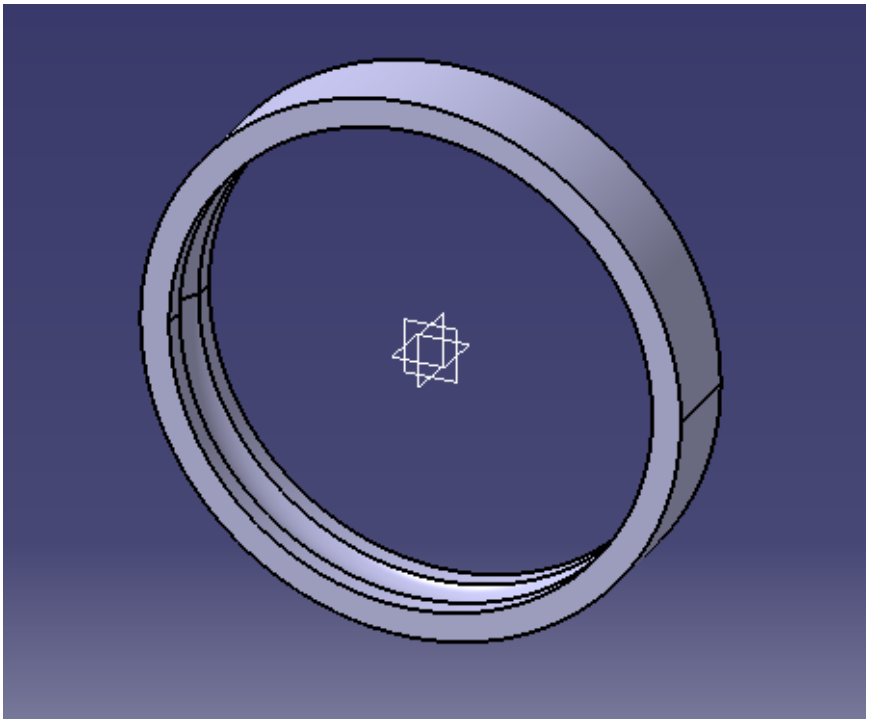

Figura 47. Creación de los aros (2).

4. Para terminar los aros, se harán los chaflanes pertinentes con la herramienta *AddNewSolidEdgeFilletWithConstantRadius*. En el siguiente código se puede observar cómo funciona esta herramienta.

```
        Dim aux As Reference
        Set aux = part1.CreateReferenceFromName("")

        Dim constRadEdgeFillet1 As ConstRadEdgeFillet
        Set constRadEdgeFillet1 =
         shapeFactory1.AddNewSolidEdgeFilletWithConstantRadius(aux,
catTangencyFilletEdgePropagation, R1)

        Dim borde1 As Reference
        Set borde1 = part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(anillo
externo;0:(Brp:(Sketch.1;3)));None:();Cf11:());Face:(Brp:(Pad.1;2);None:();Cf11:());None:(Lim
its1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupp
ort;MFBRepVersion_CXR15)", pad1)

        constRadEdgeFillet1.AddObjectToFillet borde1
```

```
constRadEdgeFillet1.EdgePropagation = catTangencyFilletEdgePropagation
```

Código 35. Chaflán.

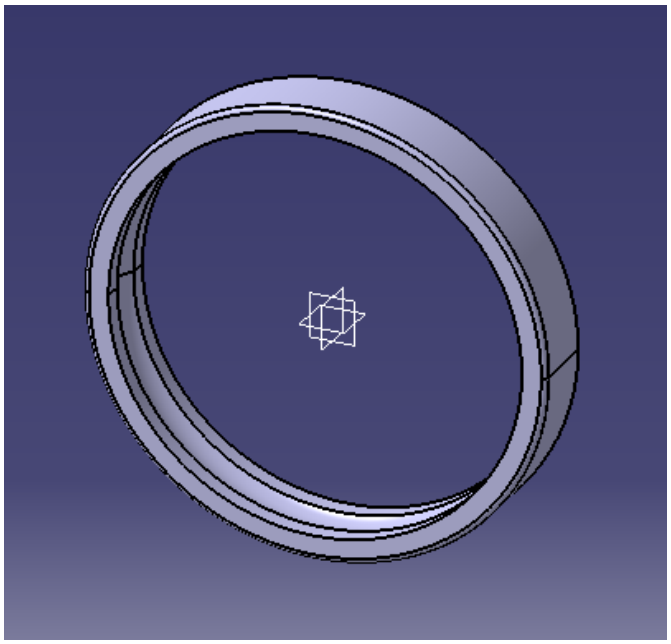

Figura 48. Creación de los aros (3).

5. Guardamos tanto el aro interno como el externo con el código 35.
6. Las bolas se crearán mediante el revolucionado de media circunferencia alrededor del "eje Z". Para esto se usa la herramienta *shaft*, ya explicada en el capítulo 6. Esta bola se creará a una distancia del origen de coordenadas igual a la posición que ocupará en el rodamiento. Esto se hará así porque la bola será sometida a un patrón circular alrededor del eje X para crear el resto de bolas. Si se creara sobre el origen de coordenadas, no podríamos hacer dicho patrón.

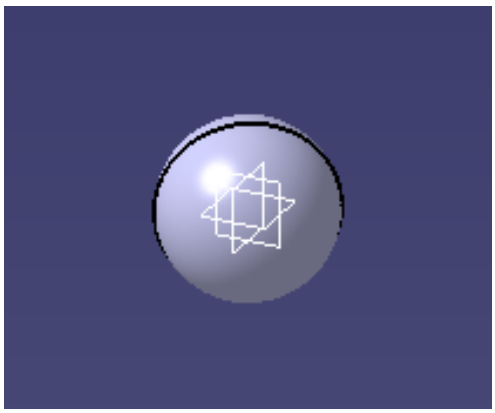

Figura 49. Bola.

7. Como acabamos de indicar, a la bola se le realizará un *CircularPattern* alrededor del eje X obteniendo una corona de bolas.

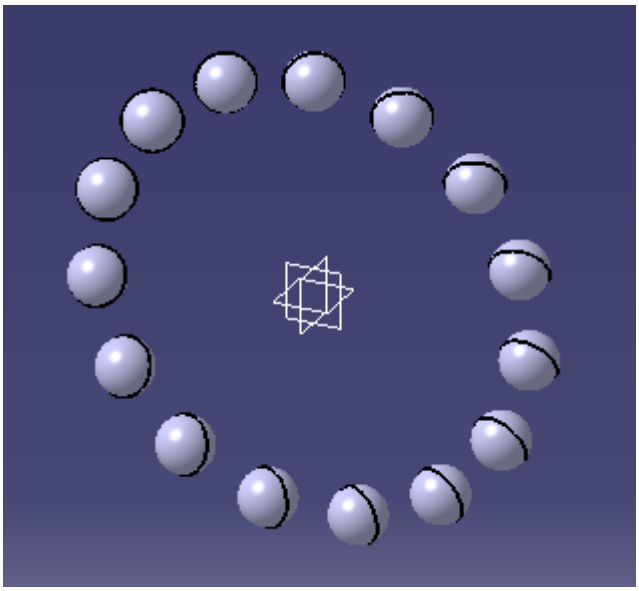

Figura 50. Bolas.

8. Para el separador o jaula, se realizan las mismas operaciones que para los aros pero en vez de realizar un *slot*, se realiza un *pocket* para crear los huecos donde se colocan las bolas y luego se repetirá dicho hueco según un patrón circular. También guardamos la jaula con el código 35.

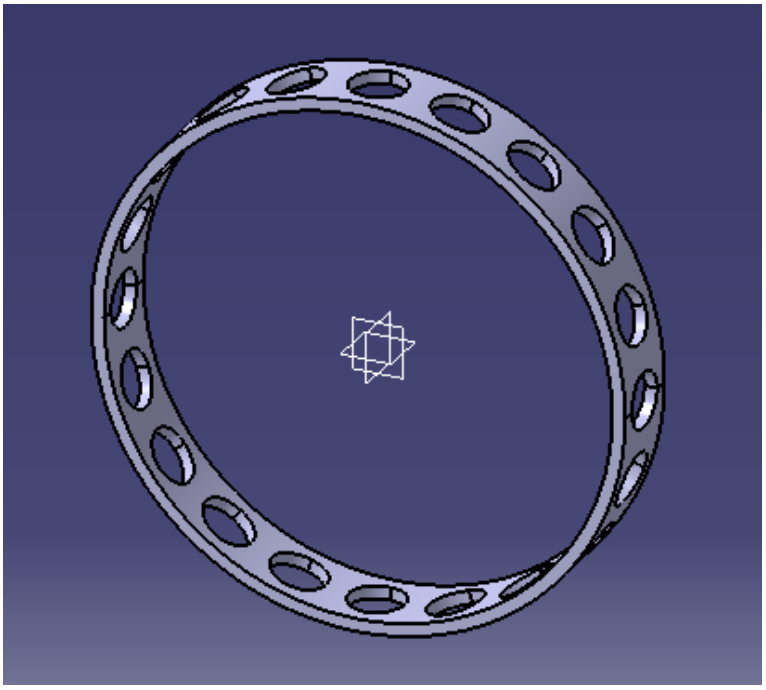

Figura 51. Creación de la jaula.

9. Con todo guardado, se abrirá un *Product*, donde se añadirán todas las piezas. A estas se les pondrán todas las *constrains* explicadas en el capítulo 7. Las restricciones impuestas son:
    - *Fix* del aro externo.
    - Coincidencia del eje del aro externo con el eje del aro interno.
    - Coincidencia del eje del aro externo con el eje de la jaula.
    - Distancia fija igual a cero entre las caras laterales de los aros.
    - Distancia fija igual a cero entre la cara lateral del aro externo y la cara lateral de la jaula.
    - Coincidencia de los planos XY, YZ, y ZX de la jaula y las bolas.

En definitiva, la estructura seguida para la elaboración de los rodamientos ha sido la descrita. Para el caso de los rodamientos de rodillos el slot realizado es con forma de rectángulo y no circular y para los rodamientos axiales, el slot se le realiza sobre una cara lateral. El único rodamiento que se ha creado con un procedimiento distinto ha sido el de bolas con contacto angular.

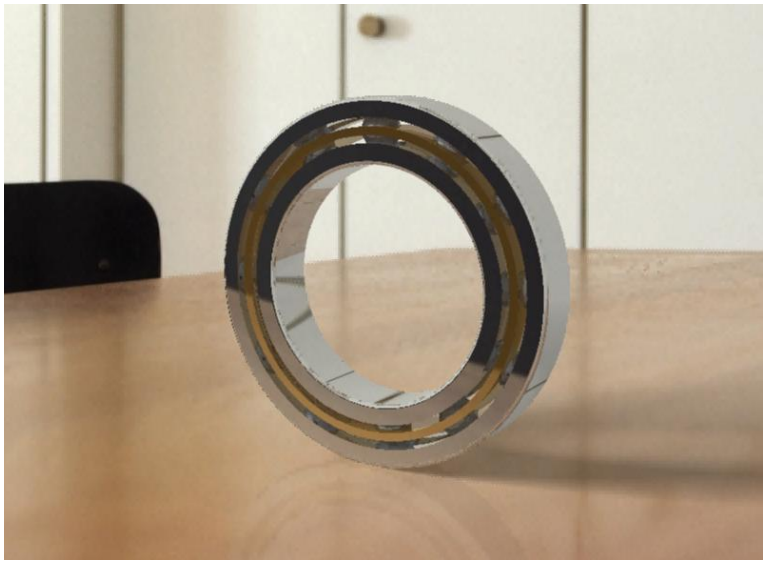

Figura 52. Montaje final del rodamiento de bolas.

6.2.2. Rodamientos de bolas con contacto angular

Los datos que son necesarios son: d, d1, D, D1, B, r, r1 y a. Además ya se resaltó la peculiaridad de un ángulo de contacto fijo de 40° cuando se monta un rodamiento. La interfaz asociada a este rodamiento es la siguiente:

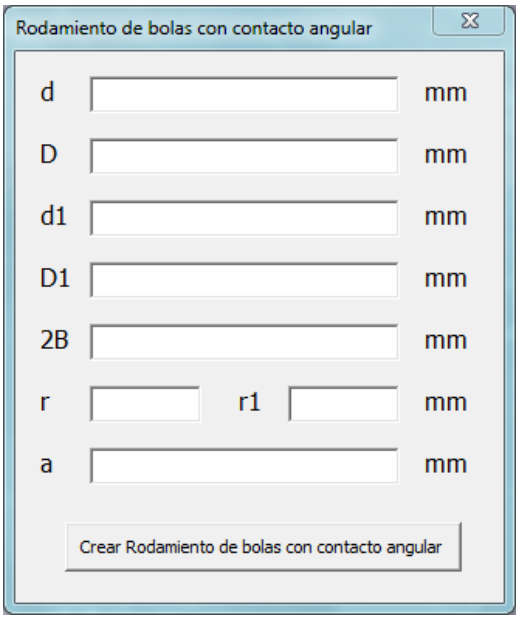

Figura 53. Formulario de los rodamientos de bolas con contacto angular.

Para tener en cuenta dicho ángulo, se ha tenido que hacer un *rib* a un *sketch* para crear los aros. Si se hubiese intentado hacer de la otra forma descrita, no se podría establecer el ángulo de contacto. Dicho dibujo se muestra en la figura 55.

En este *sketch*, las medidas que aparecen, son las obtenidas al ejecutar el programa con unos parámetros determinados.

Además, hay que aportar una guía que recorrerá este dibujo como ya se explicó en el capítulo 6.

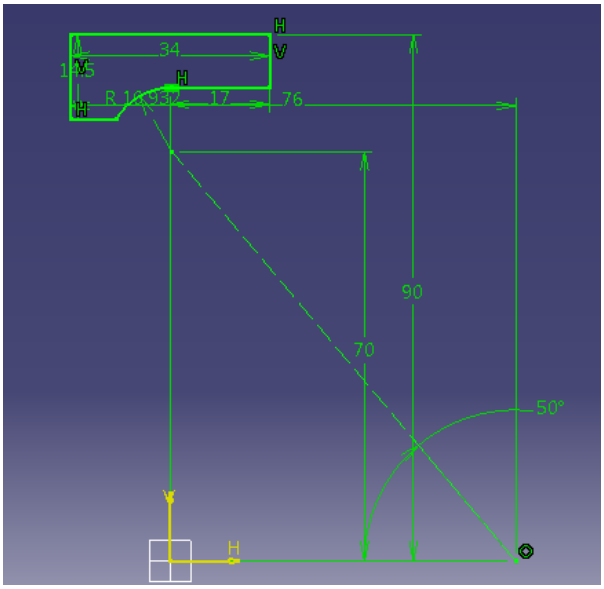

Figura 54. *Sketch* para los rodamientos de bolas con contacto angular.

Las bolas se modelan de la misma forma que en el caso general pero la jaula tiene que ajustarse al espacio entre aros, es decir, tiene forma cónica.

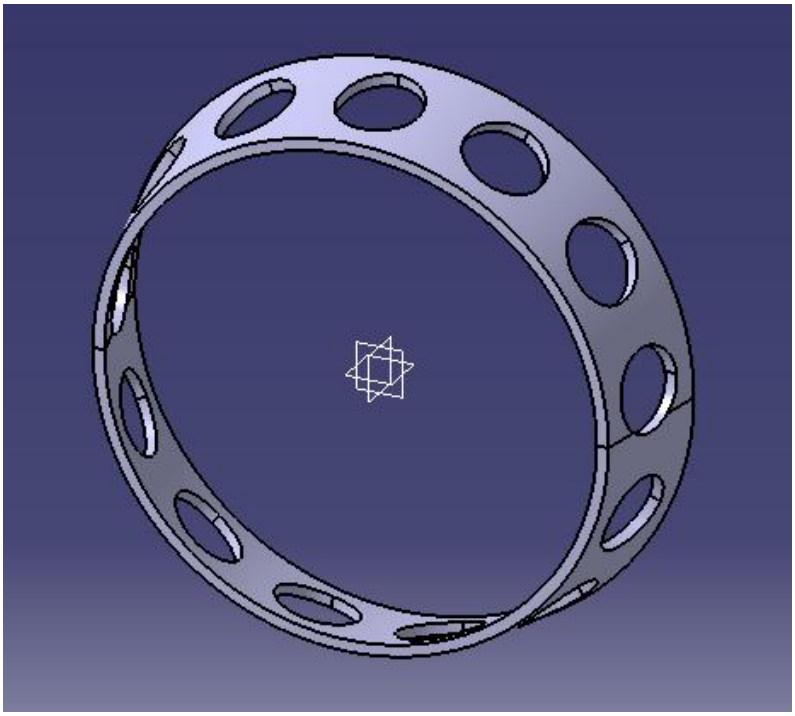

Figura 55. Jaula de los rodamientos de bolas con contacto angular.

En la siguiente figura podemos ver el resultado de montar dos rodamientos con contacto angular en tándem.

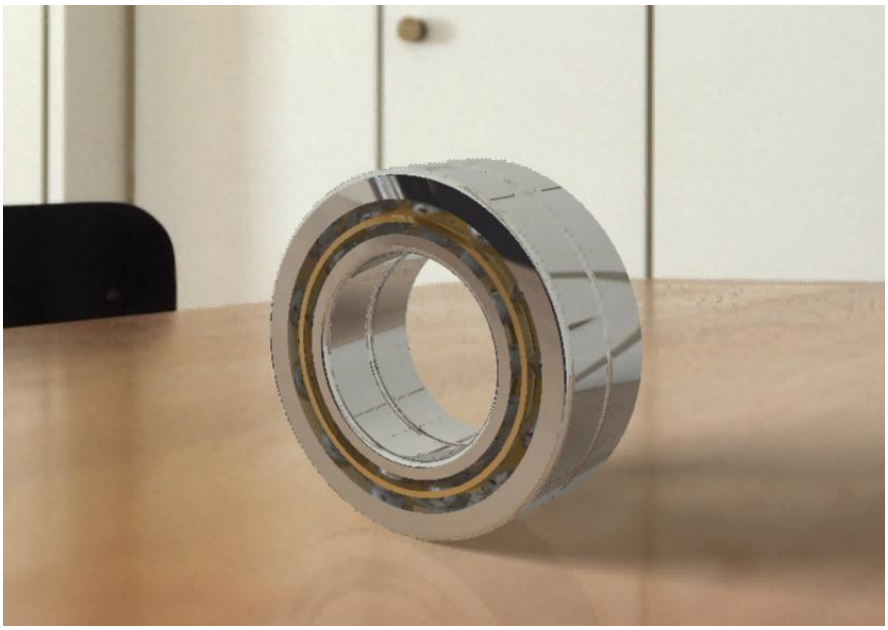

Figura 56. Montaje en tándem de dos rodamientos con contacto angular.

6.2.3. Rodamientos axiales de bolas

Como ya se ha comentado, el modelado de estos rodamientos se hacen por el procedimiento general pero el *slot* se le aplica a la cara lateral tanto del anillo interno como del externo. El formulario pedirá los siguientes datos: d, d1, D, D1, H y r. Este se muestra a continuación:

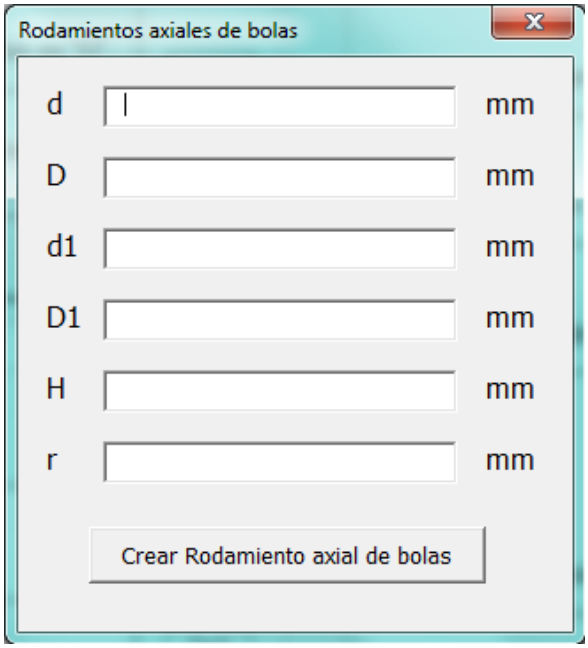

Figura 57. Formulario de los rodamientos axiales de bolas.

El resultado final es como el que se muestra a continuación:

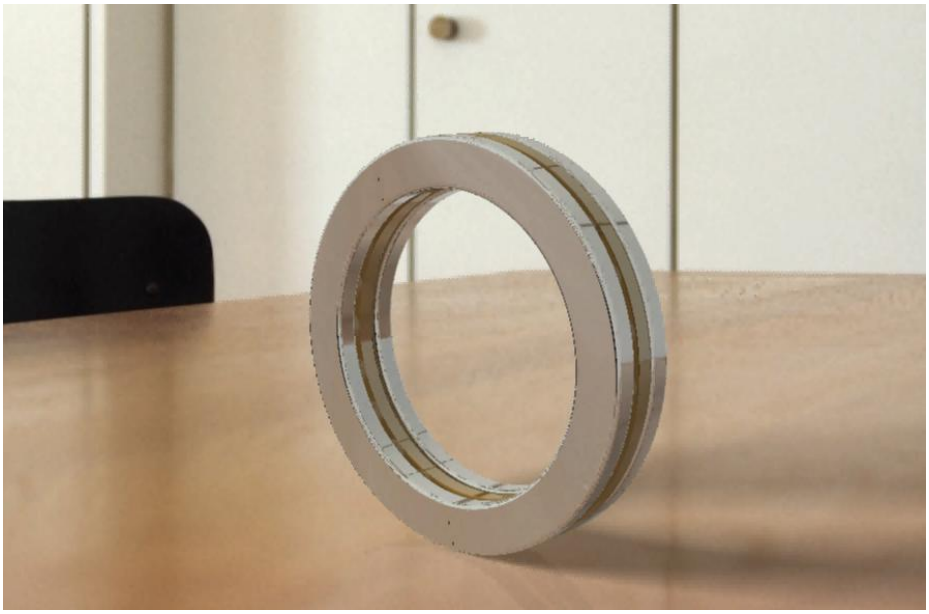

Figura 58. Montaje de un rodamiento axial de bolas.

6.3. Rodamientos de rodillos cilíndricos

6.3.1. Tipos N, NJ, NU y NUP

Cuando se elija el tipo de rodamiento de rodillos cilíndricos deseado, el formulario que aparecerá será el mismo para los cuatro casos.

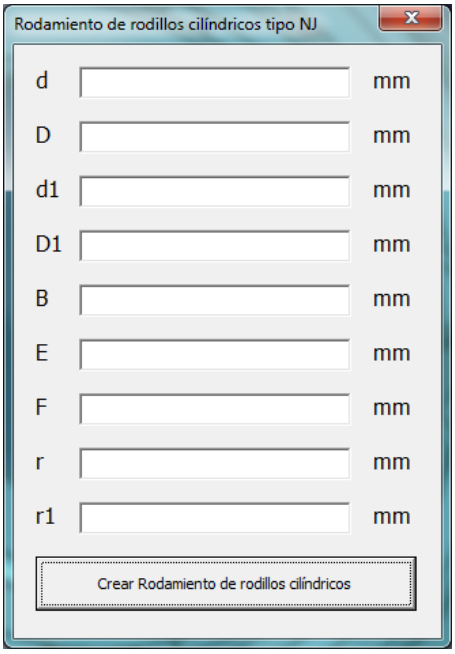

Figura 59. Formulario de los rodamientos de los rodillos cilíndricos.

El procedimiento para realizar las piezas es el mismo que el que se ha descrito detalladamente en el subcapítulo 6.2.1., salvo la evidente forma del *slot* que crea el camino de rodadura de los rodillos. En este caso, el camino de rodadura tiene forma de canal rectangular siendo el contacto entre este y los rodillos, lineal.

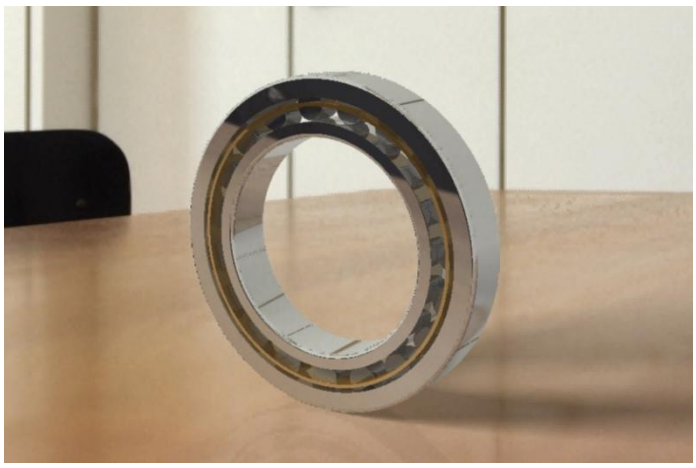

Figura 60. Montaje de un rodamiento de rodillos cilíndricos tipo NU.

Como ya se ha comentado anteriormente, se podrá elegir entre cuatro tipos de rodamientos de rodillos cilíndricos: tipo N, tipo NJ, tipo UN y tipo NUP.

6.3.2. Rodamientos axiales rodillos cilíndricos

Los datos requeridos para crear estos rodamientos son los que aparecen en el formulario mostrado en la siguiente figura.

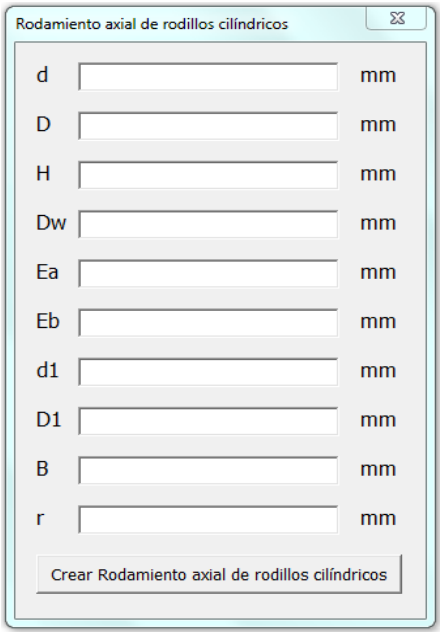

Figura 61. Formulario de los rodamientos axiales y coronas de rodillos cilíndricos.

El procedimiento para crear estos rodamientos es parecido al empleado para crear los rodamientos axiales de bolas salvo que los aros no tienen camino de rodadura por lo que no se les ha hecho un *slot*.

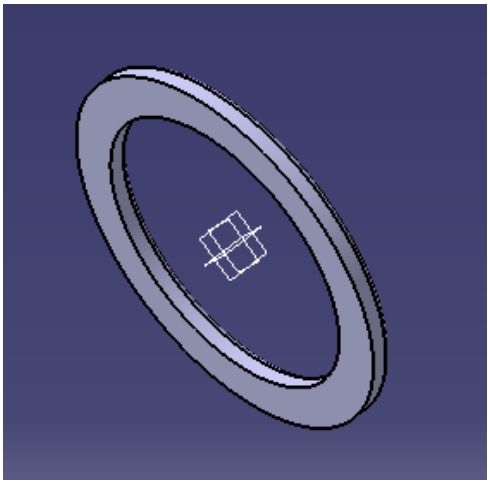

Figura 62. Aro de un rodamiento axial de rodillos.

También es posible solicitar una corona de rodillos cilíndricos. Esta corona, junto con los aros colocados a cada lado, forma el rodamiento completo.

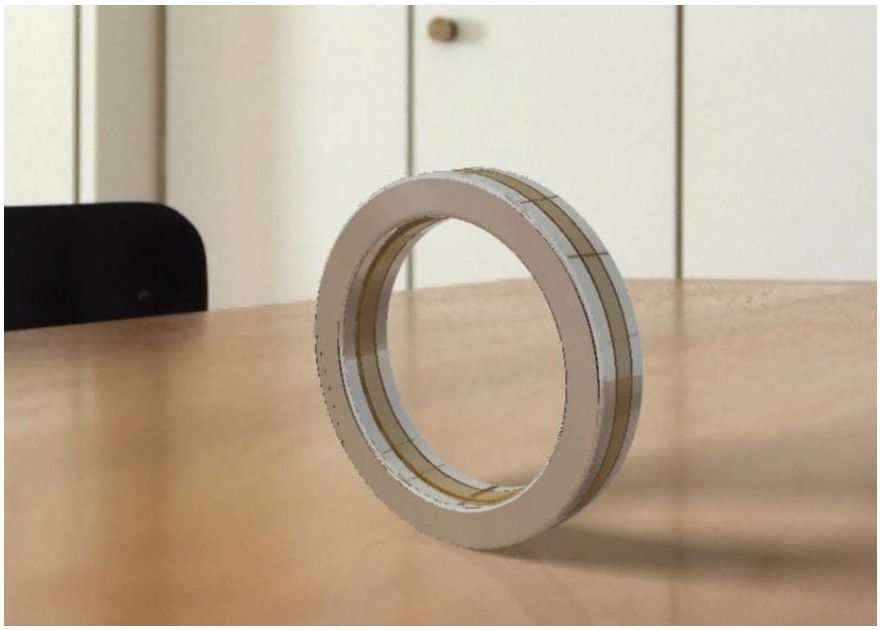

Figura 63. Montaje de un rodamiento axial de rodillos cilíndricos.

6.3.3. Corona de rodillos cilíndricos

Esta corona es la misma que forma el rodamiento axial completo pero es posible utilizarla sin los aros laterales. En este caso, el formulario es de menor tamaño ya que son necesario menos datos para definirlas.

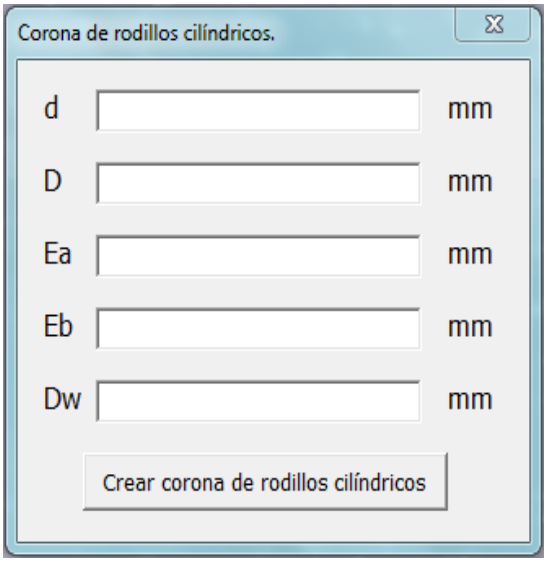

Figura 64. Formulario de las coronas de rodillos.

Estas coronas están formada por la jaula y por todos los rodillos necesarios en los espacios de la jaula. Se puede ver un ejemplo de una corona de rodillos cilíndricos en la figura siguiente.

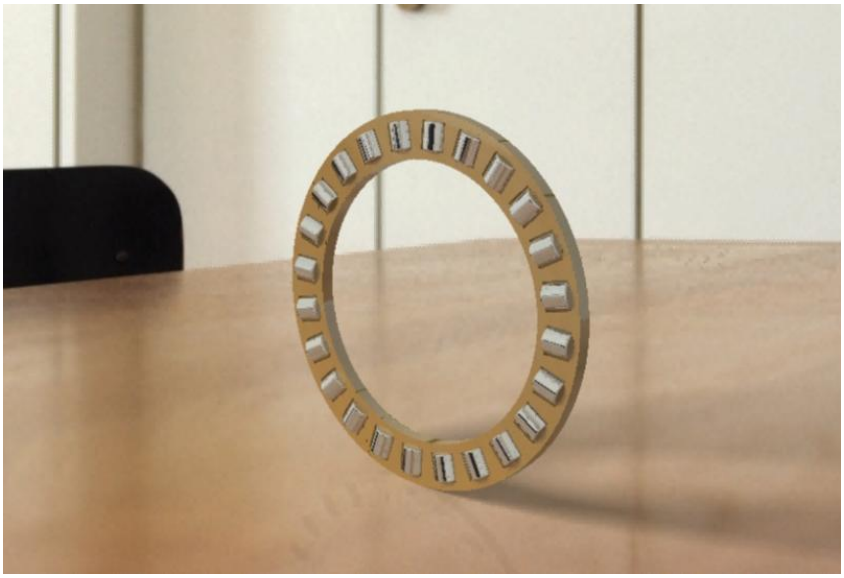

Figura 65. Montaje de una corona de rodillos cilíndricos.

6.4. Rodamientos de agujas

6.4.1. Rodamientos de agujas con aro interno

Este tipo de rodamiento es similar al rodamiento de rodillos cilíndricos tipo N ya que el aro interno no tiene pestañas. Como es obvio, se diferencian en que estos rodamientos usan agujas en vez de rodillos. El formulario que aparecerá cuando se solicita algún rodamiento de este tipo es el siguiente:

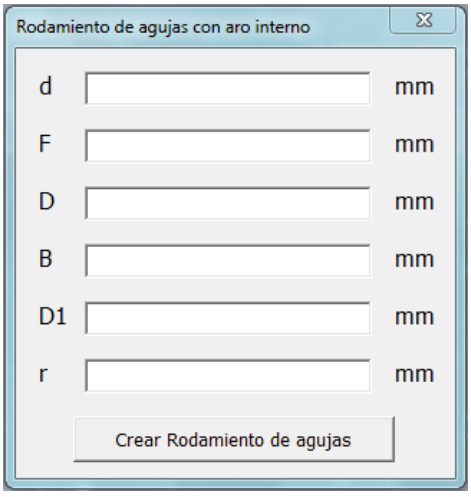

Figura 66. Formulario de los rodamientos de agujas con aro interno.

Para crear estos rodamientos, se sigue el mismo procedimiento descrito anteriormente incluyendo la variación del *slot* para crear los caminos de rodadura ya que ahora tendrán forma rectangular.

Figura 67. Montaje de un rodamiento de agujas con aro interno.

6.4.2. Rodamientos de agujas sin aro interno

Los datos requeridos en el formulario, son menos que en el caso anterior. Este formulario se puede ver en la siguiente imagen:

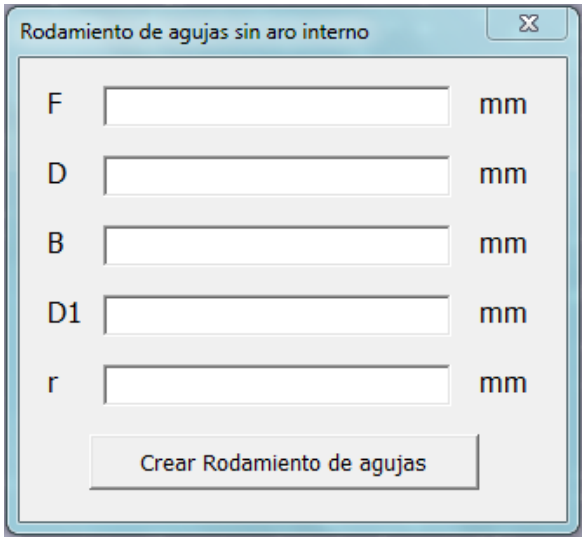

Figura 68. Formulario de los rodamientos de agujas sin aro interno.

Como el mismo nombre indica, la diferencia con los anteriores es la ausencia de aro interno. Estos rodamientos son prácticos para casos en los que el espacio radial donde alojarse sea muy pequeño y por lo tanto no cabría un rodamiento de agujas con aro interno. Un ejemplo de estos rodamientos se muestra a continuación:

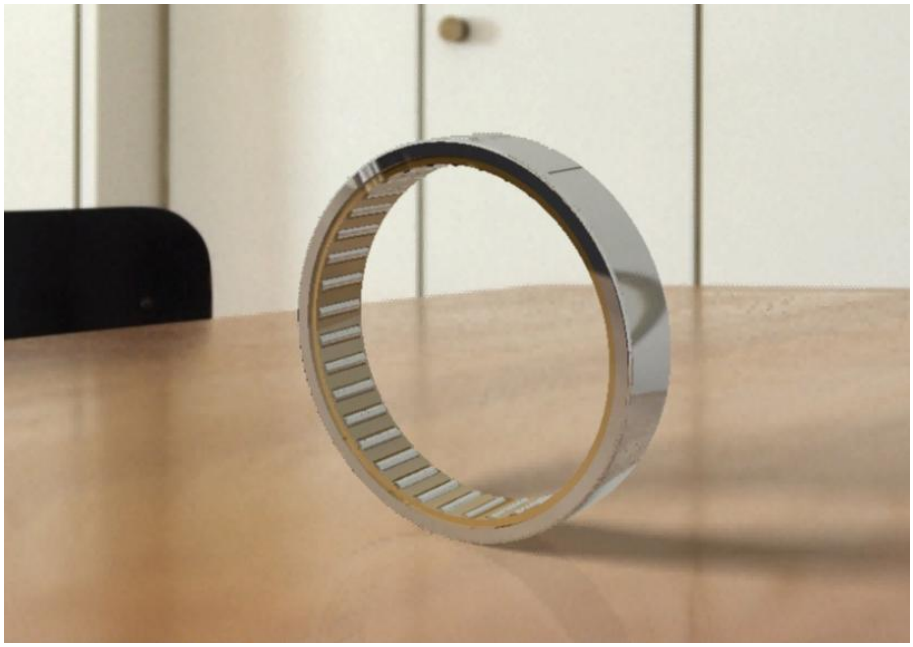

Figura 69. Montaje de un rodamiento de agujas sin aro interno.

6.4.3. Corona de agujas

El formulario de las coronas de agujas presenta el siguiente aspecto:

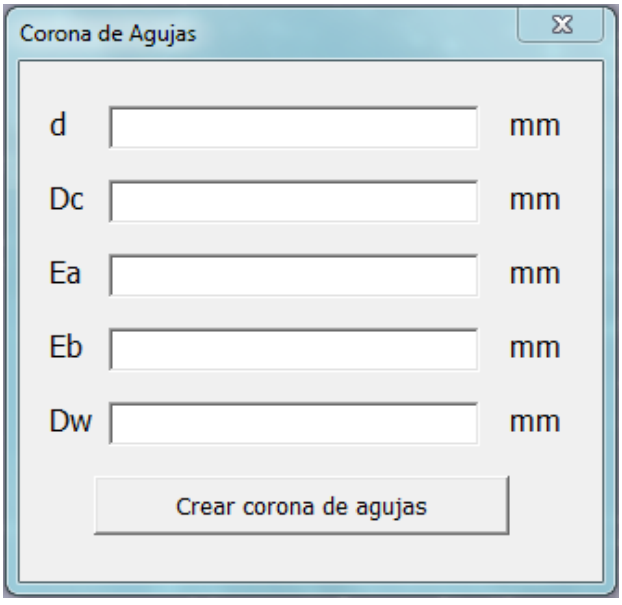

Figura 70. Formulario de las coronas de agujas.

Al igual que para el caso de las coronas de rodillos, las coronas de agujas serán las mimas que formen el rodamiento axial de agujas completo.

Para crearlas se ha seguido el mismo proceso que el codificado para las coronas de rodillos cilíndricos.

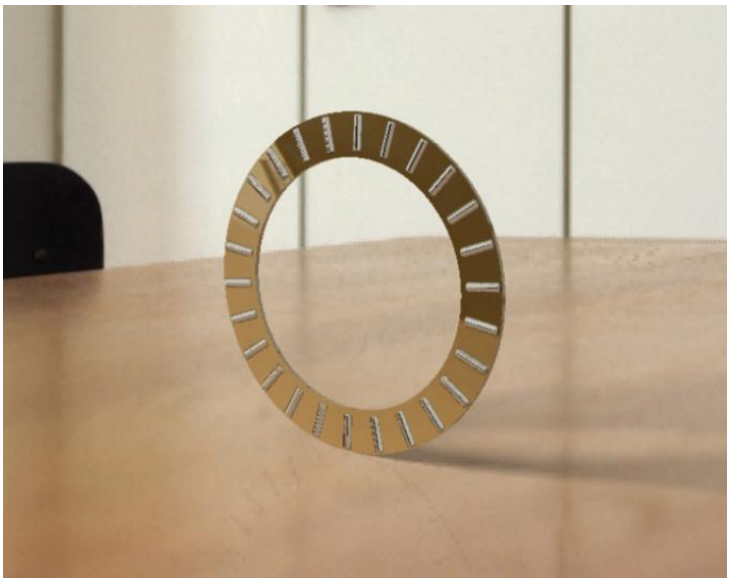

Figura 71. Montaje de las coronas de agujas.

6.4.4. Rodamientos axiales de agujas

Por último, se puede obtener un rodamiento axial de agujas, para los que hay que aportar los datos que solicita su formulario.

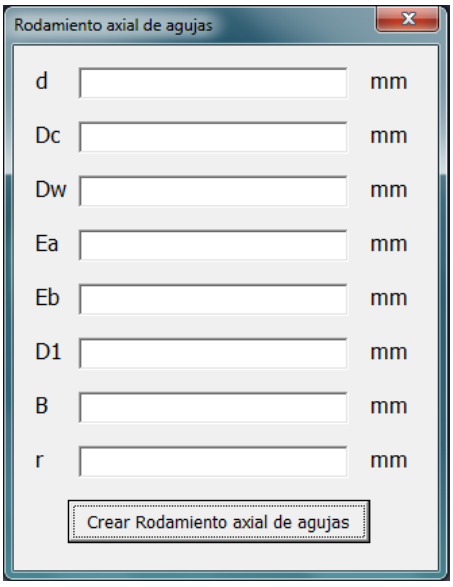

Figura 72. Formulario de los rodamientos axiales de agujas.

Como puede apreciarse, tiene un aspecto muy similar al de los rodamientos axiales de rodillos cilíndricos.

De igual manera, el procedimiento para crearlos es el mismo que el de estos rodamientos, utilizando la corona de agujas del apartado anterior y dos arandelas de soporte y de eje.

Figura 73. Montaje de un rodamiento axial de agujas.

7. Aplicación Rodamiento SKF: manual de usuario

Una vez explicado el software desarrollado para realizar los rodamientos, se va a ilustrar a continuación los pasos que debe seguir el lector para el uso del mismo.

En primer lugar, el usuario debe arrancar CATIA V5. Debe abrir la pestaña *Tools* de la barra de herramientas y acceder a las macros → *Visual Basic Editor*.

Una vez abierto la interfaz de VBA, para arrancar la macro se debe clicar sobre el formulario "SKFrodamientos" y reproducirla dándole al botón *play* de la barra de herramientas. Se abrirá la siguiente ventana:

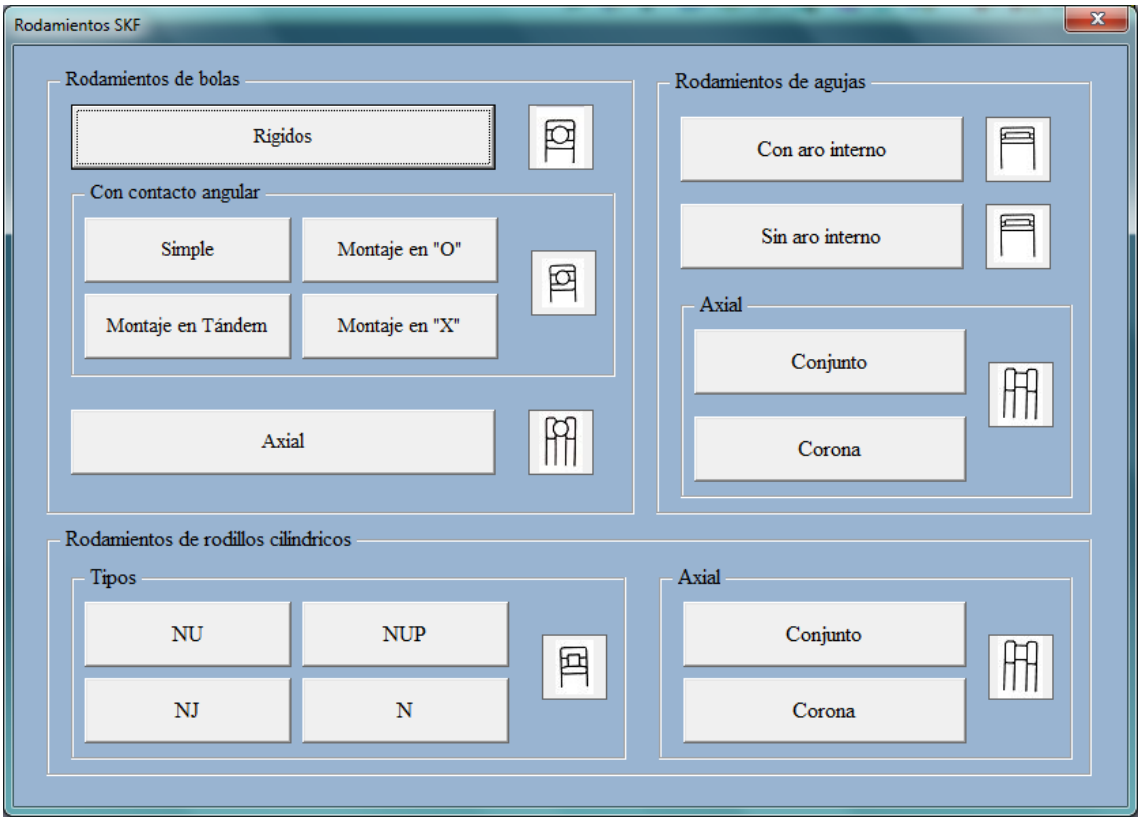

Figura 74. Interfaz de la aplicación Rodamientos SKF.

Sobre esta ventana se seleccionará el tipo de rodamiento que se desea realizar. Para ello se hará clic sobre la pestaña, abriéndose el interfaz correspondiente al rodamiento en cuestión. Se deben rellenar los datos que se requieren en cada formulario y a continuación, hay que clicar sobre el botón Crear.

Lo primero que nos aparecerá, será una ventana para que seleccionemos una carpeta donde se guardarán los aros, los elementos rodantes y la jaula. Esta ventana

también nos da la posibilidad de crear una nueva carpeta. Se deberá pinchar sobre una carpeta y a continuación se clicará sobre aceptar.

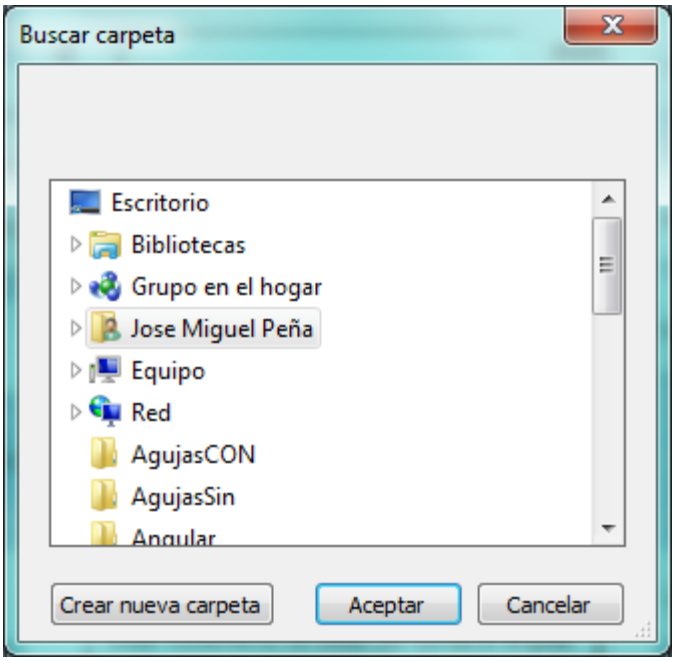

Figura 75. Ventana para escoger la carpeta de guardado.

De este modo nos aparecerán sobre la ventana de trabajo de CATIA, cinco archivos: cuatro CATParts y un CATProduct.

Como ya se ha comentado en el capítulo 1, cada rodamiento nos solicita unos datos distintos. Estos vienen dados en mm en el catálogo general SKF.

8. Conclusiones

CATIA V5 (*Computer Aided Three Dimensional Interactive Aplication*) es un programa informático de alto nivel que viene siendo utilizado en grandes y relevantes áreas de la ingeniería como son la aeronáutica o la automoción, ya que permite realizar tanto el diseño y el análisis de los modelos, como la fabricación final de los mismos.

Sin embargo, a diferencia de otros software CAD/CAM, no posee módulos ni herramientas específicas para el diseño de diferentes piezas y elementos mecánicos. Es por ello que en este proyecto se ha querido realizar una introducción al lenguaje de programación *Visual Basic for Aplication* (VBA), el cual permite crear macros y aplicaciones para CATIA V5.

A la hora de realizar el presente proyecto, se pensó que era buena idea realizar una aplicación para CATIA V5 que permitiera realizar algún tipo de pieza o elemento mecánico perteneciente a una familia de piezas, es decir, aquellas que se definen a partir de la similitud de los atributos de diseño y fabricación de sus miembros. Tras algunas deliberaciones con la tutora de este proyecto, Cristina Torrecillas, se optó por desarrollar una aplicación que permitiera realizar familias de rodamientos del catálogo general ofrecido por la empresa SKF.

En una primera aproximación se decidió generar los rodamientos como una sola pieza, es decir, sobre un solo *CATPart*. Este fue el primer gran error que se cometió ya que a medida que se iban creando más piezas empezaron a aparecer errores en el dimensionado de los rodamientos. Concretamente, cuando se implantaba el patrón circular de la jaula, aparecían agujeros donde no debían estar. A partir de este punto, se decidió comenzar de nuevo, creando cada pieza en un *CATPart* distinto y finalmente montándolas en un *CATProduct*.

Al final del proyecto, durante el montaje de las piezas, apareció otro problema con la codificación de la herramienta que en un primer lugar se pensó utilizar para realizar el patrón circular de las bolas. Este método del módulo *Assembly Design*, se llama *ReusePattern*. Finalmente resultó que para dicha herramienta, no existe un objeto para codificarla por lo que si se hubiese decidido usarla, el usuario tendría que aplicarla manualmente. Por este motivo se decidió que las bolas se crearan en un mismo *CATPart*.

Las aplicaciones desarrolladas se centran en los rodamientos que ofrece la empresa SKF aunque estos, se pueden solicitar en cualquier otra factoría. Ya se comentó que no ha sido posible desarrollar el catálogo completo por la falta de parámetros necesarios para el modelado.

En lo referente a la programación en VBA de la aplicación, se han pasado por varias etapas y han tenido lugar varios impedimentos.

En una primera aproximación, como se ha comentado, se planteó realizar los rodamientos como una única pieza. Una vez descartada la idea inicial, se comenzó a programar de nuevo, encontrándonos con que algunas operaciones usadas comúnmente en CATIA como la realización de patrones circulares o chaflanes eran complejas de programar, por lo que hubo que dedicarle mucho tiempo al entendimiento de las mismas y al análisis de los errores que iban apareciendo.

Finalmente lo que se ha obtenido es una macro para CATIA V5 programada en VBA con la cual a partir de ciertos parámetros iniciales característicos de la geometría de los rodamientos, la aplicación es capaz de hacer una representación en 3D del rodamiento que se desea.

9. Bibliografía

[1]. BSA. Bearing Specialist Association.

[2]. Rodamientos. Un manual para escuelas.

[3]. Johammes Brändlein, Paul Eschmann, Ludwig Hasbargen, Karl Weigand. *Ball and Roller Bearings. Third edition.*

[4]. SKF. Catálogo General

[5]. NSK America. Web site. 19/05/2015

[6]. SKF Web Site. 22/05/2015

[7]. Kaydon bearings infinitive solutions. Web site. 29/05/2015

[8]. Averías de rodamientos y sus causas. SKF Comercial España.

[9]. Dassault Systémes Web Site. 07/06/2015

[10]. Cristina Torrecillas. *Introducción a la programación Visual Basic en CATIA V5 Y V6* (no publicado).

[11]. Emmett Ross. *VB Scripting for CATIA V5*. Segunda edición.

[12]. Dieter Ziethen. *CATIA V5 Macro Programming with Visual Basic Script*

Anexo 1: Códigos de programación

- **Rodamiento Rígido de bolas.**

```
Private Sub CommandButton1_Click()
d_ = TextBox1.Value
D = TextBox2.Value
d_1 = TextBox3.Value
D1 = TextBox4.Value
B = TextBox5.Value
r1 = TextBox6.Value
'-----------------------------------------------------------------
---------------------------------
' Titulo: BrowseForFolderDialogBox
'
' Proposito: Abrir una ventana de dialogo para elegir
la capeta donde guardar las piezas.
' Nos devuelve la ruta de la carpeta.
'-----------------------------------------------------------------
---------------------------------
Const WINDOW_HANDLE = 0
Const NO_OPTIONS = &H1
Dim objShellApp
Dim objFolder
Dim objFldrItem
Dim objPath

Set objShellApp = CreateObject("Shell.Application")
Set objFolder =
objShellApp.BrowseForFolder(WINDOW_HANDL
E, strTitle, NO_OPTIONS)

Set objFldrItem = objFolder.Self
objPath = objFldrItem.Path
BrowseForFolderDialogBox = objPath
Set objShellApp = Nothing
Set objFolder = Nothing
Set objFldrItem = Nothing
'=====================================
'----------------------**Aro EXTERNO**-----------------
'=====================================
'definición y establecimiento del entorno de trabajo
'Arrancamos sin necesidad de tener abierto un part de
trabajo
Dim documents1 As Documents
Dim partDocument1 As PartDocument
Dim part1 As Part
Dim bodies1 As Bodies
Dim body1 As Body
Dim sketches1 As Sketches
Dim originElements1 As OriginElements
Dim reference1 As Reference
Dim sketch1 As Sketch

Set documents1 = CATIA.Documents
Set partDocument1 = documents1.Add("Part")
Set part1 = partDocument1.Part
Set bodies1 = part1.Bodies
Set body1 = bodies1.Item("PartBody")
Set sketches1 = body1.Sketches
Set originElements1 = part1.OriginElements
Set reference1 = originElements1.PlaneYZ
Set sketch1 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0#
arrayOfVariantOfDouble1(1) = 0#
arrayOfVariantOfDouble1(2) = 0#
arrayOfVariantOfDouble1(3) = 0#
arrayOfVariantOfDouble1(4) = 1#
arrayOfVariantOfDouble1(5) = 0#
arrayOfVariantOfDouble1(6) = 0#
arrayOfVariantOfDouble1(7) = 0#
arrayOfVariantOfDouble1(8) = 1#
Set sketch1Variant = sketch1

'establece el sistema de ejes absolutos del sketch en
3D
sketch1Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble1
part1.InWorkObject = sketch1

'Establecimiento del conjunto de herramientas 2D y
asignarlo al sketch de trabajo
Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()

'se establecen los elementos geométricos
Dim geometricElements1 As GeometricElements
Set geometricElements1 =
sketch1.GeometricElements

'se define el sistema de ejes dentro del sketch
Dim axis2D1 As Axis2D
Set axis2D1 =
geometricElements1.Item("AbsoluteAxis")

'Establecimiento de las direcciones horizontal y
vertical
Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2

'hacemos primero un pad de un disco de radio D/2
Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateClosedCircle(0#,
0#, D / 2)
Dim point2D1 As Point2D
Set point2D1 = axis2D1.GetItem("Origin")
circle2D1.CenterPoint = point2D1
circle2D1.ReportName = 3

'hacemos las constraints
Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints

Dim reference2 As Reference
Set reference2 =
part1.CreateReferenceFromObject(circle2D1)
Dim constraint1 As Constraint
Set constraint1 =
constraints1.AddMonoEltCst(catCstTypeRadius,
reference2)
constraint1.Mode = catCstModeDrivingDimension
Dim length1 As Length
Set length1 = constraint1.Dimension
length1.Value = D / 2
```

```
sketch1.CloseEdition
part1.InWorkObject = sketch1
part1.Update

'le hacemos el pad a ambos lados (mirrored extent)

Dim shapeFactory1 As ShapeFactory
Dim pad1 As Pad
Set shapeFactory1 = part1.ShapeFactory
Set pad1 = shapeFactory1.AddNewPad(sketch1, B /
2)
pad1.Name = "anillo externo"
pad1.IsSymmetric = True

Dim limit1 As Limit
Set limit1 = pad1.FirstLimit
Dim length2 As Length
Set length2 = limit1.Dimension
length2.Value = B / 2

part1.Update

'CHAFLAN
Dim aux As Reference
Set aux = part1.CreateReferenceFromName("")

Dim constRadEdgeFillet1 As ConstRadEdgeFillet
Set constRadEdgeFillet1 =
shapeFactory1.AddNewSolidEdgeFilletWithConstant
Radius(aux, catTangencyFilletEdgePropagation, r1)

Dim borde1 As Reference
Set borde1 =
part1.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(anillo
externo;0:(Brp:(Sketch.1;3)));None:();Cf11:());Face:(
Brp:(Pad.1;2);None:();Cf11:());None:(Limits1:();Lim
its2:());Cf11:());WithTemporaryBody;WithoutBuildE
rror;WithSelectingFeatureSupport;MFBRepVersion_
CXR15)", pad1)

constRadEdgeFillet1.AddObjectToFillet borde1

constRadEdgeFillet1.EdgePropagation =
catTangencyFilletEdgePropagation

part1.Update

Dim aux2 As Reference
Set aux2 = part1.CreateReferenceFromName("")

Dim constRadEdgeFillet2 As ConstRadEdgeFillet
Set constRadEdgeFillet2 =
shapeFactory1.AddNewSolidEdgeFilletWithConstant
Radius(aux2, catTangencyFilletEdgePropagation, r1)

Dim borde2 As Reference
Set borde2 =
part1.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(anillo
externo;1);None:();Cf11:());Face:(Brp:(anillo
externo;0:(Brp:(Sketch.1;3)));None:();Cf11:());None:
(Limits1:();Limits2:());Cf11:());WithTemporaryBody
;WithoutBuildError;WithSelectingFeatureSupport;M
FBRepVersion_CXR15)", constRadEdgeFillet1)
constRadEdgeFillet2.AddObjectToFillet borde2
constRadEdgeFillet2.EdgePropagation =
catTangencyFilletEdgePropagation

part1.Update

' Se crea la referencia para el pocket
Dim reference3 As Reference
Set reference3 =
part1.CreateReferenceFromName("Selection_RSur:(
Face:(Brp:(anillo
externo;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3
055)")

Dim sketch2 As Sketch
Set sketch2 = sketches1.Add(reference3)

Dim arrayOfVariantOfDouble2(8)
arrayOfVariantOfDouble2(0) = B / 2
arrayOfVariantOfDouble2(1) = 0#
arrayOfVariantOfDouble2(2) = 0#
arrayOfVariantOfDouble2(3) = 0#
arrayOfVariantOfDouble2(4) = 1#
arrayOfVariantOfDouble2(5) = 0#
arrayOfVariantOfDouble2(6) = 0#
arrayOfVariantOfDouble2(7) = 0#
arrayOfVariantOfDouble2(8) = 1#
Set sketch2Variant = sketch2
sketch2Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble2

part1.InWorkObject = sketch2

Dim factory2D2 As Factory2D
Set factory2D2 = sketch2.OpenEdition()
Dim geometricElements2 As GeometricElements
Set geometricElements2 =
sketch2.GeometricElements

Dim axis2D2 As Axis2D
Set axis2D2 =
geometricElements2.Item("AbsoluteAxis")
Dim line2D3 As Line2D
Set line2D3 = axis2D2.GetItem("HDirection")
line2D3.ReportName = 1
Dim line2D4 As Line2D
Set line2D4 = axis2D2.GetItem("VDirection")
line2D4.ReportName = 2

Dim circle2D2 As Circle2D
Set circle2D2 = factory2D2.CreateClosedCircle(0#,
0#, D1 / 2)

Dim point2D2 As Point2D
Set point2D2 = axis2D2.GetItem("Origin")
circle2D2.CenterPoint = point2D2
circle2D2.ReportName = 3

Dim constraints2 As Constraints
Set constraints2 = sketch2.Constraints
Dim reference4 As Reference
Set reference4 =
part1.CreateReferenceFromObject(circle2D2)
Dim constraint2 As Constraint
Set constraint2 =
constraints2.AddMonoEltCst(catCstTypeRadius,
reference4)
constraint2.Mode = catCstModeDrivingDimension
```

```
Dim length3 As Length
Set length3 = constraint2.Dimension
length3.Value = D1 / 2

sketch2.CloseEdition
part1.InWorkObject = sketch2
part1.Update

Dim pocket1 As Pocket
Set pocket1 = shapeFactory1.AddNewPocket(sketch2, B / 2)
pocket1.Name = "agujero interno del anillo exterior"
Dim limit2 As Limit
Set limit2 = pocket1.FirstLimit
limit2.LimitMode = catUpToLastLimit

part1.UpdateObject pocket1
part1.Update

'ahora nos cambiamos de plano para hacer el circulo que creará el carril de las bolas
Dim reference5 As Reference
Set reference5 = originElements1.PlaneZX

Dim sketch3 As Sketch
Set sketch3 = sketches1.Add(reference5)

Dim arrayOfVariantOfDouble3(8)
arrayOfVariantOfDouble3(0) = 0#
arrayOfVariantOfDouble3(1) = 0#
arrayOfVariantOfDouble3(2) = 0#
arrayOfVariantOfDouble3(3) = -1#
arrayOfVariantOfDouble3(4) = 0#
arrayOfVariantOfDouble3(5) = 0#
arrayOfVariantOfDouble3(6) = 0#
arrayOfVariantOfDouble3(7) = 0#
arrayOfVariantOfDouble3(8) = 1#
Set sketch3Variant = sketch3
sketch3Variant.SetAbsoluteAxisData arrayOfVariantOfDouble3

part1.InWorkObject = sketch3

Dim factory2D3 As Factory2D
Set factory2D3 = sketch3.OpenEdition()
Dim geometricElements3 As GeometricElements
Set geometricElements3 = sketch3.GeometricElements

Dim axis2D3 As Axis2D
Set axis2D3 = geometricElements3.Item("AbsoluteAxis")
Dim line2D5 As Line2D
Set line2D5 = axis2D3.GetItem("HDirection")
line2D5.ReportName = 1
Dim line2D6 As Line2D
Set line2D6 = axis2D3.GetItem("VDirection")
line2D6.ReportName = 2

Dim point2D3 As Point2D
Set point2D3 = factory2D3.CreatePoint(0#, (D / 2 - d_ / 2) / 2 + d_ / 2)
point2D3.ReportName = 3
Dim circle2D3 As Circle2D
Set circle2D3 = factory2D3.CreateClosedCircle(0#, (D / 2 - d_ / 2) / 2 + d_ / 2, B / 1.555 / 2)
circle2D3.CenterPoint = point2D3
circle2D3.ReportName = 4

Dim constraints3 As Constraints
Set constraints3 = sketch3.Constraints
Dim reference6 As Reference
Set reference6 = part1.CreateReferenceFromObject(point2D3)
Dim reference7 As Reference
Set reference7 = part1.CreateReferenceFromObject(line2D6)
Dim constraint3 As Constraint
Set constraint3 = constraints3.AddBiEltCst(catCstTypeOn, reference6, reference7)
constraint3.Mode = catCstModeDrivingDimension

Dim reference8 As Reference
Set reference8 = part1.CreateReferenceFromObject(circle2D3)
Dim constraint4 As Constraint
Set constraint4 = constraints3.AddMonoEltCst(catCstTypeRadius, reference8)
constraint4.Mode = catCstModeDrivingDimension
Dim length4 As Length
Set length4 = constraint4.Dimension
length4.Value = B / 1.555 / 2

Dim reference9 As Reference
Set reference9 = part1.CreateReferenceFromObject(point2D3)
Dim reference10 As Reference
Set reference10 = part1.CreateReferenceFromObject(line2D5)
Dim constraint5 As Constraint
Set constraint5 = constraints3.AddBiEltCst(catCstTypeDistance, reference9, reference10)
constraint5.Mode = catCstModeDrivingDimension
Dim length5 As Length
Set length5 = constraint5.Dimension
length5.Value = (D / 2 - d_ / 2) / 2 + d_ / 2

sketch3.CloseEdition
part1.InWorkObject = sketch3
part1.Update

Dim sketch4 As Sketch
Set sketch4 = sketches1.Add(reference1)

Set sketch4Variant = sketch4
sketch4Variant.SetAbsoluteAxisData arrayOfVariantOfDouble1
part1.InWorkObject = sketch4

Dim factory2D4 As Factory2D
Set factory2D4 = sketch4.OpenEdition()
Dim geometricElements4 As GeometricElements
Set geometricElements4 = sketch4.GeometricElements

Dim axis2D4 As Axis2D
Set axis2D4 = geometricElements4.Item("AbsoluteAxis")
Dim line2D7 As Line2D
Set line2D7 = axis2D4.GetItem("HDirection")
line2D7.ReportName = 1
```

```
Dim line2D8 As Line2D
Set line2D8 = axis2D4.GetItem("VDirection")
line2D8.ReportName = 2

Dim circle2D4 As Circle2D
Set circle2D4 = factory2D4.CreateClosedCircle(0#,
0#, (D / 2 - d_ / 2) / 2 + d_ / 2)
Dim point2D4 As Point2D
Set point2D4 = axis2D4.GetItem("Origin")
circle2D4.CenterPoint = point2D4
circle2D4.ReportName = 3

Dim constraints4 As Constraints
Set constraints4 = sketch4.Constraints
Dim reference11 As Reference
Set reference11 =
part1.CreateReferenceFromObject(circle2D4)
Dim constraint6 As Constraint
Set constraint6 =
constraints4.AddMonoEltCst(catCstTypeRadius,
reference11)
constraint6.Mode = catCstModeDrivingDimension
Dim length6 As Length
Set length6 = constraint6.Dimension
length6.Value = (D / 2 - d_ / 2) / 2 + d_ / 2

sketch4.CloseEdition
part1.InWorkObject = sketch4
part1.Update

'hacemos un Slot del sketch3 (circulo que crea el
carril) por el
'sketch 4 (circunferencia de radio determinado)
Dim slot1 As Slot
Set slot1 =
shapeFactory1.AddNewSlotFromRef(Nothing,
Nothing)
Dim reference12 As Reference
Set reference12 =
part1.CreateReferenceFromObject(sketch3)
slot1.SetProfileElement reference12
Dim reference13 As Reference
Set reference13 =
part1.CreateReferenceFromObject(sketch4)
slot1.CenterCurveElement = reference13

part1.UpdateObject slot1
part1.Update

Dim product1 As Product
Set product1 = partDocument1.GetItem("Part1")
product1.PartNumber = "aroexterno"
partDocument1.SaveAs objPath & "\aroext.catpart"
partDocument1.Close

'====================================
'-------------------------**Aro INTERIOR**-----
'====================================
'pad del anillo interno
Dim documents2 As Documents
Dim partDocument2 As PartDocument
Dim part2 As Part
Dim bodies2 As Bodies
Dim body2 As Body
Dim sketches2 As Sketches
Dim originElements2 As OriginElements
Dim sketch5 As Sketch
Dim reference2a As Reference

Set documents2 = CATIA.Documents
Set partDocument2 = documents2.Add("Part")
Set part2 = partDocument2.Part
Set bodies2 = part2.Bodies
Set body2 = bodies2.Item("PartBody")
Set sketches2 = body2.Sketches
Set originElements2 = part2.OriginElements
Set reference2a = originElements2.PlaneYZ
Set sketch5 = sketches2.Add(reference2a)

Dim arrayOfVariantOfDouble4(8)
arrayOfVariantOfDouble4(0) = 0#
arrayOfVariantOfDouble4(1) = 0#
arrayOfVariantOfDouble4(2) = 0#
arrayOfVariantOfDouble4(3) = 0#
arrayOfVariantOfDouble4(4) = 1#
arrayOfVariantOfDouble4(5) = 0#
arrayOfVariantOfDouble4(6) = 0#
arrayOfVariantOfDouble4(7) = 0#
arrayOfVariantOfDouble4(8) = 1#
Set sketch2Variant = sketch5

sketch2Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble4
part2.InWorkObject = sketch5

Dim factory2D5 As Factory2D
Set factory2D5 = sketch5.OpenEdition()
Dim geometricElements5 As GeometricElements
Set geometricElements5 =
sketch5.GeometricElements

Dim axis2D5 As Axis2D
Set axis2D5 =
geometricElements5.Item("AbsoluteAxis")
Dim line2D9 As Line2D
Set line2D9 = axis2D5.GetItem("HDirection")
line2D9.ReportName = 1
Dim line2D10 As Line2D
Set line2D10 = axis2D5.GetItem("VDirection")
line2D10.ReportName = 2

Dim circle2D5 As Circle2D
Set circle2D5 = factory2D5.CreateClosedCircle(0#,
0#, d_1 / 2)
Dim point2D5 As Point2D
Set point2D5 = axis2D5.GetItem("Origin")
circle2D5.CenterPoint = point2D5
circle2D5.ReportName = 3

Dim constraints5 As Constraints
Set constraints5 = sketch5.Constraints

Dim reference14 As Reference
Set reference14 =
part2.CreateReferenceFromObject(circle2D5)
Dim constraint7 As Constraint
Set constraint7 =
constraints5.AddMonoEltCst(catCstTypeRadius,
reference14)
constraint1.Mode = catCstModeDrivingDimension
Dim length7 As Length
Set length7 = constraint7.Dimension
length7.Value = d_1 / 2
```

```
sketch5.CloseEdition
part2.InWorkObject = sketch5
part2.Update

Dim shapeFactory2 As ShapeFactory
Set shapeFactory2 = part2.ShapeFactory
Dim pad2 As Pad
Set pad2 = shapeFactory2.AddNewPad(sketch5, B / 2)
pad2.Name = "anillo interno"
pad2.IsSymmetric = True
Dim limit3 As Limit
Set limit3 = pad2.FirstLimit
Dim length8 As Length
Set length8 = limit3.Dimension
length8.Value = B / 2
part2.Update

'hacemos una referencia de la cara donde vamos a hacer el pocket
'--POCKET--
Dim reference15 As Reference
Set reference15 = part2.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(anillo interno;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")

Dim sketch6 As Sketch
Set sketch6 = sketches2.Add(reference15)

Dim arrayOfVariantOfDouble5(8)
arrayOfVariantOfDouble5(0) = B / 2
arrayOfVariantOfDouble5(1) = 0#
arrayOfVariantOfDouble5(2) = 0#
arrayOfVariantOfDouble5(3) = 0#
arrayOfVariantOfDouble5(4) = 1#
arrayOfVariantOfDouble5(5) = 0#
arrayOfVariantOfDouble5(6) = 0#
arrayOfVariantOfDouble5(7) = 0#
arrayOfVariantOfDouble5(8) = 1#
Set sketch6Variant = sketch6
sketch6Variant.SetAbsoluteAxisData arrayOfVariantOfDouble5
part2.InWorkObject = sketch6

Dim factory2D6 As Factory2D
Set factory2D6 = sketch6.OpenEdition()
Dim geometricElements6 As GeometricElements
Set geometricElements6 = sketch6.GeometricElements

Dim axis2D6 As Axis2D
Set axis2D6 = geometricElements6.Item("AbsoluteAxis")
Dim line2D11 As Line2D
Set line2D11 = axis2D6.GetItem("HDirection")
line2D11.ReportName = 1
Dim line2D12 As Line2D
Set line2D12 = axis2D6.GetItem("VDirection")
line2D12.ReportName = 2

Dim circle2D6 As Circle2D
Set circle2D6 = factory2D6.CreateClosedCircle(0#, 0#, d_ / 2)
Dim point2D6 As Point2D
Set point2D6 = axis2D6.GetItem("Origin")
circle2D6.CenterPoint = point2D6
circle2D6.ReportName = 3
'constraints del círculo
Dim constraints6 As Constraints
Set constraints6 = sketch6.Constraints
Dim reference16 As Reference
Set reference16 = part2.CreateReferenceFromObject(circle2D6)
Dim constraint8 As Constraint
Set constraint8 = constraints6.AddMonoEltCst(catCstTypeRadius, reference16)
constraint2.Mode = catCstModeDrivingDimension

Dim length9 As Length
Set length9 = constraint8.Dimension
length9.Value = d_ / 2

sketch6.CloseEdition
part2.InWorkObject = sketch6
part2.Update

'con el circulo creado hacemos el pocket
Dim pocket2 As Pocket
Set pocket2 = shapeFactory2.AddNewPocket(sketch6, B / 2)
Dim limit4 As Limit
Set limit4 = pocket2.FirstLimit
limit4.LimitMode = catUpThruNextLimit

part2.UpdateObject pocket2
part2.Update

'CHAFLAN
Dim aux3 As Reference
Set aux3 = part2.CreateReferenceFromName("")

Dim constRadEdgeFillet3 As ConstRadEdgeFillet
Set constRadEdgeFillet3 = shapeFactory2.AddNewSolidEdgeFilletWithConstantRadius(aux3, catTangencyFilletEdgePropagation, r1)

Dim borde3 As Reference
Set borde3 = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(anillo interno;2);None:();Cf11:());Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket2)

constRadEdgeFillet3.AddObjectToFillet borde3
constRadEdgeFillet3.EdgePropagation = catTangencyFilletEdgePropagation
part2.UpdateObject constRadEdgeFillet3

Dim aux4 As Reference
Set aux4 = part2.CreateReferenceFromName("")

Dim constRadEdgeFillet4 As ConstRadEdgeFillet
Set constRadEdgeFillet4 = shapeFactory2.AddNewSolidEdgeFilletWithConstantRadius(aux4, catTangencyFilletEdgePropagation, r1)

Dim borde4 As Reference
```

```
Set borde4 = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());Face:(Brp:(anillo interno;1);None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", constRadEdgeFillet3)
constRadEdgeFillet4.AddObjectToFillet borde4
constRadEdgeFillet4.EdgePropagation = catTangencyFilletEdgePropagation

part2.UpdateObject constRadEdgeFillet4
part2.Update


'ahora nos cambiamos de plano para hacer el circulo que creará el carril de las bolas
Dim reference5b As Reference
Set reference5b = originElements2.PlaneZX

Dim sketch3b As Sketch
Set sketch3b = sketches2.Add(reference5b)

Dim arrayOfVariantOfDouble6(8)
arrayOfVariantOfDouble6(0) = 0#
arrayOfVariantOfDouble6(1) = 0#
arrayOfVariantOfDouble6(2) = 0#
arrayOfVariantOfDouble6(3) = -1#
arrayOfVariantOfDouble6(4) = 0#
arrayOfVariantOfDouble6(5) = 0#
arrayOfVariantOfDouble6(6) = 0#
arrayOfVariantOfDouble6(7) = 0#
arrayOfVariantOfDouble6(8) = 1#
Set sketch3bVariant = sketch3b
sketch3bVariant.SetAbsoluteAxisData arrayOfVariantOfDouble6

part2.InWorkObject = sketch3b

Dim factory2D3b As Factory2D
Set factory2D3b = sketch3b.OpenEdition()
Dim geometricElements3b As GeometricElements
Set geometricElements3b = sketch3b.GeometricElements
Dim axis2D3b As Axis2D
Set axis2D3b = geometricElements3b.Item("AbsoluteAxis")
Dim line2D5b As Line2D
Set line2D5b = axis2D3b.GetItem("HDirection")
line2D5b.ReportName = 1
Dim line2D6b As Line2D
Set line2D6b = axis2D3b.GetItem("VDirection")
line2D6b.ReportName = 2

Dim point2D3b As Point2D
Set point2D3b = factory2D3b.CreatePoint(0#, (D / 2 - d_ / 2) / 2 + d_ / 2)
point2D3b.ReportName = 3

Dim circle2D3b As Circle2D
Set circle2D3b = factory2D3b.CreateClosedCircle(0#, (D / 2 - d_ / 2) / 2 + d_ / 2, B / 1.555 / 2)
circle2D3b.CenterPoint = point2D3b
circle2D3b.ReportName = 4

Dim constraints3b As Constraints
Set constraints3b = sketch3b.Constraints
Dim reference6b As Reference
Set reference6b = part2.CreateReferenceFromObject(point2D3b)
Dim reference7b As Reference
Set reference7b = part2.CreateReferenceFromObject(line2D6b)
Dim constraint3b As Constraint
Set constraint3b = constraints3b.AddBiEltCst(catCstTypeOn, reference6b, reference7b)
constraint3b.Mode = catCstModeDrivingDimension

Dim reference8b As Reference
Set reference8b = part2.CreateReferenceFromObject(circle2D3b)
Dim constraint4b As Constraint
Set constraint4b = constraints3b.AddMonoEltCst(catCstTypeRadius, reference8b)
constraint4b.Mode = catCstModeDrivingDimension
Dim length4b As Length
Set length4b = constraint4b.Dimension
length4b.Value = B / 1.555 / 2

Dim reference9b As Reference
Set reference9b = part2.CreateReferenceFromObject(point2D3b)
Dim reference10b As Reference
Set reference10b = part2.CreateReferenceFromObject(line2D5b)
Dim constraint5b As Constraint
Set constraint5b = constraints3b.AddBiEltCst(catCstTypeDistance, reference9b, reference10b)
constraint5b.Mode = catCstModeDrivingDimension
Dim length5b As Length
Set length5b = constraint5b.Dimension
length5b.Value = (D / 2 - d_ / 2) / 2 + d_ / 2
sketch3b.CloseEdition
part2.InWorkObject = sketch3b
part2.Update

Dim sketch4b As Sketch
Set sketch4b = sketches2.Add(reference2a)

Set sketch4bVariant = sketch4b
sketch4bVariant.SetAbsoluteAxisData arrayOfVariantOfDouble4
part2.InWorkObject = sketch4b

Dim factory2D4b As Factory2D
Set factory2D4b = sketch4b.OpenEdition()
Dim geometricElements4b As GeometricElements
Set geometricElements4b = sketch4b.GeometricElements

Dim axis2D4b As Axis2D
Set axis2D4b = geometricElements4b.Item("AbsoluteAxis")
Dim line2D7b As Line2D
Set line2D7b = axis2D4b.GetItem("HDirection")
line2D7b.ReportName = 1
Dim line2D8b As Line2D
Set line2D8b = axis2D4b.GetItem("VDirection")
line2D8b.ReportName = 2
```

```
Dim circle2D4b As Circle2D
Set circle2D4b = factory2D4b.CreateClosedCircle(0#, 0#, (D / 2 - d_ / 2) / 2 + d_ / 2)
Dim point2D4b As Point2D
Set point2D4b = axis2D4b.GetItem("Origin")
circle2D4b.CenterPoint = point2D4b
circle2D4b.ReportName = 3

Dim constraints4b As Constraints
Set constraints4b = sketch4b.Constraints

Dim reference11b As Reference
Set reference11b = part2.CreateReferenceFromObject(circle2D4b)
Dim constraint6b As Constraint
Set constraint6b = constraints4b.AddMonoEltCst(catCstTypeRadius, reference11b)
constraint6b.Mode = catCstModeDrivingDimension
Dim length6b As Length
Set length6b = constraint6b.Dimension
length6b.Value = (D / 2 - d_ / 2) / 2 + d_ / 2
sketch4b.CloseEdition
part2.InWorkObject = sketch4b
part2.Update

'---SLOT---
Dim slot2 As Slot
Set slot2 = shapeFactory2.AddNewSlotFromRef(Nothing, Nothing)
Dim reference17 As Reference
Set reference17 = part2.CreateReferenceFromObject(sketch3b)
slot2.SetProfileElement reference17
Dim reference18 As Reference
Set reference18 = part2.CreateReferenceFromObject(sketch4b)
slot2.CenterCurveElement = reference18
part2.UpdateObject slot2
part2.Update

Dim product2 As Product
Set product2 = partDocument2.GetItem("Part2")
product2.PartNumber = "arointerno"

'------------GUARDADO---------
partDocument2.SaveAs objPath & "\aroint.CATPart"
partDocument2.Close

'=====================================
'--------------------------** BOLA ** -----
'=====================================
Dim documents3 As Documents
Dim partDocument3 As PartDocument
Dim part3 As Part
Set documents3 = CATIA.Documents
Set partDocument3 = documents3.Add("Part")
Set part3 = partDocument3.Part
Dim bodies3 As Bodies
Set bodies3 = part3.Bodies
Dim body3 As Body
Set body3 = bodies3.Item("PartBody")
Dim sketches3 As Sketches
Set sketches3 = body3.Sketches
Dim originElements3 As OriginElements
Set originElements3 = part3.OriginElements
Dim reference3b As Reference
Set reference3b = originElements3.PlaneYZ
Dim sketch7 As Sketch
Set sketch7 = sketches3.Add(reference3b)

Dim arrayOfVariantOfDouble7(8)
arrayOfVariantOfDouble7(0) = 0#
arrayOfVariantOfDouble7(1) = 0#
arrayOfVariantOfDouble7(2) = 0#
arrayOfVariantOfDouble7(3) = 0#
arrayOfVariantOfDouble7(4) = 1#
arrayOfVariantOfDouble7(5) = 0#
arrayOfVariantOfDouble7(6) = 0#
arrayOfVariantOfDouble7(7) = 0#
arrayOfVariantOfDouble7(8) = 1#
Set sketch7Variant = sketch7

'establece el sistema de ejes absolutos del sketch en 3D
sketch7Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble7
part3.InWorkObject = sketch7

Dim factory2D7 As Factory2D
Set factory2D7 = sketch7.OpenEdition()
Dim geometricElements7 As GeometricElements
Set geometricElements7 = sketch7.GeometricElements
Dim axis2D7 As Axis2D
Set axis2D7 = geometricElements7.Item("AbsoluteAxis")
Dim line2D15 As Line2D
Set line2D15 = axis2D7.GetItem("HDirection")
line2D15.ReportName = 1
Dim line2D16 As Line2D
Set line2D16 = axis2D7.GetItem("VDirection")
line2D16.ReportName = 2

'sketch de la media luna que luego revolucionaremos sobre el eje Z
Dim point2D7 As Point2D
Set point2D7 = factory2D7.CreatePoint(0#, d_ / 2 + (D / 2 - d_ / 2) / 2 - B / 1.555 / 2)
point2D7.ReportName = 3

Dim point2D8 As Point2D
Set point2D8 = factory2D7.CreatePoint(0#, d_ / 2 + (D / 2 - d_ / 2) / 2 + B / 1.555 / 2)
point2D8.ReportName = 4

Dim line2D17 As Line2D
Set line2D17 = factory2D7.CreateLine(0#, d_ / 2 + (D / 2 - d_ / 2) / 2 - B / 1.555 / 2, 0#, d_ / 2 + (D / 2 - d_ / 2) / 2 + B / 1.555 / 2)
line2D17.ReportName = 5
line2D17.StartPoint = point2D7
line2D17.EndPoint = point2D8
'sus constraints
Dim constraints7 As Constraints
Set constraints7 = sketch7.Constraints

Dim reference20 As Reference
Set reference20 = part3.CreateReferenceFromObject(point2D7)
Dim reference21 As Reference
```

```
Set reference21 = part3.CreateReferenceFromObject(line2D16)
Dim constraint9 As Constraint
Set constraint9 = constraints7.AddBiEltCst(catCstTypeOn, reference20, reference21)
constraint9.Mode = catCstModeDrivingDimension

Dim reference22 As Reference
Set reference22 = part3.CreateReferenceFromObject(line2D17)
Dim reference23 As Reference
Set reference23 = part3.CreateReferenceFromObject(line2D16)
Dim constraint10 As Constraint
Set constraint10 = constraints7.AddBiEltCst(catCstTypeVerticality, reference22, reference23)
constraint10.Mode = catCstModeDrivingDimension

Dim reference24 As Reference
Set reference24 = part3.CreateReferenceFromObject(point2D8)
Dim reference25 As Reference
Set reference25 = part3.CreateReferenceFromObject(line2D15)
Dim constraint11 As Constraint
Set constraint11 = constraints7.AddBiEltCst(catCstTypeDistance, reference24, reference25)
constraint11.Mode = catCstModeDrivingDimension
Dim length10 As Length
Set length10 = constraint11.Dimension
length10.Value = d_ / 2 + (D / 2 - d_ / 2) / 2 - B / 1.555 / 2

Dim reference26 As Reference
Set reference26 = part3.CreateReferenceFromObject(line2D17)
Dim constraint12 As Constraint
Set constraint12 = constraints7.AddMonoEltCst(catCstTypeLength, reference26)
constraint12.Mode = catCstModeDrivingDimension
Dim length11 As Length
Set length11 = constraint12.Dimension
length11.Value = B / 1.555

Dim point2D9 As Point2D
Set point2D9 = factory2D7.CreatePoint(0#, d_ / 2 + (D / 2 - d_ / 2) / 2)
point2D9.ReportName = 6

Dim circle2D7 As Circle2D
Set circle2D7 = factory2D7.CreateCircle(0#, d_ / 2 + (D / 2 - d_ / 2) / 2, B / 1.555 / 2, 4.712389, 7.853982)
circle2D7.CenterPoint = point2D9
circle2D7.ReportName = 7
circle2D7.StartPoint = point2D7
circle2D7.EndPoint = point2D8

Dim reference27 As Reference
Set reference27 = part3.CreateReferenceFromObject(point2D9)
Dim reference28 As Reference
Set reference28 = part3.CreateReferenceFromObject(line2D17)
Dim constraint13 As Constraint
Set constraint13 = constraints7.AddBiEltCst(catCstTypeMidPoint, reference27, reference28)
constraint13.Mode = catCstModeDrivingDimension

sketch7.CloseEdition
part3.InWorkObject = sketch7
part3.Update

'le hacemos un shaft a esta media luna
Dim shapeFactory7 As ShapeFactory
Set shapeFactory7 = part3.ShapeFactory
Dim shaft1 As Shaft
Set shaft1 = shapeFactory7.AddNewShaft(sketch7)
'selección del eje de revolución
Dim ejederev As Reference
Set ejederev = part3.CreateReferenceFromObject(line2D16)
shaft1.RevoluteAxis = ejederev
part3.Update

Dim referenceb As Reference
Set referenceb = part3.CreateReferenceFromName("")
Dim referenceb1 As Reference
Set referenceb1 = part3.CreateReferenceFromName("")
Dim circPatternb As CircPattern
Set circPatternb = shapeFactory7.AddNewCircPattern(Nothing, 1, 2, 20#, 45#, 1, 1, referenceb, referenceb1, True, 0#, True)
circPatternb.CircularPatternParameters = catInstancesandAngularSpacing
Dim angularRepartitionb As angularRepartition
Set angularRepartitionb = circPatternb.angularRepartition

'cálculo del número de agujeros para ello definimos primero el número pi
Const PIb = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es d_ / 2 + (D / 2 - d_ / 2) / 2
Dim longitudb As Double
longitudb = 2 * PIb * (d_ / 2 + (D / 2 - d_ / 2) / 2)
'crearemos una variable auxiliar que usara el comando matematico Round para redondear el numero de huecos que tendra el separador.
'para calcular este numero hacemos la hipotesis de que la distancia entre bolas será igual al diametro de una bola por lo que dividimos la
'variable longitud entre dos veces el diametro de una bola
Dim variableauxb As Double
variableauxb = Math.Round(longitudb / (2 * B / 1.555))
'ya tenemos el numero de huecos del separador
Dim intParamb As intParam
Set intParamb = angularRepartitionb.InstancesCount
intParamb.Value = variableauxb
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angleb As angle
Set angleb = angularRepartitionb.AngularSpacing
angleb.Value = 360 / variableauxb
```

```
circPatternb.SetRotationAxis reference3b

part3.UpdateObject circPatternb
part3.Update

Dim product3 As Product
Set product3 = partDocument3.GetItem("Part3")
product3.PartNumber = "bolas"

partDocument3.SaveAs objPath & "\bolas.CATPart"
partDocument3.Close

'====================================
'----------------------------**SEPARADOR**------------
'====================================

Dim documents4 As Documents
Dim partDocument4 As PartDocument
Dim part4 As Part
Set documents4 = CATIA.Documents
Set partDocument4 = documents4.Add("Part")
Set part4 = partDocument4.Part
Dim bodies4 As Bodies
Set bodies4 = part4.Bodies
Dim body4 As Body
Set body4 = bodies4.Item("PartBody")
Dim sketches4 As Sketches
Set sketches4 = body4.Sketches
Dim originElements4 As OriginElements
Set originElements4 = part4.OriginElements
Dim reference4b As Reference
Set reference4b = originElements4.PlaneYZ
Dim sketch8 As Sketch
Set sketch8 = sketches4.Add(reference4b)

Dim arrayOfVariantOfDouble8(8)
arrayOfVariantOfDouble8(0) = 0#
arrayOfVariantOfDouble8(1) = 0#
arrayOfVariantOfDouble8(2) = 0#
arrayOfVariantOfDouble8(3) = 0#
arrayOfVariantOfDouble8(4) = 1#
arrayOfVariantOfDouble8(5) = 0#
arrayOfVariantOfDouble8(6) = 0#
arrayOfVariantOfDouble8(7) = 0#
arrayOfVariantOfDouble8(8) = 1#
Set sketch8Variant = sketch8

'establece el sistema de ejes absolutos del sketch en 3D
sketch8Variant.SetAbsoluteAxisData arrayOfVariantOfDouble8
part4.InWorkObject = sketch8

Dim factory2D8 As Factory2D
Set factory2D8 = sketch8.OpenEdition()
Dim geometricElements8 As GeometricElements
Set geometricElements8 = sketch8.GeometricElements
Dim axis2D8 As Axis2D
Set axis2D8 = geometricElements8.Item("AbsoluteAxis")
Dim line2D118 As Line2D
Set line2D18 = axis2D8.GetItem("HDirection")
line2D18.ReportName = 1
Dim line2D19 As Line2D
Set line2D19 = axis2D8.GetItem("VDirection")
line2D19.ReportName = 2

'volvemos a hacer un anillo
Dim circle2D8 As Circle2D
Set circle2D8 = factory2D8.CreateClosedCircle(0#, 0#, (D / 2 - d_ / 2) / 2 + d_ / 2 + r1)
'el +2 lo pongo para que el separador no se quede a la mitad de la distancia entre anillos
'sino a 2 cm por encima de la mitad
Dim point2D10 As Point2D
Set point2D10 = axis2D8.GetItem("Origin")
circle2D8.CenterPoint = point2D10
circle2D8.ReportName = 3

Dim constraints8 As Constraints
Set constraints8 = sketch8.Constraints
Dim reference33 As Reference
Set reference33 = part4.CreateReferenceFromObject(circle2D8)
Dim constraint14 As Constraint
Set constraint14 = constraints8.AddMonoEltCst(catCstTypeRadius, reference33)
constraint14.Mode = catCstModeDrivingDimension
Dim length14 As Length
Set length14 = constraint14.Dimension
length14.Value = (D / 2 - d_ / 2) / 2 + d_ / 2 + r1

sketch8.CloseEdition
part4.InWorkObject = sketch8
part4.Update

'PAD
Dim shapeFactory8 As ShapeFactory
Set shapeFactory8 = part4.ShapeFactory
Dim pad3 As Pad
Set pad3 = shapeFactory8.AddNewPad(sketch8, B / 2)
pad3.IsSymmetric = True
pad3.Name = "separador"

Dim limit5 As Limit
Set limit5 = pad3.FirstLimit
Dim length15 As Length
Set length15 = limit5.Dimension
length15.Value = B / 2
part4.Update

'y el POCKET

Dim reference34 As Reference
Set reference34 = part4.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(separador;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")
Dim sketch9 As Sketch
Set sketch9 = sketches4.Add(reference34)
Set sketch4Variant = sketch9
Dim arrayOfVariantOfDouble9(8)
arrayOfVariantOfDouble9(0) = B / 2
arrayOfVariantOfDouble9(1) = 0#
arrayOfVariantOfDouble9(2) = 0#
arrayOfVariantOfDouble9(3) = 0#
arrayOfVariantOfDouble9(4) = 1#
arrayOfVariantOfDouble9(5) = 0#
arrayOfVariantOfDouble9(6) = 0#
arrayOfVariantOfDouble9(7) = 0#
```

```
arrayOfVariantOfDouble9(8) = 1#
Set sketch9Variant = sketch9
sketch4Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble9
part4.InWorkObject = sketch9

Dim factory2D9 As Factory2D
Set factory2D9 = sketch9.OpenEdition()
Dim geometricElements9 As GeometricElements
Set geometricElements9 = sketch9.GeometricElements
Dim axis2D9 As Axis2D
Set axis2D9 = geometricElements9.Item("AbsoluteAxis")
Dim line2D20 As Line2D
Set line2D20 = axis2D9.GetItem("HDirection")
line2D20.ReportName = 1
Dim line2D21 As Line2D
Set line2D21 = axis2D9.GetItem("VDirection")
line2D21.ReportName = 2

Dim circle2D9 As Circle2D
Set circle2D9 = factory2D9.CreateClosedCircle(0#, 0#, (D / 2 - d_ / 2) / 2 + d_ / 2 - r1)
Dim point2D11 As Point2D
Set point2D11 = axis2D9.GetItem("Origin")
circle2D9.CenterPoint = point2D11
circle2D9.ReportName = 3
Dim constraints10 As Constraints
Set constraints10 = sketch9.Constraints

Dim reference35 As Reference
Set reference35 = part4.CreateReferenceFromObject(circle2D9)
Dim constraint15 As Constraint
Set constraint15 = constraints10.AddMonoEltCst(catCstTypeRadius, reference35)
constraint15.Mode = catCstModeDrivingDimension
Dim length16 As Length
Set length16 = constraint15.Dimension
length16.Value = (D / 2 - d_ / 2) / 2 + d_ / 2 - r1

sketch9.CloseEdition
part4.InWorkObject = sketch9
part4.Update

Dim pocket3 As Pocket
Set pocket3 = shapeFactory8.AddNewPocket(sketch9, B / 2)
Dim limit6 As Limit
Set limit6 = pocket3.FirstLimit
limit6.LimitMode = catUpThruNextLimit
part4.Update

'agujeros del separador
Dim reference36 As Reference
Set reference36 = originElements4.PlaneXY
Dim sketch10 As Sketch
Set sketch10 = sketches4.Add(reference36)
Dim arrayOfVariantOfDouble10(8)
arrayOfVariantOfDouble10(0) = 0#
arrayOfVariantOfDouble10(1) = 0#
arrayOfVariantOfDouble10(2) = 0#
arrayOfVariantOfDouble10(3) = 1#
arrayOfVariantOfDouble10(4) = 0#
arrayOfVariantOfDouble10(5) = 0#
arrayOfVariantOfDouble10(6) = 0#
arrayOfVariantOfDouble10(7) = 1#
arrayOfVariantOfDouble10(8) = 0#
Set sketch10Variant = sketch10
sketch10Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble10
part4.InWorkObject = sketch10

'creamos un circulo en el plano XY que luego haremos un pocket hacia arriba que perfore el separador
Dim factory2D10 As Factory2D
Set factory2D10 = sketch10.OpenEdition()
Dim geometricElements10 As GeometricElements
Set geometricElements10 = sketch10.GeometricElements
Dim axis2D10 As Axis2D
Set axis2D10 = geometricElements10.Item("AbsoluteAxis")
Dim line2D22 As Line2D
Set line2D22 = axis2D10.GetItem("HDirection")
line2D22.ReportName = 1
Dim line2D23 As Line2D
Set line2D23 = axis2D10.GetItem("VDirection")
line2D23.ReportName = 2

Dim circle2D10 As Circle2D
Set circle2D10 = factory2D10.CreateClosedCircle(0#, 0#, B / 1.555 / 2)
Dim point2D12 As Point2D
Set point2D12 = axis2D10.GetItem("Origin")
circle2D10.CenterPoint = point2D12
circle2D10.ReportName = 3

Dim constraints11 As Constraints
Set constraints11 = sketch10.Constraints
Dim reference37 As Reference
Set reference37 = part4.CreateReferenceFromObject(circle2D10)
Dim constraint17 As Constraint
Set constraint17 = constraints11.AddMonoEltCst(catCstTypeRadius, reference37)
constraint17.Mode = catCstModeDrivingDimension
Dim length17 As Length
Set length17 = constraint17.Dimension
length17.Value = B / 1.555 / 2

sketch10.CloseEdition
part4.InWorkObject = sketch10
part4.Update

Dim shapeFactory9 As ShapeFactory
Set shapeFactory9 = part4.ShapeFactory
Dim pocket4 As Pocket
Set pocket4 = shapeFactory9.AddNewPocket(sketch10, B / 2)
pocket4.DirectionOrientation = catRegularOrientation
Dim limit7 As Limit
Set limit7 = pocket4.FirstLimit
limit7.LimitMode = catUpToLastLimit

part4.Update

'PATRON CIRCULAR
```

```
Dim reference38 As Reference
Set reference38 = part4.CreateReferenceFromName("")
Dim reference39 As Reference
Set reference39 = part4.CreateReferenceFromName("")
Dim circPattern1 As CircPattern
Set circPattern1 = shapeFactory9.AddNewCircPattern(pocket4, 1, 2, 20#, 45#, 1, 1, reference38, reference39, True, 0#, True)
circPattern1.CircularPatternParameters = catInstancesandAngularSpacing
Dim angularRepartition As angularRepartition
Set angularRepartition = circPattern1.angularRepartition

'calculo del número de agujeros para ello definimos primero el número pi
Const PI = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es d_ / 2 + (D / 2 - d_ / 2) / 2
Dim longitud As Double
longitud = 2 * PI * (d_ / 2 + (D / 2 - d_ / 2) / 2)
'crearemos una variable auxiliar que usara el comando matematico Round para redondear el numero de huecos que tendra el separador.
'para calcular este numero hacemos la hipotesis de que la distancia entre bolas será igual al diametro de una bola por lo que dividimos la
'variable longitud entre dos veces el diametro de una bola
Dim variableaux As Double
variableaux = Math.Round(longitud / (2 * B / 1.555))
'ya tenemos el numero de huecos del separador
Dim intParam As intParam
Set intParam = angularRepartition.InstancesCount
intParam.Value = variableaux
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angle As angle
Set angle = angularRepartition.AngularSpacing
angle.Value = 360 / variableaux

Dim reference40 As Reference
Set reference40 = part4.CreateReferenceFromObject(line2D22)
circPattern1.SetRotationAxis reference40

part4.UpdateObject circPattern1
part4.Update

Dim product4 As Product
Set product4 = partDocument4.GetItem("Part4")
product4.PartNumber = "jaula"

partDocument4.SaveAs objPath & "\jaula.CATPart"
partDocument4.Close

'======================================
'----------------------------ENSAMBLAJE----------------
'======================================
Dim documents5 As Documents
Set documents5 = CATIA.Documents

Dim productDocument1 As ProductDocument
Set productDocument1 = documents5.Add("Product")

Dim product5 As Product
Set product5 = productDocument1.Product

Dim products1 As Products
Set products1 = product5.Products


Dim arrayOfVariantOfBSTR1(0)
arrayOfVariantOfBSTR1(0) = objPath & "\aroext.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR1, "All"
Set productDocument1 = CATIA.ActiveDocument

Dim constraints12 As Constraints
Set constraints12 = product5.Connections("CATIAConstraints")
'Fix del aro externo
Dim reference41 As Reference
Set reference41 = product5.CreateReferenceFromName("Product5/aroexterno.1/!Product5/aroext.1/")
Dim constraint18 As Constraint
Set constraint18 = constraints12.AddMonoEltCst(catCstTypeReference, reference41)

Dim arrayOfVariantOfBSTR2(0)
arrayOfVariantOfBSTR2(0) = objPath & "\aroint.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR2, "All"

'coincidencia de ejes de los aros externo e interno
Dim reference42 As Reference
Set reference42 = product5.CreateReferenceFromName("Product5/aroinnterno.1/!Axis:(Selection_RSur:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());Slot.1_ResultOUT;Z0;G3055))")
Dim reference43 As Reference
Set reference43 = product5.CreateReferenceFromName("Product5/aroexterno.1/!Axis:(Selection_RSur:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());Slot.1_ResultOUT;Z0;G3055))")
Dim constraint19 As Constraint
Set constraint19 = constraints12.AddBiEltCst(catCstTypeOn, reference42, reference43)

Dim arrayOfVariantOfBSTR3(0)
arrayOfVariantOfBSTR3(0) = objPath & "\bolas.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR3, "All"

Dim arrayOfVariantOfBSTR4(0)
arrayOfVariantOfBSTR4(0) = objPath & "\jaula.catpart"
Set products1Variant = products1
```

```
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR4, "All"

'coincidencia de ejes aro externo y jaula
Dim reference44 As Reference
Set reference44 =
product5.CreateReferenceFromName("Product5/jaula
.1/!Axis:(Selection_RSur:(Face:(Brp:(Pocket.1;0:(Br
p:(Sketch.2;3)));None:();Cf11:());CircPattern.1_Resu
ltOUT;Z0;G3055))")
Dim constraint20 As Constraint
Set constraint20 =
constraints12.AddBiEltCst(catCstTypeOn,
reference43, reference44)

'Fijamos las bolas a los huecos de la jaula.
Dim referencec1 As Reference
Set referencec1 =
product5.CreateReferenceFromName("Product5/bola
s.1/!zx plane")
Dim referencec2 As Reference
Set referencec2 =
product5.CreateReferenceFromName("Product5/jaula
.1/!zx plane")
Dim constraintc1 As Constraint
Set constraintc1 =
constraints12.AddBiEltCst(catCstTypeDistance,
referencec1, referencec2)

Dim referencec3 As Reference
Set referencec3 =
product5.CreateReferenceFromName("Product5/bola
s.1/!yz plane")
Dim referencec4 As Reference
Set referencec4 =
product5.CreateReferenceFromName("Product5/jaula
.1/!yz plane")
Dim constraintc2 As Constraint
Set constraintc2 =
constraints12.AddBiEltCst(catCstTypeDistance,
referencec3, referencec4)

Dim referencec5 As Reference
Set referencec5 =
product5.CreateReferenceFromName("Product5/bola
s.1/!xy plane")
Dim referencec6 As Reference
Set referencec6 =
product5.CreateReferenceFromName("Product5/jaula
.1/!xy plane")
Dim constraintc3 As Constraint
Set constraintc3 =
constraints12.AddBiEltCst(catCstTypeDistance,
referencec5, referencec6)

product5.Update

'fijamos la distancia entre la cara lateral del aro
externo y el interno
Dim reference49 As Reference
Set reference49 =
product5.CreateReferenceFromName("Product5/aroe
xterno.1/!Selection_RSur:(Face:(Brp:(Pad.1;2);None:
();Cf11:());Slot.1_ResultOUT;Z0;G3055)")
Dim reference50 As Reference
Set reference50 =
product5.CreateReferenceFromName("Product5/aroi
nterno.1/!Selection_RSur:(Face:(Brp:(Pad.1;2);None:
();Cf11:());Slot.1_ResultOUT;Z0;G3055)")
Dim constraint23 As Constraint
Set constraint23 =
constraints12.AddBiEltCst(catCstTypeDistance,
reference49, reference50)
Dim length19 As Length
Set length19 = constraint23.Dimension
length19.Value = 0#
constraint23.Orientation = catCstOrientSame

'idem para el aro externo y la jaula
Dim reference51 As Reference
Set reference51 =
product5.CreateReferenceFromName("Product5/jaula
.1/!Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf1
1:());CircPattern.1_ResultOUT;Z0;G3055)")
Dim constraint24 As Constraint
Set constraint24 =
constraints12.AddBiEltCst(catCstTypeDistance,
reference49, reference51)
Dim length20 As Length
Set length20 = constraint24.Dimension
length20.Value = 0#
constraint24.Orientation = catCstOrientSame
product1.Update

product5.Update
'productDocument1.Close

End Sub
```

- **Rodamiento de bolas con contacto angular.**

```
'=====================================
'----------------------------------**Aro externo**-
'=====================================
'Circulo guía del Rib
Dim documents1 As Documents
Set documents1 = CATIA.Documents
Dim partDocument1 As PartDocument
Set partDocument1 = documents1.Add("Part")
Dim part1 As Part
Set part1 = partDocument1.Part
Dim bodies1 As Bodies
Set bodies1 = part1.Bodies
Dim body1 As Body
Set body1 = bodies1.Item("PartBody")
Dim sketches1 As Sketches
Set sketches1 = body1.Sketches
Dim originElements1 As OriginElements
Set originElements1 = part1.OriginElements
Dim reference1 As Reference
Set reference1 = originElements1.PlaneYZ
Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0#
arrayOfVariantOfDouble1(1) = 0#
arrayOfVariantOfDouble1(2) = 0#
arrayOfVariantOfDouble1(3) = 0#
arrayOfVariantOfDouble1(4) = 1#
arrayOfVariantOfDouble1(5) = 0#
arrayOfVariantOfDouble1(6) = 0#
arrayOfVariantOfDouble1(7) = 0#
arrayOfVariantOfDouble1(8) = 1#
Set sketch1Variant = sketch1
sketch1Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble1
part1.InWorkObject = sketch1

Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()
Dim geometricElements1 As GeometricElements
Set geometricElements1 = sketch1.GeometricElements
Dim axis2D1 As Axis2D
Set axis2D1 = geometricElements1.Item("AbsoluteAxis")
Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2

Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateClosedCircle(0#, 0#, D / 2)
Dim point2D1 As Point2D
Set point2D1 = axis2D1.GetItem("Origin")
circle2D1.CenterPoint = point2D1
circle2D1.ReportName = 3

Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints
Dim reference2 As Reference
Set reference2 = part1.CreateReferenceFromObject(circle2D1)
Dim constraint1 As Constraint
Set constraint1 = constraints1.AddMonoEltCst(catCstTypeRadius, reference2)
constraint1.Mode = catCstModeDrivingDimension
Dim length1 As Length
Set length1 = constraint1.Dimension
length1.Value = D / 2

sketch1.CloseEdition
part1.InWorkObject = sketch1
part1.Update

'Sketch que seguirá la guía de antes y formará el aro externo
Dim reference3 As Reference
Set reference3 = originElements1.PlaneZX
Dim sketch2 As Sketch
Set sketch2 = sketches1.Add(reference3)

Dim arrayOfVariantOfDouble2(8)
arrayOfVariantOfDouble2(0) = 0#
arrayOfVariantOfDouble2(1) = 0#
arrayOfVariantOfDouble2(2) = 0#
arrayOfVariantOfDouble2(3) = -1#
arrayOfVariantOfDouble2(4) = 0#
arrayOfVariantOfDouble2(5) = 0#
arrayOfVariantOfDouble2(6) = 0#
arrayOfVariantOfDouble2(7) = -0#
arrayOfVariantOfDouble2(8) = 1#
Set sketch2Variant = sketch2
sketch2Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble2
part1.InWorkObject = sketch2

Dim factory2D2 As Factory2D
Set factory2D2 = sketch2.OpenEdition()
Dim geometricElements2 As GeometricElements
Set geometricElements2 = sketch2.GeometricElements
Dim axis2D2 As Axis2D
Set axis2D2 = geometricElements2.Item("AbsoluteAxis")
Dim line2D3 As Line2D
Set line2D3 = axis2D2.GetItem("HDirection")
line2D3.ReportName = 3
Dim line2D4 As Line2D
Set line2D4 = axis2D2.GetItem("VDirection")
line2D4.ReportName = 4

Dim point2D2 As Point2D
Set point2D2 = factory2D2.CreatePoint(B / 2, D / 2)
point2D2.ReportName = 5
Dim point2D3 As Point2D
Set point2D3 = factory2D2.CreatePoint(-B / 2, D / 2)
point2D3.ReportName = 6
Dim line2D5 As Line2D
Set line2D5 = factory2D2.CreateLine(B / 2, D / 2, -B / 2, D / 2)
line2D5.ReportName = 7
line2D5.StartPoint = point2D2
line2D5.EndPoint = point2D3

Dim constraints2 As Constraints
Set constraints2 = sketch2.Constraints
```

```
Dim reference4 As Reference
Set reference4 = part1.CreateReferenceFromObject(line2D5)
Dim reference5 As Reference
Set reference5 = part1.CreateReferenceFromObject(line2D3)
Dim constraint2 As Constraint
Set constraint2 = constraints2.AddBiEltCst(catCstTypeHorizontality, reference4, reference5)
constraint2.Mode = catCstModeDrivingDimension

Dim reference6 As Reference
Set reference6 = part1.CreateReferenceFromObject(line2D5)
Dim constraint3 As Constraint
Set constraint3 = constraints2.AddMonoEltCst(catCstTypeLength, reference6)
constraint3.Mode = catCstModeDrivingDimension
Dim length2 As Length
Set length2 = constraint3.Dimension
length2.Value = B

Dim reference7 As Reference
Set reference7 = part1.CreateReferenceFromObject(point2D2)
Dim reference8 As Reference
Set reference8 = part1.CreateReferenceFromObject(line2D3)
Dim constraint4 As Constraint
Set constraint4 = constraints2.AddBiEltCst(catCstTypeDistance, reference7, reference8)
constraint4.Mode = catCstModeDrivingDimension
Dim length3 As Length
Set length3 = constraint4.Dimension
length3.Value = D / 2

Dim reference9 As Reference
Set reference9 = part1.CreateReferenceFromObject(point2D2)
Dim reference10 As Reference
Set reference10 = part1.CreateReferenceFromObject(line2D4)
Dim constraint5 As Constraint
Set constraint5 = constraints2.AddBiEltCst(catCstTypeDistance, reference9, reference10)
constraint5.Mode = catCstModeDrivingDimension
Dim length4 As Length
Set length4 = constraint5.Dimension
length4.Value = B / 2

Dim point2D4 As Point2D
Set point2D4 = factory2D2.CreatePoint(-B / 2, D1 / 2)
point2D4.ReportName = 8
Dim line2D6 As Line2D
Set line2D6 = factory2D2.CreateLine(-B / 2, D / 2, -B / 2, D1 / 2)
line2D6.ReportName = 9
line2D6.StartPoint = point2D3
line2D6.EndPoint = point2D4

Dim reference11 As Reference
Set reference11 = part1.CreateReferenceFromObject(line2D6)
Dim reference12 As Reference
Set reference12 = part1.CreateReferenceFromObject(line2D4)
Dim constraint6 As Constraint
Set constraint6 = constraints2.AddBiEltCst(catCstTypeVerticality, reference11, reference12)
constraint6.Mode = catCstModeDrivingDimension

Dim reference13 As Reference
Set reference13 = part1.CreateReferenceFromObject(line2D6)
Dim constraint7 As Constraint
Set constraint7 = constraints2.AddMonoEltCst(catCstTypeLength, reference13)
constraint7.Mode = catCstModeDrivingDimension
Dim length5 As Length
Set length5 = constraint7.Dimension
length5.Value = D / 2 - D1 / 2

Dim point2D5 As Point2D
Set point2D5 = factory2D2.CreatePoint(A - B / 2, 0#)
point2D5.ReportName = 10
Dim point2D6 As Point2D
Set point2D6 = factory2D2.CreatePoint(0, (D / 2 - d_ / 2) / 2 + d_ / 2)
point2D6.ReportName = 11
Dim line2D7 As Line2D
Set line2D7 = factory2D2.CreateLine(A - B / 2, 0#, 0, (D / 2 - d_ / 2) / 2 + d_ / 2)
line2D7.ReportName = 12
line2D7.Construction = True
line2D7.StartPoint = point2D5
line2D7.EndPoint = point2D6

Dim reference14 As Reference
Set reference14 = part1.CreateReferenceFromObject(point2D5)
Dim reference15 As Reference
Set reference15 = part1.CreateReferenceFromObject(line2D3)
Dim constraint8 As Constraint
Set constraint8 = constraints2.AddBiEltCst(catCstTypeOn, reference14, reference15)
constraint8.Mode = catCstModeDrivingDimension

Dim reference16 As Reference
Set reference16 = part1.CreateReferenceFromObject(point2D5)
Dim reference17 As Reference
Set reference17 = part1.CreateReferenceFromObject(line2D6)
Dim constraint9 As Constraint
Set constraint9 = constraints2.AddBiEltCst(catCstTypeDistance, reference16, reference17)
constraint9.Mode = catCstModeDrivingDimension
Dim length6 As Length
Set length6 = constraint9.Dimension
length6.Value = A

Dim reference18 As Reference
```

```
Set reference18 = part1.CreateReferenceFromObject(point2D6)
Dim reference19 As Reference
Set reference19 = part1.CreateReferenceFromObject(line2D3)
Dim constraint10 As Constraint
Set constraint10 = constraints2.AddBiEltCst(catCstTypeDistance, reference18, reference19)
constraint10.Mode = catCstModeDrivingDimension
Dim length7 As Length
Set length7 = constraint10.Dimension
length7.Value = (D / 2 - d_ / 2) / 2 + d_ / 2

Dim point2D7 As Point2D
Set point2D7 = factory2D2.CreatePoint(0, (D / 2 - d_ / 2) / 2 + d_ / 2 + B / 1.555 / 2)
point2D7.ReportName = 13

Dim point2D8 As Point2D
Set point2D8 = factory2D2.CreatePoint(-B / 1.555 / 2, (D / 2 - d_ / 2) / 2 + d_ / 2)
point2D8.ReportName = 14

Dim circle2D2 As Circle2D
Set circle2D2 = factory2D2.CreateCircle(0, (D / 2 - d_ / 2) / 2 + d_ / 2, B / 1.555 / 2, 1.570796, 2.614424)
circle2D2.CenterPoint = point2D6
circle2D2.ReportName = 15
circle2D2.StartPoint = point2D7
circle2D2.EndPoint = point2D8

Dim reference20 As Reference
Set reference20 = part1.CreateReferenceFromObject(circle2D2)
Dim constraint11 As Constraint
Set constraint11 = constraints2.AddMonoEltCst(catCstTypeRadius, reference20)
constraint11.Mode = catCstModeDrivingDimension
Dim length8 As Length
Set length8 = constraint11.Dimension
length8.Value = B / 1.555 / 2

Dim line2D8 As Line2D
Set line2D8 = factory2D2.CreateLine(-B / 2, D1 / 2, -5, D1 / 2)
line2D8.ReportName = 16
line2D8.StartPoint = point2D4
line2D8.EndPoint = point2D8

Dim reference21 As Reference
Set reference21 = part1.CreateReferenceFromObject(line2D8)
Dim reference22 As Reference
Set reference22 = part1.CreateReferenceFromObject(line2D3)
Dim constraint12 As Constraint
Set constraint12 = constraints2.AddBiEltCst(catCstTypeHorizontality, reference21, reference22)
constraint12.Mode = catCstModeDrivingDimension

Dim point2D9 As Point2D
Set point2D9 = factory2D2.CreatePoint(B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2 + 5)
point2D9.ReportName = 17

Dim line2D9 As Line2D
Set line2D9 = factory2D2.CreateLine(0, (D / 2 - d_ / 2) / 2 + d_ / 2 + 5, B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2 + 5)
line2D9.ReportName = 18
line2D9.StartPoint = point2D7
line2D9.EndPoint = point2D9

Dim reference23 As Reference
Set reference23 = part1.CreateReferenceFromObject(line2D7)
Dim reference24 As Reference
Set reference24 = part1.CreateReferenceFromObject(line2D3)
Dim constraint13 As Constraint
Set constraint13 = constraints2.AddBiEltCst(catCstTypeAngle, reference23, reference24)
constraint13.Mode = catCstModeDrivingDimension
constraint13.AngleSector = catCstAngleSector1
Dim angle1 As angle
Set angle1 = constraint13.Dimension
angle1.Value = 50#

Dim reference25 As Reference
Set reference25 = part1.CreateReferenceFromObject(line2D9)
Dim reference26 As Reference
Set reference26 = part1.CreateReferenceFromObject(line2D3)
Dim constraint14 As Constraint
Set constraint14 = constraints2.AddBiEltCst(catCstTypeHorizontality, reference25, reference26)
constraint14.Mode = catCstModeDrivingDimension

Dim reference27 As Reference
Set reference27 = part1.CreateReferenceFromObject(circle2D2)
Dim reference28 As Reference
Set reference28 = part1.CreateReferenceFromObject(line2D9)
Dim constraint15 As Constraint
Set constraint15 = constraints2.AddBiEltCst(catCstTypeTangency, reference27, reference28)
constraint15.Mode = catCstModeDrivingDimension

Dim line2D10 As Line2D
Set line2D10 = factory2D2.CreateLine(B / 2, D / 2, B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2 + 5)
line2D10.ReportName = 19
line2D10.StartPoint = point2D2
line2D10.EndPoint = point2D9

Dim reference29 As Reference
Set reference29 = part1.CreateReferenceFromObject(line2D10)
Dim reference30 As Reference
Set reference30 = part1.CreateReferenceFromObject(line2D4)
Dim constraint16 As Constraint
Set constraint16 = constraints2.AddBiEltCst(catCstTypeVerticality, reference29, reference30)
constraint16.Mode = catCstModeDrivingDimension
```

```
sketch2.CloseEdition
part1.InWorkObject = sketch2
part1.Update

Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory
Dim rib1 As Rib
Set rib1 = shapeFactory1.AddNewRibFromRef(Nothing, Nothing)
Dim reference31 As Reference
Set reference31 = part1.CreateReferenceFromObject(sketch2)
rib1.SetProfileElement reference31
Dim reference32 As Reference
Set reference32 = part1.CreateReferenceFromObject(sketch1)
rib1.CenterCurveElement = reference32

part1.Update

'Chaflán
Dim reference33 As Reference
Set reference33 = part1.CreateReferenceFromName("")

Dim constRadEdgeFillet1 As ConstRadEdgeFillet
Set constRadEdgeFillet1 = shapeFactory1.AddNewSolidEdgeFilletWithConstantRadius(reference33, catTangencyFilletEdgePropagation, r)
Dim reference34 As Reference
Set reference34 = part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;7)));None:();Cf11:());Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;9)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", rib1)
constRadEdgeFillet1.AddObjectToFillet reference34
constRadEdgeFillet1.EdgePropagation = catTangencyFilletEdgePropagation
part1.Update

Dim reference35 As Reference
Set reference35 = part1.CreateReferenceFromName("")

Dim constRadEdgeFillet2 As ConstRadEdgeFillet
Set constRadEdgeFillet2 = shapeFactory1.AddNewSolidEdgeFilletWithConstantRadius(reference35, catTangencyFilletEdgePropagation, r1)
Dim reference36 As Reference
Set reference36 = part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;19)));None:();Cf11:());Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;7)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", constRadEdgeFillet1)
constRadEdgeFillet2.AddObjectToFillet reference36
constRadEdgeFillet2.EdgePropagation = catTangencyFilletEdgePropagation
part1.Update

Dim product1 As Product
Set product1 = partDocument1.GetItem("Part1")
product1.PartNumber = "aroexterno"

partDocument1.SaveAs objPath & "\aroext.catpart"
partDocument1.Close

'====================================
'--------------------------------**Aro interno**-------------
'====================================
'guía del Rib
Dim documents2 As Documents
Set documents2 = CATIA.Documents
Dim partDocument2 As PartDocument
Set partDocument2 = documents2.Add("Part")
Dim part2 As Part
Set part2 = partDocument2.Part
Dim bodies2 As Bodies
Set bodies2 = part2.Bodies
Dim body2 As Body
Set body2 = bodies2.Item("PartBody")
Dim sketches2 As Sketches
Set sketches2 = body2.Sketches
Dim originElements2 As OriginElements
Set originElements2 = part2.OriginElements
Dim reference37 As Reference
Set reference37 = originElements2.PlaneYZ
Dim sketch3 As Sketch
Set sketch3 = sketches2.Add(reference37)

Dim arrayOfVariantOfDouble3(8)
arrayOfVariantOfDouble3(0) = 0#
arrayOfVariantOfDouble3(1) = 0#
arrayOfVariantOfDouble3(2) = 0#
arrayOfVariantOfDouble3(3) = 0#
arrayOfVariantOfDouble3(4) = 1#
arrayOfVariantOfDouble3(5) = 0#
arrayOfVariantOfDouble3(6) = 0#
arrayOfVariantOfDouble3(7) = 0#
arrayOfVariantOfDouble3(8) = 1#
Set sketch3Variant = sketch3
sketch3Variant.SetAbsoluteAxisData arrayOfVariantOfDouble3
part2.InWorkObject = sketch3

Dim factory2D3 As Factory2D
Set factory2D3 = sketch3.OpenEdition()
Dim geometricElements3 As GeometricElements
Set geometricElements3 = sketch3.GeometricElements
Dim axis2D3 As Axis2D
Set axis2D3 = geometricElements3.Item("AbsoluteAxis")
Dim line2D11 As Line2D
Set line2D11 = axis2D3.GetItem("HDirection")
line2D11.ReportName = 1
Dim line2D12 As Line2D
Set line2D12 = axis2D3.GetItem("VDirection")
line2D12.ReportName = 2

Dim circle2D3 As Circle2D
Set circle2D3 = factory2D3.CreateClosedCircle(0#, 0#, d_ / 2)
Dim point2D10 As Point2D
```

```
Set point2D10 = axis2D3.GetItem("Origin")
circle2D3.CenterPoint = point2D10
circle2D3.ReportName = 3

Dim constraints3 As Constraints
Set constraints3 = sketch3.Constraints
Dim reference38 As Reference
Set reference38 = part2.CreateReferenceFromObject(circle2D3)
Dim constraint17 As Constraint
Set constraint17 = constraints3.AddMonoEltCst(catCstTypeRadius, reference38)
constraint17.Mode = catCstModeDrivingDimension
Dim length9 As Length
Set length9 = constraint17.Dimension
length9.Value = d_ / 2

sketch3.CloseEdition
part2.InWorkObject = sketch3
part2.Update

'Sketch que seguirá la guía para dar forma al aro
Dim reference39 As Reference
Set reference39 = originElements2.PlaneZX
Dim sketch4 As Sketch
Set sketch4 = sketches2.Add(reference39)

Dim arrayOfVariantOfDouble4(8)
arrayOfVariantOfDouble4(0) = 0#
arrayOfVariantOfDouble4(1) = 0#
arrayOfVariantOfDouble4(2) = 0#
arrayOfVariantOfDouble4(3) = -1#
arrayOfVariantOfDouble4(4) = 0#
arrayOfVariantOfDouble4(5) = 0#
arrayOfVariantOfDouble4(6) = 0#
arrayOfVariantOfDouble4(7) = -0#
arrayOfVariantOfDouble4(8) = 1#
Set sketch4Variant = sketch4
sketch4Variant.SetAbsoluteAxisData arrayOfVariantOfDouble4
part2.InWorkObject = sketch4

Dim factory2D4 As Factory2D
Set factory2D4 = sketch4.OpenEdition()
Dim geometricElements4 As GeometricElements
Set geometricElements4 = sketch4.GeometricElements
Dim axis2D4 As Axis2D
Set axis2D4 = geometricElements4.Item("AbsoluteAxis")
Dim line2D13 As Line2D
Set line2D13 = axis2D4.GetItem("HDirection")
line2D13.ReportName = 2
Dim line2D14 As Line2D
Set line2D14 = axis2D4.GetItem("VDirection")
line2D14.ReportName = 3

Dim point2D11 As Point2D
Set point2D11 = factory2D4.CreatePoint(-B / 2, d_ / 2)
point2D11.ReportName = 4
Dim point2D12 As Point2D
Set point2D12 = factory2D4.CreatePoint(B / 2, d_ / 2)
point2D12.ReportName = 5
Dim line2D15 As Line2D
Set line2D15 = factory2D4.CreateLine(-B / 2, d_ / 2, B / 2, d_ / 2)
line2D15.ReportName = 6
line2D15.StartPoint = point2D11
line2D15.EndPoint = point2D12

Dim constraints4 As Constraints
Set constraints4 = sketch4.Constraints
Dim reference40 As Reference
Set reference40 = part2.CreateReferenceFromObject(line2D15)
Dim reference41 As Reference
Set reference41 = part2.CreateReferenceFromObject(line2D13)
Dim constraint18 As Constraint
Set constraint18 = constraints4.AddBiEltCst(catCstTypeHorizontality, reference40, reference41)
constraint18.Mode = catCstModeDrivingDimension

Dim reference42 As Reference
Set reference42 = part2.CreateReferenceFromObject(line2D15)
Dim reference43 As Reference
Set reference43 = part2.CreateReferenceFromObject(line2D13)
Dim constraint19 As Constraint
Set constraint19 = constraints4.AddBiEltCst(catCstTypeDistance, reference42, reference43)
constraint19.Mode = catCstModeDrivingDimension
Dim length10 As Length
Set length10 = constraint19.Dimension
length10.Value = d_ / 2

Dim reference44 As Reference
Set reference44 = part2.CreateReferenceFromObject(line2D14)
Dim reference45 As Reference
Set reference45 = part2.CreateReferenceFromObject(point2D12)
Dim constraint20 As Constraint
Set constraint20 = constraints4.AddBiEltCst(catCstTypeDistance, reference44, reference45)
constraint20.Mode = catCstModeDrivingDimension
Dim length11 As Length
Set length11 = constraint20.Dimension
length11.Value = B / 2

Dim reference46 As Reference
Set reference46 = part2.CreateReferenceFromObject(line2D15)
Dim constraint21 As Constraint
Set constraint21 = constraints4.AddMonoEltCst(catCstTypeLength, reference46)
constraint21.Mode = catCstModeDrivingDimension
Dim length12 As Length
Set length12 = constraint21.Dimension
length12.Value = B

Dim point2D13 As Point2D
Set point2D13 = factory2D4.CreatePoint(B / 2, d_1 / 2)
point2D13.ReportName = 7
Dim line2D16 As Line2D
```

```
Set line2D16 = factory2D4.CreateLine(B / 2, d_ / 2, B / 2, d_1 / 2)
line2D16.ReportName = 8
line2D16.StartPoint = point2D12
line2D16.EndPoint = point2D13

Dim reference47 As Reference
Set reference47 = part2.CreateReferenceFromObject(line2D16)
Dim reference48 As Reference
Set reference48 = part2.CreateReferenceFromObject(line2D14)
Dim constraint22 As Constraint
Set constraint22 = constraints4.AddBiEltCst(catCstTypeVerticality, reference47, reference48)
constraint22.Mode = catCstModeDrivingDimension

Dim reference49 As Reference
Set reference49 = part2.CreateReferenceFromObject(line2D16)
Dim constraint23 As Constraint
Set constraint23 = constraints4.AddMonoEltCst(catCstTypeLength, reference49)
constraint23.Mode = catCstModeDrivingDimension
Dim length13 As Length
Set length13 = constraint23.Dimension
length13.Value = d_1 / 2 - d_ / 2

Dim point2D14 As Point2D
Set point2D14 = factory2D4.CreatePoint(A - B / 2, 0#)
point2D14.ReportName = 9
Dim point2D15 As Point2D
Set point2D15 = factory2D4.CreatePoint(0, (D / 2 - d_ / 2) / 2 + d_ / 2)
point2D15.ReportName = 10
Dim line2D17 As Line2D
Set line2D17 = factory2D4.CreateLine(A - B / 2, 0, 0, (D / 2 - d_ / 2) / 2 + d_ / 2)
line2D17.ReportName = 11
line2D17.Construction = True
line2D17.StartPoint = point2D14
line2D17.EndPoint = point2D15

Dim reference50 As Reference
Set reference50 = part2.CreateReferenceFromObject(point2D14)
Dim reference51 As Reference
Set reference51 = part2.CreateReferenceFromObject(line2D13)
Dim constraint24 As Constraint
Set constraint24 = constraints4.AddBiEltCst(catCstTypeOn, reference50, reference51)
constraint24.Mode = catCstModeDrivingDimension

Dim reference52 As Reference
Set reference52 = part2.CreateReferenceFromObject(point2D14)
Dim reference53 As Reference
Set reference53 = part2.CreateReferenceFromObject(line2D14)
Dim constraint25 As Constraint
Set constraint25 = constraints4.AddBiEltCst(catCstTypeDistance, reference52, reference53)
constraint25.Mode = catCstModeDrivingDimension
Dim length14 As Length
Set length14 = constraint25.Dimension
length14.Value = A - B / 2

Dim point2D16 As Point2D
Set point2D16 = factory2D4.CreatePoint(0, (D / 2 - d_ / 2) / 2 + d_ / 2 - B / 1.555 / 2)
point2D16.ReportName = 12
Dim point2D17 As Point2D
Set point2D17 = factory2D4.CreatePoint(B / 1.555 / 2, (D / 2 - d_ / 2) / 2 + d_ / 2)
point2D17.ReportName = 13
Dim circle2D4 As Circle2D
Set circle2D4 = factory2D4.CreateCircle(0, (D / 2 - d_ / 2) / 2 + d_ / 2, B / 1.555 / 2, 4.712389, 5.858957)
circle2D4.CenterPoint = point2D15
circle2D4.ReportName = 14
circle2D4.StartPoint = point2D16
circle2D4.EndPoint = point2D17

Dim reference54 As Reference
Set reference54 = part2.CreateReferenceFromObject(circle2D4)
Dim constraint26 As Constraint
Set constraint26 = constraints4.AddMonoEltCst(catCstTypeRadius, reference54)
constraint26.Mode = catCstModeDrivingDimension
Dim length15 As Length
Set length15 = constraint26.Dimension
length15.Value = B / 1.555 / 2

Dim reference55 As Reference
Set reference55 = part2.CreateReferenceFromObject(point2D15)
Dim reference56 As Reference
Set reference56 = part2.CreateReferenceFromObject(line2D13)
Dim constraint27 As Constraint
Set constraint27 = constraints4.AddBiEltCst(catCstTypeDistance, reference55, reference56)
constraint27.Mode = catCstModeDrivingDimension
Dim length16 As Length
Set length16 = constraint27.Dimension
length16.Value = (D / 2 - d_ / 2) / 2 + d_ / 2

Dim line2D18 As Line2D
Set line2D18 = factory2D4.CreateLine(B / 2, d_1 / 2 - d_ / 2, B / 1.555 / 2, d_1 / 2 - d_2)
line2D18.ReportName = 15
line2D18.StartPoint = point2D13
line2D18.EndPoint = point2D17

Dim reference57 As Reference
Set reference57 = part2.CreateReferenceFromObject(line2D18)
Dim reference58 As Reference
Set reference58 = part2.CreateReferenceFromObject(line2D13)
Dim constraint28 As Constraint
```

```
Set constraint28 = constraints4.AddBiEltCst(catCstTypeHorizontality, reference57, reference58)
constraint28.Mode = catCstModeDrivingDimension

Dim point2D18 As Point2D
Set point2D18 = factory2D4.CreatePoint(-B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2 - B / 1.555 / 2)
point2D18.ReportName = 16
Dim line2D19 As Line2D
Set line2D19 = factory2D4.CreateLine(0, (D / 2 - d_ / 2) / 2 + d_ / 2 - B / 1.555 / 2, -B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2 - B / 1.555 / 2)
line2D19.ReportName = 17
line2D19.StartPoint = point2D16
line2D19.EndPoint = point2D18

Dim reference59 As Reference
Set reference59 = part2.CreateReferenceFromObject(line2D19)
Dim reference60 As Reference
Set reference60 = part2.CreateReferenceFromObject(line2D13)
Dim constraint29 As Constraint
Set constraint29 = constraints4.AddBiEltCst(catCstTypeHorizontality, reference59, reference60)
constraint29.Mode = catCstModeDrivingDimension

Dim reference61 As Reference
Set reference61 = part2.CreateReferenceFromObject(line2D19)
Dim reference62 As Reference
Set reference62 = part2.CreateReferenceFromObject(circle2D4)
Dim constraint30 As Constraint
Set constraint30 = constraints4.AddBiEltCst(catCstTypeTangency, reference61, reference62)
constraint30.Mode = catCstModeDrivingDimension

Dim reference63 As Reference
Set reference63 = part2.CreateReferenceFromObject(line2D17)
Dim reference64 As Reference
Set reference64 = part2.CreateReferenceFromObject(line2D13)
Dim constraint31 As Constraint
Set constraint31 = constraints4.AddBiEltCst(catCstTypeAngle, reference63, reference64)
constraint31.Mode = catCstModeDrivingDimension
constraint31.AngleSector = catCstAngleSector1
Dim angle2 As angle
Set angle2 = constraint31.Dimension
angle2.Value = 50#

Dim line2D20 As Line2D
Set line2D20 = factory2D4.CreateLine(-B / 2, d_ / 2, -B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2 - B / 1.555 / 2)
line2D20.ReportName = 18
line2D20.StartPoint = point2D11
line2D20.EndPoint = point2D18

Dim reference65 As Reference
Set reference65 = part2.CreateReferenceFromObject(line2D20)
Dim reference66 As Reference
Set reference66 = part2.CreateReferenceFromObject(line2D14)
Dim constraint32 As Constraint
Set constraint32 = constraints4.AddBiEltCst(catCstTypeVerticality, reference65, reference66)
constraint32.Mode = catCstModeDrivingDimension

sketch4.CloseEdition
part2.InWorkObject = sketch4
part2.Update

Dim shapeFactory2 As ShapeFactory
Set shapeFactory2 = part2.ShapeFactory
Dim rib2 As Rib
Set rib2 = shapeFactory2.AddNewRibFromRef(Nothing, Nothing)
Dim reference67 As Reference
Set reference67 = part2.CreateReferenceFromObject(sketch4)
rib2.SetProfileElement reference67
Dim reference68 As Reference
Set reference68 = part2.CreateReferenceFromObject(sketch3)
rib2.CenterCurveElement = reference68
rib2.Name = "rib2"

part2.Update

Dim reference69 As Reference
Set reference69 = part2.CreateReferenceFromName("")
Dim constRadEdgeFillet3 As ConstRadEdgeFillet
Set constRadEdgeFillet3 = shapeFactory2.AddNewSolidEdgeFilletWithConstantRadius(reference69, catTangencyFilletEdgePropagation, r)
Dim reference70 As Reference
Set reference70 = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;18)));None:();Cf11:());Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;6)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", rib2)
constRadEdgeFillet3.AddObjectToFillet reference70
constRadEdgeFillet3.EdgePropagation = catTangencyFilletEdgePropagation
Dim reference71 As Reference
Set reference71 = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;6)));None:();Cf11:());Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;8)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", rib2)
constRadEdgeFillet3.AddObjectToFillet reference71
constRadEdgeFillet3.EdgePropagation = catTangencyFilletEdgePropagation

part2.Update

Dim product2 As Product
```

```
Set product2 = partDocument2.GetItem("Part2")
product2.PartNumber = "arointerno"

partDocument2.SaveAs objPath & "\aroint.CATPart"
partDocument2.Close

'====================================
'----------------------**bolas**---------
'====================================

Dim documents3 As Documents
Set documents3 = CATIA.Documents
Dim partDocument3 As PartDocument
Set partDocument3 = documents3.Add("Part")
Dim part3 As Part
Set part3 = partDocument3.Part
Dim bodies3 As Bodies
Set bodies3 = part3.Bodies
Dim body3 As Body
Set body3 = bodies3.Item("PartBody")
Dim sketches3 As Sketches
Set sketches3 = body3.Sketches
Dim originElements3 As OriginElements
Set originElements3 = part3.OriginElements
Dim reference72 As Reference
Set reference72 = originElements3.PlaneYZ
Dim sketch5 As Sketch
Set sketch5 = sketches3.Add(reference72)

Dim arrayOfVariantOfDouble5(8)
arrayOfVariantOfDouble5(0) = 0#
arrayOfVariantOfDouble5(1) = 0#
arrayOfVariantOfDouble5(2) = 0#
arrayOfVariantOfDouble5(3) = 0#
arrayOfVariantOfDouble5(4) = 1#
arrayOfVariantOfDouble5(5) = 0#
arrayOfVariantOfDouble5(6) = 0#
arrayOfVariantOfDouble5(7) = 0#
arrayOfVariantOfDouble5(8) = 1#
Set sketch5Variant = sketch5
sketch5Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble5
part3.InWorkObject = sketch5

Dim factory2D5 As Factory2D
Set factory2D5 = sketch5.OpenEdition()
Dim geometricElements5 As GeometricElements
Set geometricElements5 =
sketch5.GeometricElements
Dim axis2D5 As Axis2D
Set axis2D5 =
geometricElements5.Item("AbsoluteAxis")
Dim line2D21 As Line2D
Set line2D21 = axis2D5.GetItem("HDirection")
line2D21.ReportName = 1
Dim line2D22 As Line2D
Set line2D22 = axis2D5.GetItem("VDirection")
line2D22.ReportName = 2

Dim point2D19 As Point2D
Set point2D19 = factory2D5.CreatePoint(0#, (D / 2 -
d_ / 2) / 2 + d_ / 2 + B / 1.555 / 2)
point2D19.ReportName = 3
Dim point2D20 As Point2D
Set point2D20 = factory2D5.CreatePoint(0#, (D / 2 -
d_ / 2) / 2 + d_ / 2 - B / 1.555 / 2)
point2D20.ReportName = 4

Dim line2D23 As Line2D
Set line2D23 = factory2D5.CreateLine(0#, (D / 2 - d_
/ 2) / 2 + d_ / 2 + B / 1.555 / 2, 0#, (D / 2 - d_ / 2) / 2
+ d_ / 2 - B / 1.555 / 2)
line2D23.ReportName = 5
line2D23.StartPoint = point2D19
line2D23.EndPoint = point2D20

Dim constraints5 As Constraints
Set constraints5 = sketch5.Constraints
Dim reference73 As Reference
Set reference73 =
part3.CreateReferenceFromObject(point2D19)
Dim reference74 As Reference
Set reference74 =
part3.CreateReferenceFromObject(line2D22)
Dim constraint36 As Constraint
Set constraint36 =
constraints5.AddBiEltCst(catCstTypeOn,
reference73, reference74)
constraint36.Mode = catCstModeDrivingDimension

Dim reference75 As Reference
Set reference75 =
part3.CreateReferenceFromObject(line2D23)
Dim reference76 As Reference
Set reference76 =
part3.CreateReferenceFromObject(line2D22)
Dim constraint37 As Constraint
Set constraint37 =
constraints5.AddBiEltCst(catCstTypeVerticality,
reference75, reference76)
constraint37.Mode = catCstModeDrivingDimension

Dim reference77 As Reference
Set reference77 =
part3.CreateReferenceFromObject(line2D21)
Dim reference78 As Reference
Set reference78 =
part3.CreateReferenceFromObject(point2D20)
Dim constraint38 As Constraint
Set constraint38 =
constraints5.AddBiEltCst(catCstTypeDistance,
reference77, reference78)
constraint38.Mode = catCstModeDrivingDimension
Dim length17 As Length
Set length17 = constraint38.Dimension
length17.Value = (D / 2 - d_ / 2) / 2 + d_ / 2 - B /
1.555 / 2

Dim circle2D5 As Circle2D
Set circle2D5 = factory2D5.CreateCircle(0#, (D / 2 -
d_ / 2) / 2 + d_ / 2, B / 1.555 / 2, 4.712389, 7.853982)

Dim point2D21 As Point2D
Set point2D21 = factory2D5.CreatePoint(0#, (D / 2 -
d_ / 2) / 2 + d_ / 2)
circle2D5.CenterPoint = point2D21
circle2D5.ReportName = 6
circle2D5.StartPoint = point2D19
circle2D5.EndPoint = point2D20

sketch5.CloseEdition
part3.InWorkObject = sketch5
part3.Update

Dim shapeFactory3 As ShapeFactory
```

```
Set shapeFactory3 = part3.ShapeFactory

Dim shaft1 As Shaft
Set shaft1 = shapeFactory3.AddNewShaft(sketch5)

'selección del eje de revolución
Dim ejederev As Reference
Set ejederev = part3.CreateReferenceFromObject(line2D22)
shaft1.RevoluteAxis = ejederev

part3.Update
'Patrón circular
Dim referenceb As Reference
Set referenceb = part3.CreateReferenceFromName("")
Dim referenceb1 As Reference
Set referenceb1 = part3.CreateReferenceFromName("")
Dim circPatternb As CircPattern
Set circPatternb = shapeFactory3.AddNewCircPattern(Nothing, 1, 2, 20#, 45#, 1, 1, referenceb, referenceb1, True, 0#, True)
circPatternb.CircularPatternParameters = catInstancesandAngularSpacing
Dim angularRepartitionb As angularRepartition
Set angularRepartitionb = circPatternb.angularRepartition

'calculo del número de agujeros para ello definimos primero el número pi
Const PIb = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es d_ / 2 + (D / 2 - d_ / 2) / 2
Dim longitudb As Double
longitudb = 2 * PIb * (d_ / 2 + (D / 2 - d_ / 2) / 2)
'crearemos una variable auxiliar que usara el comando matematico Round para redondear el numero de huecos que tendra el separador.
'para calcular este número hacemos la hipótesis de que la distancia entre bolas sea del radio de bola por lo que multiplicamos la
'variable longitud por 3/2 veces el diámetro de una bola
Dim variableauxb As Double
variableauxb = Math.Round(longitudb / (3 / 2 * B / 1.555))
'ya tenemos el numero de huecos del separador
Dim intParamb As intParam
Set intParamb = angularRepartitionb.InstancesCount
intParamb.Value = variableauxb
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angleb As angle
Set angleb = angularRepartitionb.AngularSpacing
angleb.Value = 360 / variableauxb
circPatternb.SetRotationAxis reference72
part3.Update

Dim product3 As Product
Set product3 = partDocument3.GetItem("Part3")
product3.PartNumber = "bolas"

partDocument3.SaveAs objPath & "\bolas.CATPart"
partDocument3.Close

'======================================
'-------------------------jaula-------------------------------
'======================================
Dim documents4 As Documents
Set documents4 = CATIA.Documents
Dim partDocument4 As PartDocument
Set partDocument4 = documents4.Add("Part")
Dim part4 As Part
Set part4 = partDocument4.Part
Dim bodies4 As Bodies
Set bodies4 = part4.Bodies
Dim body4 As Body
Set body4 = bodies4.Item("PartBody")
Dim sketches4 As Sketches
Set sketches4 = body4.Sketches
Dim originElements4 As OriginElements
Set originElements4 = part4.OriginElements
Dim reference79 As Reference
Set reference79 = originElements4.PlaneZX
Dim sketch6 As Sketch
Set sketch6 = sketches4.Add(reference79)

Dim arrayOfVariantOfDouble6(8)
arrayOfVariantOfDouble6(0) = 0#
arrayOfVariantOfDouble6(1) = 0#
arrayOfVariantOfDouble6(2) = 0#
arrayOfVariantOfDouble6(3) = -1#
arrayOfVariantOfDouble6(4) = 0#
arrayOfVariantOfDouble6(5) = 0#
arrayOfVariantOfDouble6(6) = 0#
arrayOfVariantOfDouble6(7) = -0#
arrayOfVariantOfDouble6(8) = 1#
Set sketch6Variant = sketch6
sketch6Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble6

part4.InWorkObject = sketch6

Dim factory2D6 As Factory2D
Set factory2D6 = sketch6.OpenEdition()
Dim geometricElements6 As GeometricElements
Set geometricElements6 = sketch6.GeometricElements
Dim axis2D6 As Axis2D
Set axis2D6 = geometricElements6.Item("AbsoluteAxis")
Dim line2D24 As Line2D
Set line2D24 = axis2D6.GetItem("HDirection")
line2D24.ReportName = 1
Dim line2D25 As Line2D
Set line2D25 = axis2D6.GetItem("VDirection")
line2D25.ReportName = 2

Dim point2D22 As Point2D
Set point2D22 = factory2D6.CreatePoint(A - B / 2, 0#)
point2D22.ReportName = 3
Dim line2D26 As Line2D
Set line2D26 = factory2D6.CreateLine(0#, 0#, A - B / 2, 0#)
line2D26.ReportName = 4
line2D26.Construction = True
Dim point2D23 As Point2D
Set point2D23 = axis2D6.GetItem("Origin")
line2D26.StartPoint = point2D23
line2D26.EndPoint = point2D22
```

```
Dim constraints6 As Constraints
Set constraints6 = sketch6.Constraints

Dim reference81 As Reference
Set reference81 = part4.CreateReferenceFromObject(line2D26)
Dim reference82 As Reference
Set reference82 = part4.CreateReferenceFromObject(line2D24)
Dim constraint39 As Constraint
Set constraint39 = constraints6.AddBiEltCst(catCstTypeHorizontality, reference81, reference82)
constraint39.Mode = catCstModeDrivingDimension

Dim reference83 As Reference
Set reference83 = part4.CreateReferenceFromObject(line2D26)
Dim constraint40 As Constraint
Set constraint40 = constraints6.AddMonoEltCst(catCstTypeLength, reference83)
constraint40.Mode = catCstModeDrivingDimension
Dim length18 As Length
Set length18 = constraint40.Dimension
length18.Value = A - B / 2

Dim point2D24 As Point2D
Set point2D24 = factory2D6.CreatePoint(0, (D / 2 - d_ / 2) / 2 + d_ / 2)
point2D24.ReportName = 5
Dim line2D27 As Line2D
Set line2D27 = factory2D6.CreateLine(A - B / 2, 0#, 0, (D / 2 - d_ / 2) / 2 + d_ / 2)
line2D27.ReportName = 6
line2D27.Construction = True
line2D27.StartPoint = point2D22
line2D27.EndPoint = point2D24

Dim reference84 As Reference
Set reference84 = part4.CreateReferenceFromObject(line2D27)
Dim reference85 As Reference
Set reference85 = part4.CreateReferenceFromObject(line2D24)
Dim constraint41 As Constraint
Set constraint41 = constraints6.AddBiEltCst(catCstTypeAngle, reference84, reference85)
constraint41.Mode = catCstModeDrivingDimension
constraint41.AngleSector = catCstAngleSector1
Dim angle3 As angle
Set angle3 = constraint41.Dimension
angle3.Value = 50#

Dim reference86 As Reference
Set reference86 = part4.CreateReferenceFromObject(point2D24)
Dim reference87 As Reference
Set reference87 = part4.CreateReferenceFromObject(line2D24)
Dim constraint42 As Constraint
Set constraint42 = constraints6.AddBiEltCst(catCstTypeDistance, reference86, reference87)
constraint42.Mode = catCstModeDrivingDimension
Dim length19 As Length
Set length19 = constraint42.Dimension
length19.Value = (D / 2 - d_ / 2) / 2 + d_ / 2

Dim point2D25 As Point2D
Set point2D25 = factory2D6.CreatePoint(B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2 + (d_1 / 2 - d_ / 2) / 3)
point2D25.ReportName = 7
Dim point2D26 As Point2D
Set point2D26 = factory2D6.CreatePoint(-B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2)
point2D26.ReportName = 8
Dim line2D28 As Line2D
Set line2D28 = factory2D6.CreateLine(B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2 + (d_1 / 2 - d_ / 2) / 3, -B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2)
line2D28.ReportName = 9
line2D28.StartPoint = point2D25
line2D28.EndPoint = point2D26

Dim reference88 As Reference
Set reference88 = part4.CreateReferenceFromObject(point2D25)
Dim reference89 As Reference
Set reference89 = part4.CreateReferenceFromObject(line2D25)
Dim constraint43 As Constraint
Set constraint43 = constraints6.AddBiEltCst(catCstTypeDistance, reference88, reference89)
constraint43.Mode = catCstModeDrivingDimension
Dim length20 As Length
Set length20 = constraint43.Dimension
length20.Value = B / 2

Dim reference90 As Reference
Set reference90 = part4.CreateReferenceFromObject(point2D25)
Dim reference91 As Reference
Set reference91 = part4.CreateReferenceFromObject(line2D24)
Dim constraint44 As Constraint
Set constraint44 = constraints6.AddBiEltCst(catCstTypeDistance, reference90, reference91)
constraint44.Mode = catCstModeDrivingDimension
Dim length21 As Length
Set length21 = constraint44.Dimension
length21.Value = (D / 2 - d_ / 2) / 2 + d_ / 2 + (d_1 / 2 - d_ / 2) / 3

Dim reference92 As Reference
Set reference92 = part4.CreateReferenceFromObject(point2D24)
Dim reference93 As Reference
Set reference93 = part4.CreateReferenceFromObject(line2D28)
Dim constraint45 As Constraint
Set constraint45 = constraints6.AddBiEltCst(catCstTypeDistance, reference92, reference93)
constraint45.Mode = catCstModeDrivingDimension
Dim length22 As Length
Set length22 = constraint45.Dimension
length22.Value = (d_1 / 2 - d_ / 2) / 10

Dim reference94 As Reference
```

```
Set reference94 = part4.CreateReferenceFromObject(point2D26)
Dim reference95 As Reference
Set reference95 = part4.CreateReferenceFromObject(line2D25)
Dim constraint46 As Constraint
Set constraint46 = constraints6.AddBiEltCst(catCstTypeDistance, reference94, reference95)
constraint46.Mode = catCstModeDrivingDimension
Dim length23 As Length
Set length23 = constraint46.Dimension
length23.Value = B / 2

Dim point2D27 As Point2D
Set point2D27 = factory2D6.CreatePoint(B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2)
point2D27.ReportName = 10
Dim line2D29 As Line2D
Set line2D29 = factory2D6.CreateLine(B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2 + (d_1 / 2 - d_ / 2) / 3, B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2)
line2D29.ReportName = 11
line2D29.StartPoint = point2D25
line2D29.EndPoint = point2D27

Dim reference96 As Reference
Set reference96 = part4.CreateReferenceFromObject(line2D29)
Dim reference97 As Reference
Set reference97 = part4.CreateReferenceFromObject(line2D25)
Dim constraint47 As Constraint
Set constraint47 = constraints6.AddBiEltCst(catCstTypeVerticality, reference96, reference97)
constraint47.Mode = catCstModeDrivingDimension

Dim reference98 As Reference
Set reference98 = part4.CreateReferenceFromObject(line2D29)
Dim constraint48 As Constraint
Set constraint48 = constraints6.AddMonoEltCst(catCstTypeLength, reference98)
constraint48.Mode = catCstModeDrivingDimension
Dim length24 As Length
Set length24 = constraint48.Dimension
length24.Value = (d_1 / 2 - d_ / 2) / 5

Dim point2D28 As Point2D
Set point2D28 = factory2D6.CreatePoint(-B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2)
point2D28.ReportName = 12
Dim line2D30 As Line2D
Set line2D30 = factory2D6.CreateLine(B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2, -B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2)
line2D30.ReportName = 13
line2D30.StartPoint = point2D27
line2D30.EndPoint = point2D28

Dim reference99 As Reference
Set reference99 = part4.CreateReferenceFromObject(line2D30)
Dim reference100 As Reference
Set reference100 = part4.CreateReferenceFromObject(line2D28)
Dim constraint49 As Constraint
Set constraint49 = constraints6.AddBiEltCst(catCstTypeParallelism, reference99, reference100)
constraint49.Mode = catCstModeDrivingDimension

Dim line2D31 As Line2D
Set line2D31 = factory2D6.CreateLine(-B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2, -B / 2, (D / 2 - d_ / 2) / 2 + d_ / 2 + 1)
line2D31.ReportName = 14
line2D31.StartPoint = point2D26
line2D31.EndPoint = point2D28
Dim reference101 As Reference
Set reference101 = part4.CreateReferenceFromObject(line2D31)
Dim reference102 As Reference
Set reference102 = part4.CreateReferenceFromObject(line2D25)
Dim constraint50 As Constraint
Set constraint50 = constraints6.AddBiEltCst(catCstTypeVerticality, reference101, reference102)
constraint50.Mode = catCstModeDrivingDimension

sketch6.CloseEdition
part4.InWorkObject = sketch6
part4.Update

Dim reference103 As Reference
Set reference103 = originElements4.PlaneYZ
Dim sketch7 As Sketch
Set sketch7 = sketches4.Add(reference103)
Dim arrayOfVariantOfDouble7(8)
arrayOfVariantOfDouble7(0) = 0#
arrayOfVariantOfDouble7(1) = 0#
arrayOfVariantOfDouble7(2) = 0#
arrayOfVariantOfDouble7(3) = 0#
arrayOfVariantOfDouble7(4) = 1#
arrayOfVariantOfDouble7(5) = 0#
arrayOfVariantOfDouble7(6) = 0#
arrayOfVariantOfDouble7(7) = 0#
arrayOfVariantOfDouble7(8) = 1#
Set sketch7Variant = sketch7
sketch7Variant.SetAbsoluteAxisData arrayOfVariantOfDouble7

part4.InWorkObject = sketch7

Dim factory2D7 As Factory2D
Set factory2D7 = sketch7.OpenEdition()
Dim geometricElements7 As GeometricElements
Set geometricElements7 = sketch7.GeometricElements
Dim axis2D7 As Axis2D
Set axis2D7 = geometricElements7.Item("AbsoluteAxis")
Dim line2D32 As Line2D
Set line2D32 = axis2D7.GetItem("HDirection")
line2D32.ReportName = 1
Dim line2D33 As Line2D
Set line2D33 = axis2D7.GetItem("VDirection")
line2D33.ReportName = 2

Dim circle2D6 As Circle2D
Set circle2D6 = factory2D7.CreateClosedCircle(0#, 0#, (D / 2 - d_ / 2) / 2 + d_ / 2)
```

```
Dim point2D29 As Point2D
Set point2D29 = axis2D7.GetItem("Origin")
circle2D6.CenterPoint = point2D29
circle2D6.ReportName = 3

Dim constraints7 As Constraints
Set constraints7 = sketch7.Constraints
Dim reference104 As Reference
Set reference104 = part4.CreateReferenceFromObject(circle2D6)
Dim constraint51 As Constraint
Set constraint51 = constraints7.AddMonoEltCst(catCstTypeRadius, reference104)
constraint51.Mode = catCstModeDrivingDimension
Dim length25 As Length
Set length25 = constraint51.Dimension
length25.Value = (D / 2 - d_ / 2) / 2 + d_ / 2

sketch7.CloseEdition
part4.InWorkObject = sketch7
part4.Update

Dim shapeFactory4 As ShapeFactory
Set shapeFactory4 = part4.ShapeFactory
Dim rib3 As Rib
Set rib3 = shapeFactory4.AddNewRibFromRef(Nothing, Nothing)
Dim reference105 As Reference
Set reference105 = part4.CreateReferenceFromObject(sketch6)
rib3.SetProfileElement reference105
Dim reference106 As Reference
Set reference106 = part4.CreateReferenceFromObject(sketch7)
rib3.CenterCurveElement = reference106

part4.Update

Dim reference107 As Reference
Set reference107 = originElements4.PlaneXY
Dim sketch8 As Sketch
Set sketch8 = sketches4.Add(reference107)
Dim arrayOfVariantOfDouble8(8)
arrayOfVariantOfDouble8(0) = 0#
arrayOfVariantOfDouble8(1) = 0#
arrayOfVariantOfDouble8(2) = 0#
arrayOfVariantOfDouble8(3) = 1#
arrayOfVariantOfDouble8(4) = 0#
arrayOfVariantOfDouble8(5) = 0#
arrayOfVariantOfDouble8(6) = 0#
arrayOfVariantOfDouble8(7) = 1#
arrayOfVariantOfDouble8(8) = 0#
Set sketch8Variant = sketch8
sketch8Variant.SetAbsoluteAxisData arrayOfVariantOfDouble8
part4.InWorkObject = sketch8

Dim factory2D8 As Factory2D
Set factory2D8 = sketch8.OpenEdition()
Dim geometricElements8 As GeometricElements
Set geometricElements8 = sketch8.GeometricElements
Dim axis2D8 As Axis2D
Set axis2D8 = geometricElements8.Item("AbsoluteAxis")
Dim line2D34 As Line2D
Set line2D34 = axis2D8.GetItem("HDirection")
line2D34.ReportName = 1
Dim line2D35 As Line2D
Set line2D35 = axis2D8.GetItem("VDirection")
line2D35.ReportName = 2

Dim circle2D7 As Circle2D
Set circle2D7 = factory2D8.CreateClosedCircle(0#, 0#, B / 1.555 / 2)
Dim point2D33 As Point2D
Set point2D33 = axis2D8.GetItem("Origin")
circle2D7.CenterPoint = point2D33
circle2D7.ReportName = 3

Dim constraints8 As Constraints
Set constraints8 = sketch8.Constraints
Dim reference108 As Reference
Set reference108 = part4.CreateReferenceFromObject(circle2D7)
Dim constraint52 As Constraint
Set constraint52 = constraints8.AddMonoEltCst(catCstTypeRadius, reference108)
constraint52.Mode = catCstModeDrivingDimension
Dim length26 As Length
Set length26 = constraint52.Dimension
length26.Value = B / 1.555 / 2

sketch8.CloseEdition
part4.InWorkObject = sketch8
part4.Update

Dim pocket1 As Pocket
Set pocket1 = shapeFactory4.AddNewPocket(sketch8, 20#)
pocket1.DirectionOrientation = catRegularOrientation
Dim limit1 As Limit
Set limit1 = pocket1.FirstLimit
limit1.LimitMode = catUpToLastLimit

part4.Update

Dim reference109 As Reference
Set reference109 = part4.CreateReferenceFromName("")
Dim reference110 As Reference
Set reference110 = part4.CreateReferenceFromName("")
Dim circPattern1 As CircPattern
Set circPattern1 = shapeFactory4.AddNewCircPattern(pocket1, 1, 2, 20#, 45#, 1, 1, reference109, reference110, True, 0#, True)
circPattern1.CircularPatternParameters = catInstancesandAngularSpacing

Dim angularRepartition1 As angularRepartition
Set angularRepartition1 = circPattern1.angularRepartition

'calculo del número de agujeros para ello definimos primero el número pi
Const PI = 3.14159265358979
```

```
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es d_ / 2 + (D / 2 - d_ / 2) / 2
Dim longitud As Double
longitud = 2 * PI * (d_ / 2 + (D / 2 - d_ / 2) / 2)
'crearemos una variable auxiliar que usara el comando matematico Round para redondear el numero de huecos que tendra el separador.
'para calcular este número hacemos la hipótesis de que la distancia entre bolas sea del radio de bola por lo que multiplicamos la
'variable longitud por 3/2 veces el diámetro de una bola
Dim variableaux As Double
variableaux = Math.Round(longitud / (3 / 2 * B / 1.555))
'ya tenemos el numero de huecos del separador
Dim intParam As intParam
Set intParam = angularRepartition1.InstancesCount
intParam.Value = variableaux
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angle As angle
Set angle = angularRepartition1.AngularSpacing
angle.Value = 360 / variableaux

Dim reference111 As Reference
Set reference111 = part4.CreateReferenceFromObject(line2D34)
circPattern1.SetRotationAxis reference111

part4.Update
Dim product4 As Product
Set product4 = partDocument4.GetItem("Part4")
product4.PartNumber = "jaula"

partDocument4.SaveAs objPath & "\jaula.CATPart"
partDocument4.Close

'=====================================
'----------------------------ENSAMBLAJE
'=====================================
Dim documents5 As Documents
Set documents5 = CATIA.Documents

Dim productDocument1 As ProductDocument
Set productDocument1 = documents5.Add("Product")

Dim product5 As Product
Set product5 = productDocument1.Product

Dim products1 As Products
Set products1 = product5.Products


Dim arrayOfVariantOfBSTR1(0)
arrayOfVariantOfBSTR1(0) = objPath & "\aroext.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR1, "All"
Set productDocument1 = CATIA.ActiveDocument

Dim constraints9 As Constraints
Set constraints9 = product5.Connections("CATIAConstraints")

'Fix del aro externo
Dim reference112 As Reference
Set reference112 = product5.CreateReferenceFromName("Product5/aroexterno.1/!Product5/aroexterno.1/")
Dim constraint53 As Constraint
Set constraint53 = constraints9.AddMonoEltCst(catCstTypeReference, reference112)

Dim arrayOfVariantOfBSTR2(0)
arrayOfVariantOfBSTR2(0) = objPath & "\aroint.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR2, "All"

'coincidencia de ejes de los aros externo e interno
Dim reference113 As Reference
Set reference113 = product5.CreateReferenceFromName("Product5/arointerno.1/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;6)));None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055))")
Dim reference114 As Reference
Set reference114 = product5.CreateReferenceFromName("Product5/aroexterno.1/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;7)));None:();Cf11:());EdgeFillet.2_ResultOUT;Z0;G3055))")
Dim constraint54 As Constraint
Set constraint54 = constraints9.AddBiEltCst(catCstTypeOn, reference113, reference114)

Dim arrayOfVariantOfBSTR3(0)
arrayOfVariantOfBSTR3(0) = objPath & "\bolas.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR3, "All"

Dim arrayOfVariantOfBSTR4(0)
arrayOfVariantOfBSTR4(0) = objPath & "\jaula.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR4, "All"

Dim reference115 As Reference
Set reference115 = product5.CreateReferenceFromName("Product5/jaula.1/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.2;3);Brp:(Sketch.1;9)));None:();Cf11:());CircPattern.1_ResultOUT;Z0;G3055))")
Dim constraint55 As Constraint
Set constraint55 = constraints9.AddBiEltCst(catCstTypeOn, reference114, reference115)

'fijamos la distancia entre la cara lateral del aro externo y el interno
Dim reference116 As Reference
Set reference116 = product5.CreateReferenceFromName("Product5/aroexterno.1/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(
```

```
Sketch.1;3);Brp:(Sketch.2;9)));None:();Cf11:());EdgeFillet.2_ResultOUT;Z0;G3055)")
Dim reference117 As Reference
Set reference117 = product5.CreateReferenceFromName("Product5/arointerno.1/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;18)));None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055)")
Dim constraint56 As Constraint
Set constraint56 = constraints9.AddBiEltCst(catCstTypeDistance, reference116, reference117)
Dim length27 As Length
Set length27 = constraint56.Dimension
length27.Value = 0#
constraint56.Orientation = catCstOrientSame

'idem para el aro externo y la jaula
Dim reference118 As Reference
Set reference118 = product5.CreateReferenceFromName("Product5/jaula.1/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.2;3);Brp:(Sketch.1;14)));None:();Cf11:());CircPattern.1_ResultOUT;Z0;G3055)")
Dim constraint57 As Constraint
Set constraint57 = constraints9.AddBiEltCst(catCstTypeDistance, reference116, reference118)
Dim length28 As Length
Set length28 = constraint57.Dimension
length28.Value = 0#
constraint57.Orientation = catCstOrientSame
product5.Update

'Fijamos las bolas a los huecos de la jaula.
Dim referencec1 As Reference
Set referencec1 = product5.CreateReferenceFromName("Product5/bolas.1/!zx plane")
Dim referencec2 As Reference
Set referencec2 = product5.CreateReferenceFromName("Product5/jaula.1/!zx plane")
Dim constraintc1 As Constraint
Set constraintc1 = constraints9.AddBiEltCst(catCstTypeDistance, referencec1, referencec2)

Dim referencec3 As Reference
Set referencec3 = product5.CreateReferenceFromName("Product5/bolas.1/!yz plane")
Dim referencec4 As Reference
Set referencec4 = product5.CreateReferenceFromName("Product5/jaula.1/!yz plane")
Dim constraintc2 As Constraint
Set constraintc2 = constraints9.AddBiEltCst(catCstTypeDistance, referencec3, referencec4)

Dim referencec5 As Reference
Set referencec5 = product5.CreateReferenceFromName("Product5/bolas.1/!xy plane")
Dim referencec6 As Reference
Set referencec6 = product5.CreateReferenceFromName("Product5/jaula.1/!xy plane")
Dim constraintc3 As Constraint
Set constraintc3 = constraints9.AddBiEltCst(catCstTypeDistance, referencec5, referencec6)

product5.Update
productDocument1.SaveAs objPath & "\Angular.CATProduct"
End Sub
```

- **Ensamblaje en Tándem de rodamientos de bolas con contacto angular.**

```
Dim arrayOfVariantOfBSTR5(0)
arrayOfVariantOfBSTR5(0) = objPath & "\aroext.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR5, "All"
Set productDocument1 = CATIA.ActiveDocument

Dim constraints10 As Constraints
Set constraints10 = product5.Connections("CATIAConstraints")

Dim arrayOfVariantOfBSTR6(0)
arrayOfVariantOfBSTR6(0) = objPath & "\aroint.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR6, "All"

'coincidencia de ejes de los aros externo e interno
Dim reference119 As Reference
Set reference119 = product5.CreateReferenceFromName("Product5/aroiinterno.2/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;6)));None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055))")
Dim reference120 As Reference
Set reference120 = product5.CreateReferenceFromName("Product5/aroexterno.2/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;7)));None:();Cf11:());EdgeFillet.2_ResultOUT;Z0;G3055))")
Dim constraint58 As Constraint
Set constraint58 = constraints10.AddBiEltCst(catCstTypeOn, reference119, reference120)

Dim arrayOfVariantOfBSTR7(0)
arrayOfVariantOfBSTR7(0) = objPath & "\bolas.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR7, "All"

Dim arrayOfVariantOfBSTR8(0)
arrayOfVariantOfBSTR8(0) = objPath & "\jaula.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR8, "All"

Dim reference121 As Reference
```

```
Set reference121 = product5.CreateReferenceFromName("Product5/jaula.2/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.2;3);Brp:(Sketch.1;9)));None:();Cf11:());CircPattern.1_ResultOUT;Z0;G3055))")
Dim constraint59 As Constraint
Set constraint59 = constraints10.AddBiEltCst(catCstTypeOn, reference120, reference121)

'fijamos la distancia entre la cara lateral del aro externo y el interno
Dim reference122 As Reference
Set reference122 = product5.CreateReferenceFromName("Product5/aroexterno.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;9)));None:();Cf11:());EdgeFillet.2_ResultOUT;Z0;G3055)")
Dim reference123 As Reference
Set reference123 = product5.CreateReferenceFromName("Product5/aroiinterno.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;18)));None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055)")
Dim constraint60 As Constraint
Set constraint60 = constraints10.AddBiEltCst(catCstTypeDistance, reference122, reference123)
Dim length29 As Length
Set length29 = constraint60.Dimension
length29.Value = 0#
constraint60.Orientation = catCstOrientSame

'idem para el aro externo y la jaula
Dim reference124 As Reference
Set reference124 = product5.CreateReferenceFromName("Product5/jaula.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.2;3);Brp:(Sketch.1;14)));None:();Cf11:());CircPattern.1_ResultOUT;Z0;G3055)")
Dim constraint61 As Constraint
Set constraint61 = constraints10.AddBiEltCst(catCstTypeDistance, reference122, reference124)
Dim length30 As Length
Set length30 = constraint61.Dimension
length30.Value = 0#
constraint61.Orientation = catCstOrientSame

'contacto de las caras de los rodamientos completos
Dim reference125 As Reference
Set reference125 = product5.CreateReferenceFromName("Product5/aroexterno.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;9)));None:();Cf11:());EdgeFillet.2_ResultOUT;Z0;G3055)")
Dim reference126 As Reference
Set reference126 = product5.CreateReferenceFromName("Product5/aroexterno.1/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;19)));None:();Cf11:());EdgeFillet.2_ResultOUT;Z0;G3055)")
Dim constraint62 As Constraint
Set constraint62 = constraints10.AddBiEltCst(catCstTypeSurfContact, reference125, reference126)

'Fijamos las bolas a los huecos de la jaula.
Dim referencec1 As Reference
Set referencec1 = product5.CreateReferenceFromName("Product5/bolas.1/!zx plane")
Dim referencec2 As Reference
Set referencec2 = product5.CreateReferenceFromName("Product5/jaula.1/!zx plane")
Dim constraintc1 As Constraint
Set constraintc1 = constraints10.AddBiEltCst(catCstTypeDistance, referencec1, referencec2)

Dim referencec3 As Reference
Set referencec3 = product5.CreateReferenceFromName("Product5/bolas.1/!yz plane")
Dim referencec4 As Reference
Set referencec4 = product5.CreateReferenceFromName("Product5/jaula.1/!yz plane")
Dim constraintc2 As Constraint
Set constraintc2 = constraints10.AddBiEltCst(catCstTypeDistance, referencec3, referencec4)

Dim referencec5 As Reference
Set referencec5 = product5.CreateReferenceFromName("Product5/bolas.1/!xy plane")
Dim referencec6 As Reference
Set referencec6 = product5.CreateReferenceFromName("Product5/jaula.1/!xy plane")
Dim constraintc3 As Constraint
Set constraintc3 = constraints10.AddBiEltCst(catCstTypeDistance, referencec5, referencec6)

product5.Update

'Fijamos las bolas a los huecos de la jaula.
Dim referenced1 As Reference
Set referenced1 = product5.CreateReferenceFromName("Product5/bolas.2/!zx plane")
Dim referenced2 As Reference
Set referenced2 = product5.CreateReferenceFromName("Product5/jaula.2/!zx plane")
Dim constraintd1 As Constraint
Set constraintd1 = constraints10.AddBiEltCst(catCstTypeDistance, referenced1, referenced2)

Dim referenced3 As Reference
Set referenced3 = product5.CreateReferenceFromName("Product5/bolas.2/!yz plane")
Dim referenced4 As Reference
Set referenced4 = product5.CreateReferenceFromName("Product5/jaula.2/!yz plane")
Dim constraintd2 As Constraint
```

```
Set constraintd2 = constraints10.AddBiEltCst(catCstTypeDistance, referenced3, referenced4)

Dim referenced5 As Reference
Set referenced5 = product5.CreateReferenceFromName("Product5/bolas.2/!xy plane")
Dim referenced6 As Reference
Set referenced6 = product5.CreateReferenceFromName("Product5/jaula.2/!xy plane")
Dim constraintd3 As Constraint
Set constraintd3 = constraints10.AddBiEltCst(catCstTypeDistance, referenced5, referenced6)

product5.Update
productDocument1.SaveAs objPath & "\AngularEnTandem.CATProduct"
'productDocument1.Close
```

- **Ensamblaje en "X" de rodamientos de bolas con contacto angular.**

```
Dim arrayOfVariantOfBSTR5(0)
arrayOfVariantOfBSTR5(0) = objPath & "\aroext.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR5, "All"
Set productDocument1 = CATIA.ActiveDocument

Dim constraints10 As Constraints
Set constraints10 = product5.Connections("CATIAConstraints")

Dim arrayOfVariantOfBSTR6(0)
arrayOfVariantOfBSTR6(0) = objPath & "\aroint.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR6, "All"

'coincidencia de ejes de los aros externo e interno
Dim reference119 As Reference
Set reference119 = product5.CreateReferenceFromName("Product5/aroi
nterno.2/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(
Brp:(Sketch.1;3);Brp:(Sketch.2;6)));None:();Cf11:());
EdgeFillet.1_ResultOUT;Z0;G3055))")
Dim reference120 As Reference
Set reference120 = product5.CreateReferenceFromName("Product5/aroe
xterno.2/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(
Brp:(Sketch.1;3);Brp:(Sketch.2;7)));None:();Cf11:());
EdgeFillet.2_ResultOUT;Z0;G3055))")
Dim constraint58 As Constraint
Set constraint58 = constraints10.AddBiEltCst(catCstTypeOn, reference119, reference120)

Dim arrayOfVariantOfBSTR7(0)
arrayOfVariantOfBSTR7(0) = objPath & "\bolas.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR7, "All"

Dim arrayOfVariantOfBSTR8(0)
arrayOfVariantOfBSTR8(0) = objPath & "\jaula.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR8, "All"

Dim reference121 As Reference
Set reference121 = product5.CreateReferenceFromName("Product5/jaula
.2/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(S
ketch.2;3);Brp:(Sketch.1;9)));None:();Cf11:());CircPa
ttern.1_ResultOUT;Z0;G3055))")
Dim constraint59 As Constraint
Set constraint59 = constraints10.AddBiEltCst(catCstTypeOn, reference120, reference121)

'fijamos la distancia entre la cara lateral del aro externo y el interno
Dim reference122 As Reference
Set reference122 = product5.CreateReferenceFromName("Product5/aroe
xterno.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(
Sketch.1;3);Brp:(Sketch.2;9)));None:();Cf11:());Edge
Fillet.2_ResultOUT;Z0;G3055)")
Dim reference123 As Reference
Set reference123 = product5.CreateReferenceFromName("Product5/aroi
nterno.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(
Sketch.1;3);Brp:(Sketch.2;18)));None:();Cf11:());Edg
eFillet.1_ResultOUT;Z0;G3055)")
Dim constraint60 As Constraint
Set constraint60 = constraints10.AddBiEltCst(catCstTypeDistance, reference122, reference123)
Dim length29 As Length
Set length29 = constraint60.Dimension
length29.Value = 0#
constraint60.Orientation = catCstOrientSame

'idem para el aro externo y la jaula
Dim reference124 As Reference
Set reference124 = product5.CreateReferenceFromName("Product5/jaula
.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.
2;3);Brp:(Sketch.1;14)));None:();Cf11:());CircPattern
.1_ResultOUT;Z0;G3055)")
Dim constraint61 As Constraint
Set constraint61 = constraints10.AddBiEltCst(catCstTypeDistance, reference122, reference124)
Dim length30 As Length
Set length30 = constraint61.Dimension
length30.Value = 0#
constraint61.Orientation = catCstOrientSame

'contacto de las caras de los rodamientos completos
Dim reference125 As Reference
Set reference125 = product5.CreateReferenceFromName("Product5/aroe
xterno.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(
Sketch.1;3);Brp:(Sketch.2;19)));None:();Cf11:());Edg
eFillet.2_ResultOUT;Z0;G3055)")
Dim reference126 As Reference
Set reference126 = product5.CreateReferenceFromName("Product5/aroe
```

```
xterno.1/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(
Sketch.1;3);Brp:(Sketch.2;19)));None:();Cf11:());Edg
eFillet.2_ResultOUT;Z0;G3055)")
Dim constraint62 As Constraint
Set constraint62 =
constraints10.AddBiEltCst(catCstTypeSurfContact,
reference125, reference126)

'Fijamos las bolas a los huecos de la jaula.
Dim referencec1 As Reference
Set referencec1 =
product5.CreateReferenceFromName("Product5/bola
s.1/!zx plane")
Dim referencec2 As Reference
Set referencec2 =
product5.CreateReferenceFromName("Product5/jaula
.1/!zx plane")
Dim constraintc1 As Constraint
Set constraintc1 =
constraints10.AddBiEltCst(catCstTypeDistance,
referencec1, referencec2)

Dim referencec3 As Reference
Set referencec3 =
product5.CreateReferenceFromName("Product5/bola
s.1/!yz plane")
Dim referencec4 As Reference
Set referencec4 =
product5.CreateReferenceFromName("Product5/jaula
.1/!yz plane")
Dim constraintc2 As Constraint
Set constraintc2 =
constraints10.AddBiEltCst(catCstTypeDistance,
referencec3, referencec4)

Dim referencec5 As Reference
Set referencec5 =
product5.CreateReferenceFromName("Product5/bola
s.1/!xy plane")
Dim referencec6 As Reference
Set referencec6 =
product5.CreateReferenceFromName("Product5/jaula
.1/!xy plane")
Dim constraintc3 As Constraint
Set constraintc3 =
constraints10.AddBiEltCst(catCstTypeDistance,
referencec5, referencec6)

product5.Update

'Fijamos las bolas a los huecos de la jaula.
Dim referenced1 As Reference
Set referenced1 =
product5.CreateReferenceFromName("Product5/bola
s.2/!zx plane")
Dim referenced2 As Reference
Set referenced2 =
product5.CreateReferenceFromName("Product5/jaula
.2/!zx plane")
Dim constraintd1 As Constraint
Set constraintd1 =
constraints10.AddBiEltCst(catCstTypeDistance,
referenced1, referenced2)

Dim referenced3 As Reference
Set referenced3 =
product5.CreateReferenceFromName("Product5/bola
s.2/!yz plane")
Dim referenced4 As Reference
Set referenced4 =
product5.CreateReferenceFromName("Product5/jaula
.2/!yz plane")
Dim constraintd2 As Constraint
Set constraintd2 =
constraints10.AddBiEltCst(catCstTypeDistance,
referenced3, referenced4)

Dim referenced5 As Reference
Set referenced5 =
product5.CreateReferenceFromName("Product5/bola
s.2/!xy plane")
Dim referenced6 As Reference
Set referenced6 =
product5.CreateReferenceFromName("Product5/jaula
.2/!xy plane")
Dim constraintd3 As Constraint
Set constraintd3 =
constraints10.AddBiEltCst(catCstTypeDistance,
referenced5, referenced6)

product5.Update
productDocument1.SaveAs objPath &
"\AngularEnX.CATProduct"
```

- **Ensamblaje en "O" de rodamientos de bolas con contacto angular.**

```
Dim arrayOfVariantOfBSTR5(0)
arrayOfVariantOfBSTR5(0) = objPath &
"\aroext.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR5, "All"
Set productDocument1 = CATIA.ActiveDocument

Dim constraints10 As Constraints
Set constraints10 =
product5.Connections("CATIAConstraints")

Dim arrayOfVariantOfBSTR6(0)
arrayOfVariantOfBSTR6(0) = objPath &
"\aroint.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR6, "All"

'coincidencia de ejes de los aros externo e interno
Dim reference119 As Reference
Set reference119 =
product5.CreateReferenceFromName("Product5/aroi
nterno.2/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(
Brp:(Sketch.1;3);Brp:(Sketch.2;6)));None:();Cf11:());
EdgeFillet.1_ResultOUT;Z0;G3055))")
Dim reference120 As Reference
Set reference120 =
product5.CreateReferenceFromName("Product5/aroe
xterno.2/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(
Brp:(Sketch.1;3);Brp:(Sketch.2;7)));None:();Cf11:());
EdgeFillet.2_ResultOUT;Z0;G3055))")
Dim constraint58 As Constraint
Set constraint58 =
constraints10.AddBiEltCst(catCstTypeOn,
reference119, reference120)
```

```
Dim arrayOfVariantOfBSTR7(0)
arrayOfVariantOfBSTR7(0) = objPath & "\bolas.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR7, "All"

Dim arrayOfVariantOfBSTR8(0)
arrayOfVariantOfBSTR8(0) = objPath & "\jaula.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR8, "All"

Dim reference121 As Reference
Set reference121 = product5.CreateReferenceFromName("Product5/jaula.2/!Axis:(Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.2;3);Brp:(Sketch.1;9)));None:();Cf11:());CircPattern.1_ResultOUT;Z0;G3055))")
Dim constraint59 As Constraint
Set constraint59 = constraints10.AddBiEltCst(catCstTypeOn, reference120, reference121)

'fijamos la distancia entre la cara lateral del aro externo y el interno
Dim reference122 As Reference
Set reference122 = product5.CreateReferenceFromName("Product5/aroexterno.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;9)));None:();Cf11:());EdgeFillet.2_ResultOUT;Z0;G3055)")
Dim reference123 As Reference
Set reference123 = product5.CreateReferenceFromName("Product5/arointerno.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;18)));None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055)")
Dim constraint60 As Constraint
Set constraint60 = constraints10.AddBiEltCst(catCstTypeDistance, reference122, reference123)
Dim length29 As Length
Set length29 = constraint60.Dimension
length29.Value = 0#
constraint60.Orientation = catCstOrientSame

'idem para el aro externo y la jaula
Dim reference124 As Reference
Set reference124 = product5.CreateReferenceFromName("Product5/jaula.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.2;3);Brp:(Sketch.1;14)));None:();Cf11:());CircPattern.1_ResultOUT;Z0;G3055)")
Dim constraint61 As Constraint
Set constraint61 = constraints10.AddBiEltCst(catCstTypeDistance, reference122, reference124)
Dim length30 As Length
Set length30 = constraint61.Dimension
length30.Value = 0#
constraint61.Orientation = catCstOrientSame

'contacto de las caras de los rodamientos completos
Dim reference125 As Reference
Set reference125 = product5.CreateReferenceFromName("Product5/aroexterno.2/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;9)));None:();Cf11:());EdgeFillet.2_ResultOUT;Z0;G3055)")
Dim reference126 As Reference
Set reference126 = product5.CreateReferenceFromName("Product5/aroexterno.1/!Selection_RSur:(Face:(Brp:(Rib.1;0:(Brp:(Sketch.1;3);Brp:(Sketch.2;9)));None:();Cf11:());EdgeFillet.2_ResultOUT;Z0;G3055)")
Dim constraint62 As Constraint
Set constraint62 = constraints10.AddBiEltCst(catCstTypeSurfContact, reference125, reference126)

'Fijamos las bolas a los huecos de la jaula.
Dim referencec1 As Reference
Set referencec1 = product5.CreateReferenceFromName("Product5/bolas.1/!zx plane")
Dim referencec2 As Reference
Set referencec2 = product5.CreateReferenceFromName("Product5/jaula.1/!zx plane")
Dim constraintc1 As Constraint
Set constraintc1 = constraints10.AddBiEltCst(catCstTypeDistance, referencec1, referencec2)

Dim referencec3 As Reference
Set referencec3 = product5.CreateReferenceFromName("Product5/bolas.1/!yz plane")
Dim referencec4 As Reference
Set referencec4 = product5.CreateReferenceFromName("Product5/jaula.1/!yz plane")
Dim constraintc2 As Constraint
Set constraintc2 = constraints10.AddBiEltCst(catCstTypeDistance, referencec3, referencec4)

Dim referencec5 As Reference
Set referencec5 = product5.CreateReferenceFromName("Product5/bolas.1/!xy plane")
Dim referencec6 As Reference
Set referencec6 = product5.CreateReferenceFromName("Product5/jaula.1/!xy plane")
Dim constraintc3 As Constraint
Set constraintc3 = constraints10.AddBiEltCst(catCstTypeDistance, referencec5, referencec6)

product5.Update

'Fijamos las bolas a los huecos de la jaula.
Dim referenced1 As Reference
Set referenced1 = product5.CreateReferenceFromName("Product5/bolas.2/!zx plane")
Dim referenced2 As Reference
Set referenced2 = product5.CreateReferenceFromName("Product5/jaula.2/!zx plane")
```

```
Dim constraintd1 As Constraint
Set constraintd1 = constraints10.AddBiEltCst(catCstTypeDistance, referenced1, referenced2)

Dim referenced3 As Reference
Set referenced3 = product5.CreateReferenceFromName("Product5/bolas.2/!yz plane")
Dim referenced4 As Reference
Set referenced4 = product5.CreateReferenceFromName("Product5/jaula.2/!yz plane")
Dim constraintd2 As Constraint
Set constraintd2 = constraints10.AddBiEltCst(catCstTypeDistance, referenced3, referenced4)

Dim referenced5 As Reference
Set referenced5 = product5.CreateReferenceFromName("Product5/bolas.2/!xy plane")
Dim referenced6 As Reference
Set referenced6 = product5.CreateReferenceFromName("Product5/jaula.2/!xy plane")
Dim constraintd3 As Constraint
Set constraintd3 = constraints10.AddBiEltCst(catCstTypeDistance, referenced5, referenced6)

product5.Update
productDocument1.SaveAs objPath & "\AngularEnO.CATProduct"
```

- **Rodamiento axial de bolas.**

```
Private Sub CommandButton1_Click()
'=====================================
'-------------------------Anillo 1-------------------------
'=====================================

Dim documents1 As Documents
Set documents1 = CATIA.Documents
Dim partDocument1 As PartDocument
Set partDocument1 = documents1.Add("Part")
Dim part1 As Part
Set part1 = partDocument1.Part
Dim bodies1 As Bodies
Set bodies1 = part1.Bodies
Dim body1 As Body
Set body1 = bodies1.Item("PartBody")
Dim sketches1 As Sketches
Set sketches1 = body1.Sketches
Dim originElements1 As OriginElements
Set originElements1 = part1.OriginElements
Dim reference1 As Reference
Set reference1 = originElements1.PlaneYZ
Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)

'PAD
Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0#
arrayOfVariantOfDouble1(1) = 0#
arrayOfVariantOfDouble1(2) = 0#
arrayOfVariantOfDouble1(3) = 0#
arrayOfVariantOfDouble1(4) = 1#
arrayOfVariantOfDouble1(5) = 0#
arrayOfVariantOfDouble1(6) = 0#
arrayOfVariantOfDouble1(7) = 0#
arrayOfVariantOfDouble1(8) = 1#
Set sketch1Variant = sketch1
sketch1Variant.SetAbsoluteAxisData arrayOfVariantOfDouble1
part1.InWorkObject = sketch1

Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()
Dim geometricElements1 As GeometricElements
Set geometricElements1 = sketch1.GeometricElements
Dim axis2D1 As Axis2D
Set axis2D1 = geometricElements1.Item("AbsoluteAxis")
Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2

Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateClosedCircle(0#, 0#, D / 2)
Dim point2D1 As Point2D
Set point2D1 = axis2D1.GetItem("Origin")
circle2D1.CenterPoint = point2D1
circle2D1.ReportName = 3

Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints
Dim reference2 As Reference
Set reference2 = part1.CreateReferenceFromObject(circle2D1)
Dim constraint1 As Constraint
Set constraint1 = constraints1.AddMonoEltCst(catCstTypeRadius, reference2)
constraint1.Mode = catCstModeDrivingDimension
Dim length1 As Length
Set length1 = constraint1.Dimension
length1.Value = D / 2

sketch1.CloseEdition
part1.InWorkObject = sketch1
part1.Update

Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory
Dim pad1 As Pad
Set pad1 = shapeFactory1.AddNewPad(sketch1, 0.33 * H / 2)
Dim limit1 As Limit
Set limit1 = pad1.FirstLimit
Dim length2 As Length
Set length2 = limit1.Dimension
length2.Value = 0.33 * H / 2
pad1.IsSymmetric = True

part1.Update

'PoCKET
Dim reference3 As Reference
```

```
Set reference3 = part1.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")
Dim sketch2 As Sketch
Set sketch2 = sketches1.Add(reference3)

Dim arrayOfVariantOfDouble2(8)
arrayOfVariantOfDouble2(0) = 0.33 * H / 2
arrayOfVariantOfDouble2(1) = 0#
arrayOfVariantOfDouble2(2) = 0#
arrayOfVariantOfDouble2(3) = 0#
arrayOfVariantOfDouble2(4) = 1#
arrayOfVariantOfDouble2(5) = 0#
arrayOfVariantOfDouble2(6) = 0#
arrayOfVariantOfDouble2(7) = 0#
arrayOfVariantOfDouble2(8) = 1#
Set sketch2Variant = sketch2
sketch2Variant.SetAbsoluteAxisData arrayOfVariantOfDouble2
part1.InWorkObject = sketch2

Dim factory2D2 As Factory2D
Set factory2D2 = sketch2.OpenEdition()
Dim geometricElements2 As GeometricElements
Set geometricElements2 = sketch2.GeometricElements
Dim axis2D2 As Axis2D
Set axis2D2 = geometricElements2.Item("AbsoluteAxis")
Dim line2D3 As Line2D
Set line2D3 = axis2D2.GetItem("HDirection")
line2D3.ReportName = 1
Dim line2D4 As Line2D
Set line2D4 = axis2D2.GetItem("VDirection")
line2D4.ReportName = 2

Dim circle2D2 As Circle2D
Set circle2D2 = factory2D2.CreateClosedCircle(0#, 0#, D1 / 2)
Dim point2D2 As Point2D
Set point2D2 = axis2D2.GetItem("Origin")
circle2D2.CenterPoint = point2D2
circle2D2.ReportName = 3

Dim constraints2 As Constraints
Set constraints2 = sketch2.Constraints
Dim reference4 As Reference
Set reference4 = part1.CreateReferenceFromObject(circle2D2)
Dim constraint2 As Constraint
Set constraint2 = constraints2.AddMonoEltCst(catCstTypeRadius, reference4)
constraint2.Mode = catCstModeDrivingDimension
Dim length3 As Length
Set length3 = constraint2.Dimension
length3.Value = D1 / 2

sketch2.CloseEdition
part1.InWorkObject = sketch2
part1.Update

Dim pocket1 As Pocket
Set pocket1 = shapeFactory1.AddNewPocket(sketch2, 0.33 * H)
Dim limit2 As Limit
Set limit2 = pocket1.FirstLimit
limit2.LimitMode = catUpToLastLimit

part1.Update

'Chaflán
Dim reference5 As Reference
Set reference5 = part1.CreateReferenceFromName("")
Dim constRadEdgeFillet1 As ConstRadEdgeFillet
Set constRadEdgeFillet1 = shapeFactory1.AddNewSolidEdgeFilletWithConstantRadius(reference5, catTangencyFilletEdgePropagation, r)

Dim reference6 As Reference
Set reference6 = part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());Face:(Brp:(Pad.1;2);None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket1)
constRadEdgeFillet1.AddObjectToFillet reference6

Dim reference7 As Reference
Set reference7 = part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;1);None:();Cf11:());Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket1)
constRadEdgeFillet1.AddObjectToFillet reference7

Dim reference8 As Reference
Set reference8 = part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;2);None:();Cf11:());Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket1)
constRadEdgeFillet1.AddObjectToFillet reference8

Dim reference9 As Reference
Set reference9 = part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());Face:(Brp:(Pad.1;1);None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket1)
constRadEdgeFillet1.AddObjectToFillet reference9

constRadEdgeFillet1.EdgePropagation = catTangencyFilletEdgePropagation

part1.Update

'Slot
Dim reference10 As Reference
Set reference10 = originElements1.PlaneXY
Dim sketch3 As Sketch
Set sketch3 = sketches1.Add(reference10)

Dim arrayOfVariantOfDouble3(8)
```

```
arrayOfVariantOfDouble3(0) = 0#
arrayOfVariantOfDouble3(1) = 0#
arrayOfVariantOfDouble3(2) = 0#
arrayOfVariantOfDouble3(3) = 1#
arrayOfVariantOfDouble3(4) = 0#
arrayOfVariantOfDouble3(5) = 0#
arrayOfVariantOfDouble3(6) = 0#
arrayOfVariantOfDouble3(7) = 1#
arrayOfVariantOfDouble3(8) = 0#
Set sketch3Variant = sketch3
sketch3Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble3
part1.InWorkObject = sketch3

Dim factory2D3 As Factory2D
Set factory2D3 = sketch3.OpenEdition()
Dim geometricElements3 As GeometricElements
Set geometricElements3 =
sketch3.GeometricElements
Dim axis2D3 As Axis2D
Set axis2D3 =
geometricElements3.Item("AbsoluteAxis")
Dim line2D5 As Line2D
Set line2D5 = axis2D3.GetItem("HDirection")
line2D5.ReportName = 1
Dim line2D6 As Line2D
Set line2D6 = axis2D3.GetItem("VDirection")
line2D6.ReportName = 2

Dim point2D3 As Point2D
Set point2D3 = factory2D3.CreatePoint(-0.67 * H / 2,
(D - D1) / 4 + D1 / 2)
point2D3.ReportName = 3
Dim circle2D3 As Circle2D
Set circle2D3 = factory2D3.CreateClosedCircle(-0.67
* H / 2, (D - D1) / 4 + D1 / 2, 0.44 * H / 2)
circle2D3.CenterPoint = point2D3
circle2D3.ReportName = 4

Dim constraints3 As Constraints
Set constraints3 = sketch3.Constraints
Dim reference11 As Reference
Set reference11 =
part1.CreateReferenceFromObject(circle2D3)
Dim constraint3 As Constraint
Set constraint3 =
constraints3.AddMonoEltCst(catCstTypeRadius,
reference11)
constraint3.Mode = catCstModeDrivingDimension
Dim length5 As Length
Set length5 = constraint3.Dimension
length5.Value = 0.44 * H / 2

Dim reference12 As Reference
Set reference12 =
part1.CreateReferenceFromObject(line2D5)
Dim reference13 As Reference
Set reference13 =
part1.CreateReferenceFromObject(point2D3)
Dim constraint4 As Constraint
Set constraint4 =
constraints3.AddBiEltCst(catCstTypeDistance,
reference12, reference13)
constraint4.Mode = catCstModeDrivingDimension
Dim length6 As Length
Set length6 = constraint4.Dimension
length6.Value = (D - D1) / 4 + D1 / 2

Dim reference14 As Reference
Set reference14 =
part1.CreateReferenceFromObject(line2D6)
Dim reference15 As Reference
Set reference15 =
part1.CreateReferenceFromObject(point2D3)
Dim constraint5 As Constraint
Set constraint5 =
constraints3.AddBiEltCst(catCstTypeDistance,
reference14, reference15)
constraint5.Mode = catCstModeDrivingDimension
Dim length7 As Length
Set length7 = constraint5.Dimension
length7.Value = 0.67 * H / 2

sketch3.CloseEdition
part1.InWorkObject = sketch3
part1.Update

'Plano auxiliar
Dim hybridShapeFactory1 As HybridShapeFactory
Set hybridShapeFactory1 =
part1.HybridShapeFactory
Dim hybridShapePlaneExplicit1 As
HybridShapePlaneExplicit
Set hybridShapePlaneExplicit1 =
originElements1.PlaneYZ
Dim reference16 As Reference
Set reference16 =
part1.CreateReferenceFromObject(hybridShapePlane
Explicit1)
Dim hybridShapePlaneOffset1 As
HybridShapePlaneOffset
Set hybridShapePlaneOffset1 =
hybridShapeFactory1.AddNewPlaneOffset(reference
16, 0.67 * H / 2, True)
body1.InsertHybridShape hybridShapePlaneOffset1
part1.InWorkObject = hybridShapePlaneOffset1

part1.Update

'guía
Dim hybridShapes1 As HybridShapes
Set hybridShapes1 = body1.HybridShapes
Dim reference17 As Reference
Set reference17 = hybridShapes1.Item("Plane.1")
Dim sketch4 As Sketch
Set sketch4 = sketches1.Add(reference17)

Dim arrayOfVariantOfDouble4(8)
arrayOfVariantOfDouble4(0) = -0.67 * H / 2
arrayOfVariantOfDouble4(1) = 0#
arrayOfVariantOfDouble4(2) = 0#
arrayOfVariantOfDouble4(3) = 0#
arrayOfVariantOfDouble4(4) = 1#
arrayOfVariantOfDouble4(5) = 0#
arrayOfVariantOfDouble4(6) = 0#
arrayOfVariantOfDouble4(7) = 0#
arrayOfVariantOfDouble4(8) = 1#
Set sketch4Variant = sketch4
sketch4Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble4
part1.InWorkObject = sketch4

Dim factory2D4 As Factory2D
Set factory2D4 = sketch4.OpenEdition()
```

```
Dim geometricElements4 As GeometricElements
Set geometricElements4 = sketch4.GeometricElements
Dim axis2D4 As Axis2D
Set axis2D4 = geometricElements4.Item("AbsoluteAxis")
Dim line2D7 As Line2D
Set line2D7 = axis2D4.GetItem("HDirection")
line2D7.ReportName = 1
Dim line2D8 As Line2D
Set line2D8 = axis2D4.GetItem("VDirection")
line2D8.ReportName = 2
Dim circle2D4 As Circle2D
Set circle2D4 = factory2D4.CreateClosedCircle(0#, 0#, (D - D1) / 4 + D1 / 2)
Dim point2D4 As Point2D
Set point2D4 = axis2D4.GetItem("Origin")
circle2D4.CenterPoint = point2D4
circle2D4.ReportName = 3

Dim constraints4 As Constraints
Set constraints4 = sketch4.Constraints
Dim reference18 As Reference
Set reference18 = part1.CreateReferenceFromObject(circle2D4)
Dim constraint6 As Constraint
Set constraint6 = constraints4.AddMonoEltCst(catCstTypeRadius, reference18)
constraint6.Mode = catCstModeDrivingDimension
Dim length8 As Length
Set length8 = constraint6.Dimension
length8.Value = (D - D1) / 4 + D1 / 2

sketch4.CloseEdition
part1.InWorkObject = sketch4
part1.Update

Dim slot1 As Slot
Set slot1 = shapeFactory1.AddNewSlotFromRef(Nothing, Nothing)
Dim reference19 As Reference
Set reference19 = part1.CreateReferenceFromObject(sketch3)
slot1.SetProfileElement reference19
Dim reference20 As Reference
Set reference20 = part1.CreateReferenceFromObject(sketch4)
slot1.CenterCurveElement = reference20
part1.UpdateObject slot1

Dim product1 As Product
Set product1 = partDocument1.GetItem("Part1")
product1.PartNumber = "aro1"
partDocument1.SaveAs objPath & "\aro1.catpart"
partDocument1.Close

'====================================
'----------------------------------Anillo 2----------------
'====================================
Dim documents2 As Documents
Set documents2 = CATIA.Documents
Dim partDocument2 As PartDocument
Set partDocument2 = documents2.Add("Part")
Dim part2 As Part
Set part2 = partDocument2.Part

Dim bodies2 As Bodies
Set bodies2 = part2.Bodies
Dim body2 As Body
Set body2 = bodies2.Item("PartBody")
Dim sketches2 As Sketches
Set sketches2 = body2.Sketches
Dim originElements2 As OriginElements
Set originElements2 = part2.OriginElements
Dim reference21 As Reference
Set reference21 = originElements2.PlaneYZ
Dim sketch5 As Sketch
Set sketch5 = sketches2.Add(reference21)

'PAD
Dim arrayOfVariantOfDouble5(8)
arrayOfVariantOfDouble5(0) = 0#
arrayOfVariantOfDouble5(1) = 0#
arrayOfVariantOfDouble5(2) = 0#
arrayOfVariantOfDouble5(3) = 0#
arrayOfVariantOfDouble5(4) = 1#
arrayOfVariantOfDouble5(5) = 0#
arrayOfVariantOfDouble5(6) = 0#
arrayOfVariantOfDouble5(7) = 0#
arrayOfVariantOfDouble5(8) = 1#
Set sketch5Variant = sketch5
sketch5Variant.SetAbsoluteAxisData arrayOfVariantOfDouble5
part2.InWorkObject = sketch5

Dim factory2D5 As Factory2D
Set factory2D5 = sketch5.OpenEdition()
Dim geometricElements5 As GeometricElements
Set geometricElements5 = sketch5.GeometricElements
Dim axis2D5 As Axis2D
Set axis2D5 = geometricElements5.Item("AbsoluteAxis")
Dim line2D9 As Line2D
Set line2D9 = axis2D5.GetItem("HDirection")
line2D9.ReportName = 1
Dim line2D10 As Line2D
Set line2D10 = axis2D5.GetItem("VDirection")
line2D10.ReportName = 2

Dim circle2D5 As Circle2D
Set circle2D5 = factory2D5.CreateClosedCircle(0#, 0#, d_1 / 2)
Dim point2D5 As Point2D
Set point2D5 = axis2D5.GetItem("Origin")
circle2D5.CenterPoint = point2D5
circle2D5.ReportName = 3

Dim constraints5 As Constraints
Set constraints5 = sketch5.Constraints
Dim reference22 As Reference
Set reference22 = part2.CreateReferenceFromObject(circle2D5)
Dim constraint7 As Constraint
Set constraint7 = constraints5.AddMonoEltCst(catCstTypeRadius, reference22)
constraint7.Mode = catCstModeDrivingDimension
Dim length9 As Length
Set length9 = constraint7.Dimension
length9.Value = d_1 / 2

sketch5.CloseEdition
```

```
part2.InWorkObject = sketch5
part2.Update

Dim shapeFactory2 As ShapeFactory
Set shapeFactory2 = part2.ShapeFactory
Dim pad2 As Pad
Set pad2 = shapeFactory2.AddNewPad(sketch5, 0.33
* H / 2)
Dim limit3 As Limit
Set limit3 = pad2.FirstLimit
Dim length10 As Length
Set length10 = limit3.Dimension
length10.Value = 0.33 * H / 2
pad2.IsSymmetric = True

part2.Update

'PoCKET
Dim reference23 As Reference
Set reference23 =
part2.CreateReferenceFromName("Selection_RSur:(
Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultO
UT;Z0;G3055)")
Dim sketch6 As Sketch
Set sketch6 = sketches2.Add(reference23)

Dim arrayOfVariantOfDouble6(8)
arrayOfVariantOfDouble6(0) = 0.33 * H / 2
arrayOfVariantOfDouble6(1) = 0#
arrayOfVariantOfDouble6(2) = 0#
arrayOfVariantOfDouble6(3) = 0#
arrayOfVariantOfDouble6(4) = 1#
arrayOfVariantOfDouble6(5) = 0#
arrayOfVariantOfDouble6(6) = 0#
arrayOfVariantOfDouble6(7) = 0#
arrayOfVariantOfDouble6(8) = 1#
Set sketch6Variant = sketch6
sketch6Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble6
part2.InWorkObject = sketch6

Dim factory2D6 As Factory2D
Set factory2D6 = sketch6.OpenEdition()
Dim geometricElements6 As GeometricElements
Set geometricElements6 =
sketch6.GeometricElements
Dim axis2D6 As Axis2D
Set axis2D6 =
geometricElements6.Item("AbsoluteAxis")
Dim line2D11 As Line2D
Set line2D11 = axis2D6.GetItem("HDirection")
line2D11.ReportName = 1
Dim line2D12 As Line2D
Set line2D12 = axis2D6.GetItem("VDirection")
line2D12.ReportName = 2

Dim circle2D7 As Circle2D
Set circle2D7 = factory2D6.CreateClosedCircle(0#,
0#, d_ / 2)
Dim point2D7 As Point2D
Set point2D7 = axis2D6.GetItem("Origin")
circle2D7.CenterPoint = point2D7
circle2D7.ReportName = 3

Dim constraints6 As Constraints
Set constraints6 = sketch6.Constraints
Dim reference24 As Reference
Set reference24 =
part2.CreateReferenceFromObject(circle2D7)
Dim constraint8 As Constraint
Set constraint8 =
constraints6.AddMonoEltCst(catCstTypeRadius,
reference24)
constraint8.Mode = catCstModeDrivingDimension
Dim length11 As Length
Set length11 = constraint8.Dimension
length11.Value = d_ / 2

sketch6.CloseEdition
part2.InWorkObject = sketch6
part2.Update

Dim pocket2 As Pocket
Set pocket2 =
shapeFactory2.AddNewPocket(sketch6, 0.33 * H)
Dim limit4 As Limit
Set limit4 = pocket2.FirstLimit
limit4.LimitMode = catUpToLastLimit

part2.Update

'Chaflán
Dim reference25 As Reference
Set reference25 =
part2.CreateReferenceFromName("")
Dim constRadEdgeFillet2 As ConstRadEdgeFillet
Set constRadEdgeFillet2 =
shapeFactory2.AddNewSolidEdgeFilletWithConstant
Radius(reference25,
catTangencyFilletEdgePropagation, r)

Dim reference26 As Reference
Set reference26 =
part2.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf
11:());Face:(Brp:(Pad.1;2);None:();Cf11:());None:(Li
mits1:();Limits2:());Cf11:());WithTemporaryBody;W
ithoutBuildError;WithSelectingFeatureSupport;MFB
RepVersion_CXR15)", pocket2)
constRadEdgeFillet2.AddObjectToFillet reference26

Dim reference27 As Reference
Set reference27 =
part2.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;1);None:();Cf11:());Face:(Brp:(P
ad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());None:(Li
mits1:();Limits2:());Cf11:());WithTemporaryBody;W
ithoutBuildError;WithSelectingFeatureSupport;MFB
RepVersion_CXR15)", pocket2)
constRadEdgeFillet2.AddObjectToFillet reference27

Dim reference28 As Reference
Set reference28 =
part2.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;2);None:();Cf11:());Face:(Brp:(P
ocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());None:
(Limits1:();Limits2:());Cf11:());WithTemporaryBody
;WithoutBuildError;WithSelectingFeatureSupport;M
FBRepVersion_CXR15)", pocket2)
constRadEdgeFillet2.AddObjectToFillet reference28

Dim reference29 As Reference
Set reference29 =
part2.CreateReferenceFromBRepName("REdge:(Edg
```

```
e:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();
Cf11:());Face:(Brp:(Pad.1;1);None:();Cf11:());None:(
Limits1:();Limits2:());Cf11:());WithTemporaryBody;
WithoutBuildError;WithSelectingFeatureSupport;MF
BRepVersion_CXR15)", pocket2)
constRadEdgeFillet2.AddObjectToFillet reference29
constRadEdgeFillet2.EdgePropagation =
catTangencyFilletEdgePropagation
part2.Update

'Slot
Dim reference30 As Reference
Set reference30 = originElements2.PlaneXY
Dim sketch7 As Sketch
Set sketch7 = sketches2.Add(reference30)

Dim arrayOfVariantOfDouble7(8)
arrayOfVariantOfDouble7(0) = 0#
arrayOfVariantOfDouble7(1) = 0#
arrayOfVariantOfDouble7(2) = 0#
arrayOfVariantOfDouble7(3) = 1#
arrayOfVariantOfDouble7(4) = 0#
arrayOfVariantOfDouble7(5) = 0#
arrayOfVariantOfDouble7(6) = 0#
arrayOfVariantOfDouble7(7) = 1#
arrayOfVariantOfDouble7(8) = 0#
Set sketch7Variant = sketch7
sketch7Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble7
part2.InWorkObject = sketch7

Dim factory2D7 As Factory2D
Set factory2D7 = sketch7.OpenEdition()
Dim geometricElements7 As GeometricElements
Set geometricElements7 =
sketch7.GeometricElements
Dim axis2D7 As Axis2D
Set axis2D7 =
geometricElements7.Item("AbsoluteAxis")
Dim line2D13 As Line2D
Set line2D13 = axis2D7.GetItem("HDirection")
line2D13.ReportName = 1
Dim line2D14 As Line2D
Set line2D14 = axis2D7.GetItem("VDirection")
line2D14.ReportName = 2

Dim point2D8 As Point2D
Set point2D8 = factory2D7.CreatePoint(0.67 * H / 2,
((d_1 - d_) / 4) + d_ / 2)
point2D8.ReportName = 3
Dim circle2D8 As Circle2D
Set circle2D8 = factory2D7.CreateClosedCircle(0.67
* H / 2, ((d_1 - d_) / 4) + d_ / 2, 0.44 * H / 2)
circle2D8.CenterPoint = point2D8
circle2D8.ReportName = 4

Dim constraints7 As Constraints
Set constraints7 = sketch7.Constraints
Dim reference31 As Reference
Set reference31 =
part2.CreateReferenceFromObject(circle2D8)
Dim constraint9 As Constraint
Set constraint9 =
constraints7.AddMonoEltCst(catCstTypeRadius,
reference31)
constraint9.Mode = catCstModeDrivingDimension
Dim length12 As Length
Set length12 = constraint9.Dimension
length12.Value = 0.44 * H / 2

Dim reference32 As Reference
Set reference32 =
part2.CreateReferenceFromObject(line2D13)
Dim reference33 As Reference
Set reference33 =
part2.CreateReferenceFromObject(point2D8)
Dim constraint10 As Constraint
Set constraint10 =
constraints7.AddBiEltCst(catCstTypeDistance,
reference32, reference33)
constraint10.Mode = catCstModeDrivingDimension
Dim length13 As Length
Set length13 = constraint10.Dimension
length13.Value = ((d_1 - d_) / 4) + d_ / 2

Dim reference34 As Reference
Set reference34 =
part2.CreateReferenceFromObject(line2D14)
Dim reference35 As Reference
Set reference35 =
part2.CreateReferenceFromObject(point2D8)
Dim constraint11 As Constraint
Set constraint11 =
constraints7.AddBiEltCst(catCstTypeDistance,
reference34, reference35)
constraint11.Mode = catCstModeDrivingDimension
Dim length14 As Length
Set length14 = constraint11.Dimension
length14.Value = 0.67 * H / 2

sketch7.CloseEdition
part2.InWorkObject = sketch7
part2.Update

'Plano auxiliar
Dim hybridShapeFactory2 As HybridShapeFactory
Set hybridShapeFactory2 =
part2.HybridShapeFactory
Dim hybridShapePlaneExplicit2 As
HybridShapePlaneExplicit
Set hybridShapePlaneExplicit2 =
originElements2.PlaneYZ
Dim reference36 As Reference
Set reference36 =
part2.CreateReferenceFromObject(hybridShapePlane
Explicit2)
Dim hybridShapePlaneOffset2 As
HybridShapePlaneOffset
Set hybridShapePlaneOffset2 =
hybridShapeFactory2.AddNewPlaneOffset(reference
36, 0.67 * H / 2, True)
body2.InsertHybridShape hybridShapePlaneOffset2
part2.InWorkObject = hybridShapePlaneOffset2

part2.Update

'guía
Dim hybridShapes2 As HybridShapes
Set hybridShapes2 = body2.HybridShapes
Dim reference37 As Reference
Set reference37 = hybridShapes2.Item("Plane.1")
Dim sketch8 As Sketch
Set sketch8 = sketches2.Add(reference37)
```

```
Dim arrayOfVariantOfDouble8(8)
arrayOfVariantOfDouble8(0) = 0.67 * H / 2
arrayOfVariantOfDouble8(1) = 0#
arrayOfVariantOfDouble8(2) = 0#
arrayOfVariantOfDouble8(3) = 0#
arrayOfVariantOfDouble8(4) = 1#
arrayOfVariantOfDouble8(5) = 0#
arrayOfVariantOfDouble8(6) = 0#
arrayOfVariantOfDouble8(7) = 0#
arrayOfVariantOfDouble8(8) = 1#
Set sketch8Variant = sketch8
sketch8Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble8
part2.InWorkObject = sketch8

Dim factory2D8 As Factory2D
Set factory2D8 = sketch8.OpenEdition()
Dim geometricElements8 As GeometricElements
Set geometricElements8 =
sketch8.GeometricElements
Dim axis2D8 As Axis2D
Set axis2D8 =
geometricElements8.Item("AbsoluteAxis")
Dim line2D15 As Line2D
Set line2D15 = axis2D8.GetItem("HDirection")
line2D15.ReportName = 1
Dim line2D16 As Line2D
Set line2D16 = axis2D8.GetItem("VDirection")
line2D16.ReportName = 2

Dim circle2D9 As Circle2D
Set circle2D9 = factory2D8.CreateClosedCircle(0#,
0#, ((d_1 - d_) / 4) + d_ / 2)
Dim point2D9 As Point2D
Set point2D9 = axis2D8.GetItem("Origin")
circle2D9.CenterPoint = point2D9
circle2D9.ReportName = 3

Dim constraints8 As Constraints
Set constraints8 = sketch8.Constraints
Dim reference38 As Reference
Set reference38 =
part2.CreateReferenceFromObject(circle2D9)
Dim constraint12 As Constraint
Set constraint12 =
constraints8.AddMonoEltCst(catCstTypeRadius,
reference38)
constraint12.Mode = catCstModeDrivingDimension
Dim length15 As Length
Set length15 = constraint10.Dimension
length15.Value = ((d_1 - d_) / 4) + d_ / 2

sketch8.CloseEdition
part2.InWorkObject = sketch8
part2.Update

Dim slot2 As Slot
Set slot2 =
shapeFactory2.AddNewSlotFromRef(Nothing,
Nothing)
Dim reference39 As Reference
Set reference39 =
part2.CreateReferenceFromObject(sketch7)
slot2.SetProfileElement reference39
Dim reference40 As Reference
Set reference40 =
part2.CreateReferenceFromObject(sketch8)
slot2.CenterCurveElement = reference40

part2.UpdateObject slot2

Dim product2 As Product
Set product2 = partDocument2.GetItem("Part2")
product2.PartNumber = "aro2"
partDocument2.SaveAs objPath & "\aro2.CATPart"
partDocument2.Close

'=====================================
'----------------------------------------Bola----------------
'=====================================

Dim documents3 As Documents
Set documents3 = CATIA.Documents
Dim partDocument3 As PartDocument
Set partDocument3 = documents3.Add("Part")
Dim part3 As Part
Set part3 = partDocument3.Part
Dim bodies3 As Bodies
Set bodies3 = part3.Bodies
Dim body3 As Body
Set body3 = bodies3.Item("PartBody")
Dim sketches3 As Sketches
Set sketches3 = body3.Sketches
Dim originElements3 As OriginElements
Set originElements3 = part3.OriginElements
Dim reference41 As Reference
Set reference41 = originElements3.PlaneYZ
Dim sketch9 As Sketch
Set sketch9 = sketches3.Add(reference41)

Dim arrayOfVariantOfDouble9(8)
arrayOfVariantOfDouble9(0) = 0#
arrayOfVariantOfDouble9(1) = 0#
arrayOfVariantOfDouble9(2) = 0#
arrayOfVariantOfDouble9(3) = 0#
arrayOfVariantOfDouble9(4) = 1#
arrayOfVariantOfDouble9(5) = 0#
arrayOfVariantOfDouble9(6) = 0#
arrayOfVariantOfDouble9(7) = 0#
arrayOfVariantOfDouble9(8) = 1#
Set sketch9Variant = sketch9
sketch9Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble9
part3.InWorkObject = sketch9

Dim factory2D9 As Factory2D
Set factory2D9 = sketch9.OpenEdition()
Dim geometricElements9 As GeometricElements
Set geometricElements9 =
sketch9.GeometricElements
Dim axis2D9 As Axis2D
Set axis2D9 =
geometricElements9.Item("AbsoluteAxis")
Dim line2D17 As Line2D
Set line2D17 = axis2D9.GetItem("HDirection")
line2D17.ReportName = 1
Dim line2D18 As Line2D
Set line2D18 = axis2D9.GetItem("VDirection")
line2D18.ReportName = 2

Dim point2D10 As Point2D
Set point2D10 = factory2D9.CreatePoint(0#, (D -
D1) / 4 + D1 / 2 + (0.44 * H / 2))
point2D10.ReportName = 3
```

```
Dim point2D11 As Point2D
Set point2D11 = factory2D9.CreatePoint(0#, (D - D1) / 4 + D1 / 2 - (0.44 * H / 2))
point2D11.ReportName = 4
Dim line2D19 As Line2D
Set line2D19 = factory2D9.CreateLine(0#, (D - D1) / 4 + D1 / 2 + (0.44 * H / 2), 0#, (D - D1) / 2 + D1 / 4 - (0.44 * H / 2))
line2D19.ReportName = 5
line2D19.StartPoint = point2D10
line2D19.EndPoint = point2D11

Dim constraints9 As Constraints
Set constraints9 = sketch9.Constraints
Dim reference42 As Reference
Set reference42 = part3.CreateReferenceFromObject(point2D10)
Dim reference43 As Reference
Set reference43 = part3.CreateReferenceFromObject(line2D18)
Dim constraint13 As Constraint
Set constraint13 = constraints9.AddBiEltCst(catCstTypeOn, reference42, reference43)
constraint13.Mode = catCstModeDrivingDimension

Dim reference44 As Reference
Set reference44 = part3.CreateReferenceFromObject(line2D19)
Dim reference45 As Reference
Set reference45 = part3.CreateReferenceFromObject(line2D18)
Dim constraint14 As Constraint
Set constraint14 = constraints9.AddBiEltCst(catCstTypeVerticality, reference44, reference45)
constraint14.Mode = catCstModeDrivingDimension

Dim reference46 As Reference
Set reference46 = part3.CreateReferenceFromObject(line2D17)
Dim reference47 As Reference
Set reference47 = part3.CreateReferenceFromObject(point2D10)
Dim constraint15 As Constraint
Set constraint15 = constraints9.AddBiEltCst(catCstTypeDistance, reference46, reference47)
constraint15.Mode = catCstModeDrivingDimension
Dim length16 As Length
Set length16 = constraint15.Dimension
length16.Value = (D - D1) / 4 + D1 / 2 + (0.44 * (H / 2))

Dim circle2D10 As Circle2D
Set circle2D10 = factory2D9.CreateCircle(0#, (D - D1) / 4 + D1 / 2, (0.44 * H / 2), 4.712389, 7.853982)

Dim point2D12 As Point2D
Set point2D12 = factory2D9.CreatePoint(0, (D - D1) / 4 + D1 / 2)
circle2D10.CenterPoint = point2D12
circle2D10.ReportName = 6
circle2D10.StartPoint = point2D10
circle2D10.EndPoint = point2D11

sketch9.CloseEdition
part3.InWorkObject = sketch9
part3.Update

Dim shapeFactory3 As ShapeFactory
Set shapeFactory3 = part3.ShapeFactory

Dim shaft1 As Shaft
Set shaft1 = shapeFactory3.AddNewShaft(sketch9)

'selección del eje de revolución
Dim ejederev As Reference
Set ejederev = part3.CreateReferenceFromObject(line2D18)
shaft1.RevoluteAxis = ejederev
part3.Update

Dim referenceb As Reference
Set referenceb = part3.CreateReferenceFromName("")
Dim referenceb1 As Reference
Set referenceb1 = part3.CreateReferenceFromName("")
Dim circPatternb As CircPattern
Set circPatternb = shapeFactory3.AddNewCircPattern(Nothing, 1, 2, 20#, 45#, 1, 1, referenceb, referenceb1, True, 0#, True)
circPatternb.CircularPatternParameters = catInstancesandAngularSpacing
Dim angularRepartitionb As angularRepartition
Set angularRepartitionb = circPatternb.angularRepartition

'calculo del número de agujeros para ello definimos primero el número pi
Const PIb = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es D1+D/4
Dim longitudb As Double
longitudb = 2 * PIb * ((D - D1) / 4 + D1 / 2)
'crearemos una variable auxiliar que usara el comando matemático Round para redondear el numero de huecos que tendrá el separador.
'para calcular este número hacemos la hipótesis de que la distancia entre BOLAS será de 5 mm
Dim variableauxb As Double
variableauxb = Math.Round(longitudb / ((0.44 * H) + (0.22 * H)))
'ya tenemos el numero de huecos del separador
Dim intParamb As intParam
Set intParamb = angularRepartitionb.InstancesCount
intParamb.Value = variableauxb
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angleb As angle
Set angleb = angularRepartitionb.AngularSpacing
angleb.Value = 360 / variableauxb

circPatternb.SetRotationAxis reference41

part3.UpdateObject circPatternb
part3.Update

Dim product3 As Product
Set product3 = partDocument3.GetItem("Part3")
product3.PartNumber = "bolas"
```

```
partDocument3.SaveAs objPath & "\bolas.CATPart"
partDocument3.Close

'=====================================
'----------Separador----------
'=====================================
Dim documents4 As Documents
Set documents4 = CATIA.Documents
Dim partDocument4 As PartDocument
Set partDocument4 = documents4.Add("Part")
Dim part4 As Part
Set part4 = partDocument4.Part
Dim bodies4 As Bodies
Set bodies4 = part4.Bodies
Dim body4 As Body
Set body4 = bodies4.Item("PartBody")
Dim sketches4 As Sketches
Set sketches4 = body4.Sketches
Dim originElements4 As OriginElements
Set originElements4 = part4.OriginElements
Dim reference48 As Reference
Set reference48 = originElements4.PlaneYZ
Dim sketch10 As Sketch
Set sketch10 = sketches4.Add(reference48)

Dim arrayOfVariantOfDouble10(8)
arrayOfVariantOfDouble10(0) = 0#
arrayOfVariantOfDouble10(1) = 0#
arrayOfVariantOfDouble10(2) = 0#
arrayOfVariantOfDouble10(3) = 0#
arrayOfVariantOfDouble10(4) = 1#
arrayOfVariantOfDouble10(5) = 0#
arrayOfVariantOfDouble10(6) = 0#
arrayOfVariantOfDouble10(7) = 0#
arrayOfVariantOfDouble10(8) = 1#
Set sketch10Variant = sketch10
sketch10Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble10

part4.InWorkObject = sketch10

Dim factory2D10 As Factory2D
Set factory2D10 = sketch10.OpenEdition()
Dim geometricElements10 As GeometricElements
Set geometricElements10 =
sketch10.GeometricElements
Dim axis2D10 As Axis2D
Set axis2D10 =
geometricElements10.Item("AbsoluteAxis")
Dim line2D20 As Line2D
Set line2D20 = axis2D10.GetItem("HDirection")
line2D20.ReportName = 1
Dim line2D21 As Line2D
Set line2D21 = axis2D10.GetItem("VDirection")
line2D21.ReportName = 2
Dim circle2D11 As Circle2D
Set circle2D11 =
factory2D10.CreateClosedCircle(0#, 0#, D / 2)
Dim point2D13 As Point2D
Set point2D13 = axis2D10.GetItem("Origin")
circle2D11.CenterPoint = point2D13
circle2D11.ReportName = 3

Dim constraints10 As Constraints
Set constraints10 = sketch10.Constraints
Dim reference49 As Reference
Set reference49 =
part4.CreateReferenceFromObject(circle2D11)
Dim constraint16 As Constraint
Set constraint16 =
constraints10.AddMonoEltCst(catCstTypeRadius,
reference49)
constraint16.Mode = catCstModeDrivingDimension
Dim length17 As Length
Set length17 = constraint16.Dimension
length17.Value = D / 2

sketch10.CloseEdition
part4.InWorkObject = sketch10
part4.Update

Dim shapeFactory4 As ShapeFactory
Set shapeFactory4 = part4.ShapeFactory
Dim pad3 As Pad
Set pad3 = shapeFactory4.AddNewPad(sketch10,
0.22 * H)
pad3.IsSymmetric = True
Dim limit5 As Limit
Set limit5 = pad3.FirstLimit
Dim length18 As Length
Set length18 = limit5.Dimension
length18.Value = 0.11 * H
part4.Update

'POCKET
Dim reference50 As Reference
Set reference50 =
part4.CreateReferenceFromName("Selection_RSur:(
Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultO
UT;Z0;G3055)")

Dim sketch11 As Sketch
Set sketch11 = sketches4.Add(reference50)

Dim arrayOfVariantOfDouble11(8)
arrayOfVariantOfDouble11(0) = 0.11 * H
arrayOfVariantOfDouble11(1) = 0#
arrayOfVariantOfDouble11(2) = 0#
arrayOfVariantOfDouble11(3) = 0#
arrayOfVariantOfDouble11(4) = 1#
arrayOfVariantOfDouble11(5) = 0#
arrayOfVariantOfDouble11(6) = 0#
arrayOfVariantOfDouble11(7) = 0#
arrayOfVariantOfDouble11(8) = 1#
Set sketch11Variant = sketch11
sketch11Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble11
part4.InWorkObject = sketch11

Dim factory2D11 As Factory2D
Set factory2D11 = sketch11.OpenEdition()
Dim geometricElements11 As GeometricElements
Set geometricElements11 =
sketch11.GeometricElements
Dim axis2D11 As Axis2D
Set axis2D11 =
geometricElements11.Item("AbsoluteAxis")
Dim line2D22 As Line2D
Set line2D22 = axis2D11.GetItem("HDirection")
line2D22.ReportName = 1
Dim line2D23 As Line2D
Set line2D23 = axis2D11.GetItem("VDirection")
line2D23.ReportName = 2
```

```
Dim circle2D12 As Circle2D
Set circle2D12 = factory2D11.CreateClosedCircle(0#, 0#, D1 / 2)
Dim point2D14 As Point2D
Set point2D14 = axis2D11.GetItem("Origin")
circle2D12.CenterPoint = point2D14
circle2D12.ReportName = 3

Dim constraints11 As Constraints
Set constraints11 = sketch11.Constraints
Dim reference51 As Reference
Set reference51 = part4.CreateReferenceFromObject(circle2D12)
Dim constraint17 As Constraint
Set constraint17 = constraints11.AddMonoEltCst(catCstTypeRadius, reference51)
constraint17.Mode = catCstModeDrivingDimension
Dim length19 As Length
Set length19 = constraint17.Dimension
length19.Value = D1 / 2

sketch11.CloseEdition
part4.InWorkObject = sketch11
part4.Update

Dim pocket3 As Pocket
Set pocket3 = shapeFactory4.AddNewPocket(sketch11, 0.22 * H)
Dim limit6 As Limit
Set limit6 = pocket3.FirstLimit
limit6.LimitMode = catUpToLastLimit
part4.Update

'Agujeros del separador
Dim cara As Reference
Set cara = part4.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pocket.1_ResultOUT;Z0;G3055)")

Dim sketch12 As Sketch
Set sketch12 = sketches4.Add(cara)

Dim arrayOfVariantOfDouble12(8)
arrayOfVariantOfDouble12(0) = 0.11 * H
arrayOfVariantOfDouble12(1) = 0#
arrayOfVariantOfDouble12(2) = 0#
arrayOfVariantOfDouble12(3) = 0#
arrayOfVariantOfDouble12(4) = 1#
arrayOfVariantOfDouble12(5) = 0#
arrayOfVariantOfDouble12(6) = 0#
arrayOfVariantOfDouble12(7) = 0#
arrayOfVariantOfDouble12(8) = 1#
Set sketch12Variant = sketch12
sketch12Variant.SetAbsoluteAxisData arrayOfVariantOfDouble12
part4.InWorkObject = sketch12

Dim factory2D12 As Factory2D
Set factory2D12 = sketch12.OpenEdition()
Dim geometricElements12 As GeometricElements
Set geometricElements12 = sketch12.GeometricElements
Dim axis2D12 As Axis2D
Set axis2D12 = geometricElements12.Item("AbsoluteAxis")
Dim line2D24 As Line2D
Set line2D24 = axis2D12.GetItem("HDirection")
line2D24.ReportName = 1
Dim line2D25 As Line2D
Set line2D25 = axis2D12.GetItem("VDirection")
line2D25.ReportName = 2

Dim point2D15 As Point2D
Set point2D15 = factory2D12.CreatePoint(0#, (D - D1) / 4 + D1 / 2)
point2D15.ReportName = 3

Dim circle2D13 As Circle2D
Set circle2D13 = factory2D12.CreateClosedCircle(0#, (D - D1) / 4 + D1 / 2, 0.11 * H)
circle2D13.CenterPoint = point2D15
circle2D13.ReportName = 4

Dim constraints12 As Constraints
Set constraints12 = sketch12.Constraints
Dim reference52 As Reference
Set reference52 = part4.CreateReferenceFromObject(point2D15)
Dim reference53 As Reference
Set reference53 = part4.CreateReferenceFromObject(line2D25)
Dim constraint18 As Constraint
Set constraint18 = constraints12.AddBiEltCst(catCstTypeOn, reference52, reference53)
constraint18.Mode = catCstModeDrivingDimension

Dim reference54 As Reference
Set reference54 = part4.CreateReferenceFromObject(circle2D13)
Dim constraint19 As Constraint
Set constraint19 = constraints12.AddMonoEltCst(catCstTypeRadius, reference54)
constraint19.Mode = catCstModeDrivingDimension
Dim length20 As Length
Set length20 = constraint19.Dimension
length20.Value = 0.22 * H

Dim reference55 As Reference
Set reference55 = part4.CreateReferenceFromObject(point2D15)
Dim reference56 As Reference
Set reference56 = part4.CreateReferenceFromObject(line2D24)
Dim constraint20 As Constraint
Set constraint20 = constraints12.AddBiEltCst(catCstTypeDistance, reference55, reference56)
constraint20.Mode = catCstModeDrivingDimension

Dim length21 As Length
Set length21 = constraint20.Dimension
length21.Value = (D - D1) / 4 + D1 / 2

sketch12.CloseEdition
part4.InWorkObject = sketch12
part4.Update
```

```
Dim pocket4 As Pocket
Set pocket4 = shapeFactory4.AddNewPocket(sketch12, 0.22 * H)
Dim limit7 As Limit
Set limit7 = pocket4.FirstLimit
limit7.LimitMode = catUpToLastLimit
'part4.Update

'PATRÓN CIRCULAR
Dim reference57 As Reference
Set reference57 = part4.CreateReferenceFromName("")
Dim reference58 As Reference
Set reference58 = part4.CreateReferenceFromName("")

Dim circPattern1 As CircPattern
Set circPattern1 = shapeFactory4.AddNewCircPattern(pocket4, 1, 2, 20#, 45#, 1, 1, reference57, reference58, True, 0#, True)
circPattern1.CircularPatternParameters = catInstancesandAngularSpacing

Dim angularRepartition1 As angularRepartition
Set angularRepartition1 = circPattern1.angularRepartition

'calculo del número de agujeros para ello definimos primero el número pi
Const PI = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es (D+D1)/4
Dim longitud As Double
longitud = 2 * PI * ((D - D1) / 4 + D1 / 2)
'crearemos una variable auxiliar que usara el comando matemático Round para redondear el numero de huecos que tendrá el separador.
'para calcular este número hacemos la hipótesis de que la distancia entre BOLAS será del radio de estas
Dim variableaux As Double
variableaux = Math.Round(longitud / (0.44 * H + 0.22 * H))
'ya tenemos el numero de huecos del separador
Dim intParam As intParam
Set intParam = angularRepartition1.InstancesCount
intParam.Value = variableaux
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angularRepartition2 As angularRepartition
Set angularRepartition2 = circPattern1.angularRepartition
Dim angle As angle
Set angle = angularRepartition2.AngularSpacing
angle.Value = 360 / variableaux

Dim refrot7 As Reference
Set refrot7 = part4.CreateReferenceFromObject(reference48)
circPattern1.SetRotationAxis refrot7

part4.UpdateObject circPattern1
part4.Update

Dim product4 As Product
Set product4 = partDocument4.GetItem("Part4")
product4.PartNumber = "jaula"

partDocument4.SaveAs objPath & "\jaula.CATPart"
partDocument4.Close
'======================================
'----------------------------ENSAMBLAJE----------------
'======================================
Dim documents5 As Documents
Set documents5 = CATIA.Documents
Dim productDocument1 As ProductDocument
Set productDocument1 = documents5.Add("Product")
Dim product5 As Product
Set product5 = productDocument1.Product
Dim products1 As Products
Set products1 = product5.Products

Dim arrayOfVariantOfBSTR1(0)
arrayOfVariantOfBSTR1(0) = objPath & "\aro1.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR1, "All"

Dim constraints13 As Constraints
Set constraints13 = product5.Connections("CATIAConstraints")
'fijamos el aro1
Dim reference59 As Reference
Set reference59 = product5.CreateReferenceFromName("Product5/aro1.1/!Product5/aro1.1/")
Dim constraint21 As Constraint
Set constraint21 = constraints13.AddMonoEltCst(catCstTypeReference, reference59)

Dim arrayOfVariantOfBSTR2(0)
arrayOfVariantOfBSTR2(0) = objPath & "\aro2.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR2, "All"

'coincidencia de ejes de los aros
Dim reference60 As Reference
Set reference60 = product5.CreateReferenceFromName("Product5/aro2.1/!Axis:(Selection_RSur:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());Slot.1_ResultOUT;Z0;G3055))")
Dim reference61 As Reference
Set reference61 = product5.CreateReferenceFromName("Product5/aro1.1/!Axis:(Selection_RSur:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());Slot.1_ResultOUT;Z0;G3055))")
Dim constraint22 As Constraint
Set constraint222 = constraints13.AddBiEltCst(catCstTypeOn, reference60, reference61)

Dim arrayOfVariantOfBSTR3(0)
arrayOfVariantOfBSTR3(0) = objPath & "\bolas.CATPart"
Set products1Variant = products1
```

```
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR3, "All"

Dim arrayOfVariantOfBSTR4(0)
arrayOfVariantOfBSTR4(0) = objPath &
"\jaula.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR4, "All"

'coincidencia de ejes entre el aro1 y la jaula
Dim reference62 As Reference
Set reference62 =
product5.CreateReferenceFromName("Product1/jaula
.1/!Axis:(Selection_RSur:(Face:(Brp:(Pad.1;0:(Brp:(
Sketch.1;3)));None:();Cf11:());CircPattern.1_ResultO
UT;Z0;G3055))")
Dim constraint23 As Constraint
Set constraint23 =
constraints13.AddBiEltCst(catCstTypeOn,
reference61, reference62)

'distancia entre el aro1 y aro 2
Dim reference63 As Reference
Set reference63 =
product5.CreateReferenceFromName("Product5/aro1
.1/!Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf1
1:());Slot.1_ResultOUT;Z0;G3055)")
Dim reference64 As Reference
Set reference64 =
product5.CreateReferenceFromName("Product5/aro2
.1/!Selection_RSur:(Face:(Brp:(Pad.1;1);None:();Cf1
1:());Slot.1_ResultOUT;Z0;G3055)")
Dim constraint24 As Constraint
Set constraint24 =
constraints13.AddBiEltCst(catCstTypeDistance,
reference63, reference64)
Dim length22 As Length
Set length22 = constraint24.Dimension
length22.Value = -H
constraint24.Orientation = catCstOrientOpposite

'distancia entre el aro1 y la jaula
Dim reference65 As Reference
Set reference65 =
product5.CreateReferenceFromName("Product5/jaula
.1/!yz plane")
Dim constraint25 As Constraint
Set constraint25 =
constraints13.AddBiEltCst(catCstTypeDistance,
reference63, reference65)
Dim length23 As Length
Set length23 = constraint25.Dimension
length23.Value = -H / 2
constraint25.Orientation = catCstOrientSame

'Fijamos las bolas a los huecos de la jaula.
Dim referencec1 As Reference
Set referencec1 =
product5.CreateReferenceFromName("Product5/bola
s.1/!zx plane")
Dim referencec2 As Reference
Set referencec2 =
product5.CreateReferenceFromName("Product5/jaula
.1/!zx plane")
Dim constraintc1 As Constraint
Set constraintc1 =
constraints13.AddBiEltCst(catCstTypeDistance,
referencec1, referencec2)

Dim referencec3 As Reference
Set referencec3 =
product5.CreateReferenceFromName("Product5/bola
s.1/!yz plane")
Dim referencec4 As Reference
Set referencec4 =
product5.CreateReferenceFromName("Product5/jaula
.1/!yz plane")
Dim constraintc2 As Constraint
Set constraintc2 =
constraints13.AddBiEltCst(catCstTypeDistance,
referencec3, referencec4)

Dim referencec5 As Reference
Set referencec5 =
product5.CreateReferenceFromName("Product5/bola
s.1/!xy plane")
Dim referencec6 As Reference
Set referencec6 =
product5.CreateReferenceFromName("Product5/jaula
.1/!xy plane")
Dim constraintc3 As Constraint
Set constraintc3 =
constraints13.AddBiEltCst(catCstTypeDistance,
referencec5, referencec6)

product5.Update
```

- **Rodamiento de rodillos cilíndricos tipo NUP.**

```
Private Sub CommandButton1_Click()
'=====================================
'----------------------------------Anillo externo----------
'=====================================
Dim documents1 As Documents
Set documents1 = CATIA.Documents
Dim partDocument1 As PartDocument
Set partDocument1 = documents1.Add("Part")
Dim part1 As Part
Set part1 = partDocument1.Part
Dim bodies1 As Bodies
Set bodies1 = part1.Bodies
Dim body1 As Body
Set body1 = bodies1.Item("PartBody")
Dim sketches1 As Sketches
Set sketches1 = body1.Sketches
Dim originElements1 As OriginElements
Set originElements1 = part1.OriginElements
Dim reference1 As Reference
Set reference1 = originElements1.PlaneYZ
Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0#
arrayOfVariantOfDouble1(1) = 0#
arrayOfVariantOfDouble1(2) = 0#
arrayOfVariantOfDouble1(3) = 0#
arrayOfVariantOfDouble1(4) = 1#
arrayOfVariantOfDouble1(5) = 0#
arrayOfVariantOfDouble1(6) = 0#
arrayOfVariantOfDouble1(7) = 0#
```

```
arrayOfVariantOfDouble1(8) = 1#
Set sketch1Variant = sketch1
sketch1Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble1
part1.InWorkObject = sketch1

Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()
Dim geometricElements1 As GeometricElements
Set geometricElements1 =
sketch1.GeometricElements
Dim axis2D1 As Axis2D
Set axis2D1 =
geometricElements1.Item("AbsoluteAxis")
Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2

Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateClosedCircle(0#,
0#, D / 2)
Dim point2D1 As Point2D
Set point2D1 = axis2D1.GetItem("Origin")
circle2D1.CenterPoint = point2D1
circle2D1.ReportName = 3

Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints
Dim reference2 As Reference
Set reference2 =
part1.CreateReferenceFromObject(circle2D1)
Dim constraint1 As Constraint
Set constraint1 =
constraints1.AddMonoEltCst(catCstTypeRadius,
reference2)
constraint1.Mode = catCstModeDrivingDimension
Dim length1 As Length
Set length1 = constraint1.Dimension
length1.Value = D / 2

sketch1.CloseEdition
part1.InWorkObject = sketch1
part1.Update

'PAD
Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory
Dim pad1 As Pad
Set pad1 = shapeFactory1.AddNewPad(sketch1, B /
2)
pad1.IsSymmetric = True
Dim limit1 As Limit
Set limit1 = pad1.FirstLimit
Dim length2 As Length
Set length2 = limit1.Dimension
length2.Value = B / 2
part1.Update

'CHAFLÁN
Dim reference3 As Reference
Set reference3 =
part1.CreateReferenceFromName("")
Dim constRadEdgeFillet1 As ConstRadEdgeFillet
Set constRadEdgeFillet1 =
shapeFactory1.AddNewSolidEdgeFilletWithConstant
Radius(reference3,
catTangencyFilletEdgePropagation, r1)

Dim reference4 As Reference
Set reference4 =
part1.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf
11:());Face:(Brp:(Pad.1;2);None:();Cf11:());None:(Li
mits1:();Limits2:());Cf11:());WithTemporaryBody;W
ithoutBuildError;WithSelectingFeatureSupport;MFB
RepVersion_CXR15)", pad1)
constRadEdgeFillet1.AddObjectToFillet reference4
constRadEdgeFillet1.EdgePropagation =
catTangencyFilletEdgePropagation
Dim reference5 As Reference
Set reference5 =
part1.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;1);None:();Cf11:());Face:(Brp:(P
ad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());None:(Li
mits1:();Limits2:());Cf11:());WithTemporaryBody;W
ithoutBuildError;WithSelectingFeatureSupport;MFB
RepVersion_CXR15)", pad1)
constRadEdgeFillet1.AddObjectToFillet reference5
constRadEdgeFillet1.EdgePropagation =
catTangencyFilletEdgePropagation
part1.Update

'POCKET
Dim reference6 As Reference
Set reference6 =
part1.CreateReferenceFromName("Selection_RSur:(
Face:(Brp:(Pad.1;2);None:();Cf11:());EdgeFillet.1_R
esultOUT;Z0;G3055)")

Dim sketch2 As Sketch
Set sketch2 = sketches1.Add(reference6)

Dim arrayOfVariantOfDouble2(8)
arrayOfVariantOfDouble2(0) = B / 2
arrayOfVariantOfDouble2(1) = 0#
arrayOfVariantOfDouble2(2) = 0#
arrayOfVariantOfDouble2(3) = 0#
arrayOfVariantOfDouble2(4) = 1#
arrayOfVariantOfDouble2(5) = 0#
arrayOfVariantOfDouble2(6) = 0#
arrayOfVariantOfDouble2(7) = 0#
arrayOfVariantOfDouble2(8) = 1#
Set sketch2Variant = sketch2
sketch2Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble2
part1.InWorkObject = sketch2

Dim factory2D2 As Factory2D
Set factory2D2 = sketch2.OpenEdition()
Dim geometricElements2 As GeometricElements
Set geometricElements2 =
sketch2.GeometricElements
Dim axis2D2 As Axis2D
Set axis2D2 =
geometricElements2.Item("AbsoluteAxis")
Dim line2D3 As Line2D
Set line2D3 = axis2D2.GetItem("HDirection")
line2D3.ReportName = 1
Dim line2D4 As Line2D
Set line2D4 = axis2D2.GetItem("VDirection")
```

```
line2D4.ReportName = 2

Dim circle2D2 As Circle2D
Set circle2D2 = factory2D2.CreateClosedCircle(0#, 0#, D1 / 2)
Dim point2D2 As Point2D
Set point2D2 = axis2D2.GetItem("Origin")
circle2D2.CenterPoint = point2D2
circle2D2.ReportName = 3

Dim constraints2 As Constraints
Set constraints2 = sketch2.Constraints
Dim reference7 As Reference
Set reference7 = part1.CreateReferenceFromObject(circle2D2)
Dim constraint2 As Constraint
Set constraint2 = constraints2.AddMonoEltCst(catCstTypeRadius, reference7)
constraint2.Mode = catCstModeDrivingDimension
Dim length4 As Length
Set length4 = constraint2.Dimension
length4.Value = D1 / 2

sketch2.CloseEdition
part1.InWorkObject = sketch2
part1.Update

Dim pocket1 As Pocket
Set pocket1 = shapeFactory1.AddNewPocket(sketch2, B / 2)
Dim limit2 As Limit
Set limit2 = pocket1.FirstLimit
limit2.LimitMode = catUpToLastLimit

part1.Update

'SLOT para el canal de los rodillos
Dim reference8 As Reference
Set reference8 = originElements1.PlaneZX
Dim sketch3 As Sketch
Set sketch3 = sketches1.Add(reference8)

Dim arrayOfVariantOfDouble3(8)
arrayOfVariantOfDouble3(0) = 0#
arrayOfVariantOfDouble3(1) = 0#
arrayOfVariantOfDouble3(2) = 0#
arrayOfVariantOfDouble3(3) = -1#
arrayOfVariantOfDouble3(4) = 0#
arrayOfVariantOfDouble3(5) = 0#
arrayOfVariantOfDouble3(6) = 0#
arrayOfVariantOfDouble3(7) = -0#
arrayOfVariantOfDouble3(8) = 1#
Set sketch3Variant = sketch3
sketch3Variant.SetAbsoluteAxisData arrayOfVariantOfDouble3
part1.InWorkObject = sketch3

Dim factory2D3 As Factory2D
Set factory2D3 = sketch3.OpenEdition()
Dim geometricElements3 As GeometricElements
Set geometricElements3 = sketch3.GeometricElements
Dim axis2D3 As Axis2D
Set axis2D3 = geometricElements3.Item("AbsoluteAxis")
Dim line2D5 As Line2D
Set line2D5 = axis2D3.GetItem("HDirection")
line2D5.ReportName = 1
Dim line2D6 As Line2D
Set line2D6 = axis2D3.GetItem("VDirection")
line2D6.ReportName = 2

'dibujamos un rectangulo
Dim point2D3 As Point2D
Set point2D3 = factory2D3.CreatePoint(-B / 4, F / 2)
point2D3.ReportName = 3

Dim point2D4 As Point2D
Set point2D4 = factory2D3.CreatePoint(B / 4, F / 2)
point2D4.ReportName = 4

Dim line2D7 As Line2D
Set line2D7 = factory2D3.CreateLine(-B / 4, F / 2, B / 4, F / 2)
line2D7.ReportName = 5
line2D7.StartPoint = point2D3
line2D7.EndPoint = point2D4

Dim constraints3 As Constraints
Set constraints3 = sketch3.Constraints

Dim reference9 As Reference
Set reference9 = part1.CreateReferenceFromObject(line2D7)
Dim reference10 As Reference
Set reference10 = part1.CreateReferenceFromObject(line2D5)
Dim constraint3 As Constraint
Set constraint3 = constraints3.AddBiEltCst(catCstTypeHorizontality, reference9, reference10)
constraint3.Mode = catCstModeDrivingDimension

Dim reference11 As Reference
Set reference11 = part1.CreateReferenceFromObject(line2D7)
Dim constraint4 As Constraint
Set constraint4 = constraints3.AddMonoEltCst(catCstTypeLength, reference11)
constraint4.Mode = catCstModeDrivingDimension
Dim length5 As Length
Set length5 = constraint4.Dimension
length5.Value = B / 2

Dim point2D5 As Point2D
Set point2D5 = factory2D3.CreatePoint(B / 4, E / 2)
point2D5.ReportName = 6

Dim line2D8 As Line2D
Set line2D8 = factory2D3.CreateLine(B / 4, F / 2, B / 4, E / 2)
line2D8.ReportName = 7
line2D8.StartPoint = point2D4
line2D8.EndPoint = point2D5

Dim reference12 As Reference
Set reference12 = part1.CreateReferenceFromObject(line2D8)
Dim reference13 As Reference
Set reference13 = part1.CreateReferenceFromObject(line2D6)
Dim constraint5 As Constraint
```

```
Set constraint5 = constraints3.AddBiEltCst(catCstTypeVerticality, reference12, reference13)
constraint5.Mode = catCstModeDrivingDimension

Dim reference14 As Reference
Set reference14 = part1.CreateReferenceFromObject(line2D8)
Dim constraint6 As Constraint
Set constraint6 = constraints3.AddMonoEltCst(catCstTypeLength, reference14)
constraint6.Mode = catCstModeDrivingDimension
Dim length6 As Length
Set length6 = constraint6.Dimension
length6.Value = (E - F) / 2

Dim point2D6 As Point2D
Set point2D6 = factory2D3.CreatePoint(-B / 4, E / 2)
point2D6.ReportName = 8

Dim line2D9 As Line2D
Set line2D9 = factory2D3.CreateLine(B / 4, E / 2, -B / 4, E / 2)
line2D9.ReportName = 9
line2D9.StartPoint = point2D5
line2D9.EndPoint = point2D6

Dim reference15 As Reference
Set reference15 = part1.CreateReferenceFromObject(line2D9)
Dim reference16 As Reference
Set reference16 = part1.CreateReferenceFromObject(line2D5)
Dim constraint7 As Constraint
Set constraint7 = constraints3.AddBiEltCst(catCstTypeHorizontality, reference15, reference16)
constraint7.Mode = catCstModeDrivingDimension

Dim line2D10 As Line2D
Set line2D10 = factory2D3.CreateLine(-B / 4, E / 2, -B / 4, F / 2)
line2D10.ReportName = 10
line2D10.StartPoint = point2D6
line2D10.EndPoint = point2D3

Dim reference17 As Reference
Set reference17 = part1.CreateReferenceFromObject(line2D10)
Dim reference18 As Reference
Set reference18 = part1.CreateReferenceFromObject(line2D6)
Dim constraint8 As Constraint
Set constraint8 = constraints3.AddBiEltCst(catCstTypeVerticality, reference17, reference18)
constraint8.Mode = catCstModeDrivingDimension

Dim reference19 As Reference
Set reference19 = part1.CreateReferenceFromObject(line2D7)
Dim reference20 As Reference
Set reference20 = part1.CreateReferenceFromObject(line2D5)
Dim constraint9 As Constraint
Set constraint9 = constraints3.AddBiEltCst(catCstTypeDistance, reference19, reference20)
constraint9.Mode = catCstModeDrivingDimension
Dim length7 As Length
Set length7 = constraint9.Dimension
length7.Value = F / 2

Dim reference21 As Reference
Set reference21 = part1.CreateReferenceFromObject(line2D8)
Dim reference22 As Reference
Set reference22 = part1.CreateReferenceFromObject(line2D6)
Dim constraint10 As Constraint
Set constraint10 = constraints3.AddBiEltCst(catCstTypeDistance, reference21, reference22)
constraint10.Mode = catCstModeDrivingDimension
Dim length8 As Length
Set length8 = constraint10.Dimension
length8.Value = B / 4

sketch3.CloseEdition
part1.InWorkObject = sketch3
part1.Update

Dim sketch4 As Sketch
Set sketch4 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble4(8)
arrayOfVariantOfDouble4(0) = 0#
arrayOfVariantOfDouble4(1) = 0#
arrayOfVariantOfDouble4(2) = 0#
arrayOfVariantOfDouble4(3) = 0#
arrayOfVariantOfDouble4(4) = 1#
arrayOfVariantOfDouble4(5) = 0#
arrayOfVariantOfDouble4(6) = 0#
arrayOfVariantOfDouble4(7) = 0#
arrayOfVariantOfDouble4(8) = 1#
Set sketch4Variant = sketch4
sketch4Variant.SetAbsoluteAxisData arrayOfVariantOfDouble4
part1.InWorkObject = sketch4

Dim factory2D4 As Factory2D
Set factory2D4 = sketch4.OpenEdition()
Dim geometricElements4 As GeometricElements
Set geometricElements4 = sketch4.GeometricElements
Dim axis2D4 As Axis2D
Set axis2D4 = geometricElements4.Item("AbsoluteAxis")
Dim line2D11 As Line2D
Set line2D11 = axis2D4.GetItem("HDirection")
line2D11.ReportName = 1
Dim line2D12 As Line2D
Set line2D12 = axis2D4.GetItem("VDirection")
line2D12.ReportName = 2

Dim circle2D3 As Circle2D
Set circle2D3 = factory2D4.CreateClosedCircle(0#, 0#, F / 2 + (E - F) / 4)
Dim point2D7 As Point2D
Set point2D7 = axis2D4.GetItem("Origin")
circle2D3.CenterPoint = point2D7
circle2D3.ReportName = 3
```

```
Dim constraints4 As Constraints
Set constraints4 = sketch4.Constraints

Dim reference23 As Reference
Set reference23 = part1.CreateReferenceFromObject(circle2D3)
Dim constraint11 As Constraint
Set constraint11 = constraints4.AddMonoEltCst(catCstTypeRadius, reference23)
constraint11.Mode = catCstModeDrivingDimension
Dim length9 As Length
Set length9 = constraint11.Dimension
length9.Value = F / 2 + (E - F) / 4

sketch4.CloseEdition
part1.InWorkObject = sketch4
part1.Update

Dim slot1 As Slot
Set slot1 = shapeFactory1.AddNewSlotFromRef(Nothing, Nothing)

Dim reference24 As Reference
Set reference24 = part1.CreateReferenceFromObject(sketch3)
slot1.SetProfileElement reference24
Dim reference25 As Reference
Set reference25 = part1.CreateReferenceFromObject(sketch4)
slot1.CenterCurveElement = reference25

part1.Update

Dim product1 As Product
Set product1 = partDocument1.GetItem("Part1")
product1.PartNumber = "aroexterno"
partDocument1.SaveAs objPath & "\aroext.catpart"
partDocument1.Close

'====================================
'----------------------anillo interno----------------------
'====================================
Dim documents2 As Documents
Set documents2 = CATIA.Documents
Dim partDocument2 As PartDocument
Set partDocument2 = documents2.Add("Part")
Dim part2 As Part
Set part2 = partDocument2.Part
Dim bodies2 As Bodies
Set bodies2 = part2.Bodies
Dim body2 As Body
Set body2 = bodies2.Item("PartBody")
Dim sketches2 As Sketches
Set sketches2 = body2.Sketches
Dim originElements2 As OriginElements
Set originElements2 = part2.OriginElements
Dim reference26 As Reference
Set reference26 = originElements2.PlaneYZ
Dim sketch5 As Sketch
Set sketch5 = sketches2.Add(reference26)

Dim arrayOfVariantOfDouble5(8)
arrayOfVariantOfDouble5(0) = 0#
arrayOfVariantOfDouble5(1) = 0#
arrayOfVariantOfDouble5(2) = 0#
arrayOfVariantOfDouble5(3) = 0#
arrayOfVariantOfDouble5(4) = 1#
arrayOfVariantOfDouble5(5) = 0#
arrayOfVariantOfDouble5(6) = 0#
arrayOfVariantOfDouble5(7) = 0#
arrayOfVariantOfDouble5(8) = 1#
Set sketch5Variant = sketch5
sketch5Variant.SetAbsoluteAxisData arrayOfVariantOfDouble5

part2.InWorkObject = sketch5

'PAD
Dim factory2D5 As Factory2D
Set factory2D5 = sketch5.OpenEdition()
Dim geometricElements5 As GeometricElements
Set geometricElements5 = sketch5.GeometricElements

Dim axis2D5 As Axis2D
Set axis2D5 = geometricElements5.Item("AbsoluteAxis")
Dim line2D13 As Line2D
Set line2D13 = axis2D5.GetItem("HDirection")
line2D13.ReportName = 1
Dim line2D14 As Line2D
Set line2D14 = axis2D5.GetItem("VDirection")
line2D14.ReportName = 2

Dim circle2D4 As Circle2D
Set circle2D4 = factory2D5.CreateClosedCircle(0#, 0#, d_1 / 2)
Dim point2D8 As Point2D
Set point2D8 = axis2D5.GetItem("Origin")
circle2D4.CenterPoint = point2D8
circle2D4.ReportName = 3

Dim constraints5 As Constraints
Set constraints5 = sketch5.Constraints

Dim reference27 As Reference
Set reference27 = part2.CreateReferenceFromObject(circle2D4)
Dim constraint12 As Constraint
Set constraint12 = constraints5.AddMonoEltCst(catCstTypeRadius, reference27)
constraint12.Mode = catCstModeDrivingDimension

Dim length10 As Length
Set length10 = constraint12.Dimension
length10.Value = d_1 / 2

sketch5.CloseEdition
part2.InWorkObject = sketch5
part2.Update

Dim shapeFactory2 As ShapeFactory
Set shapeFactory2 = part2.ShapeFactory
Dim pad2 As Pad
Set pad2 = shapeFactory2.AddNewPad(sketch5, B / 2)
Dim limit3 As Limit
Set limit3 = pad2.FirstLimit
Dim length11 As Length
```

```
Set length11 = limit3.Dimension
length11.Value = B / 2
pad2.IsSymmetric = True

part2.Update

'POCKET
Dim reference28 As Reference
Set reference28 = part2.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")

Dim sketch6 As Sketch
Set sketch6 = sketches2.Add(reference28)

Dim arrayOfVariantOfDouble6(8)
arrayOfVariantOfDouble6(0) = B / 2
arrayOfVariantOfDouble6(1) = 0#
arrayOfVariantOfDouble6(2) = 0#
arrayOfVariantOfDouble6(3) = 0#
arrayOfVariantOfDouble6(4) = 1#
arrayOfVariantOfDouble6(5) = 0#
arrayOfVariantOfDouble6(6) = 0#
arrayOfVariantOfDouble6(7) = 0#
arrayOfVariantOfDouble6(8) = 1#
Set sketch6Variant = sketch6
sketch6Variant.SetAbsoluteAxisData arrayOfVariantOfDouble6
part2.InWorkObject = sketch6

Dim factory2D6 As Factory2D
Set factory2D6 = sketch6.OpenEdition()
Dim geometricElements6 As GeometricElements
Set geometricElements6 = sketch6.GeometricElements

Dim axis2D6 As Axis2D
Set axis2D6 = geometricElements6.Item("AbsoluteAxis")
Dim line2D15 As Line2D
Set line2D15 = axis2D6.GetItem("HDirection")
line2D15.ReportName = 1
Dim line2D16 As Line2D
Set line2D16 = axis2D6.GetItem("VDirection")
line2D16.ReportName = 2

Dim circle2D5 As Circle2D
Set circle2D5 = factory2D6.CreateClosedCircle(0#, 0#, d_ / 2)
Dim point2D9 As Point2D
Set point2D9 = axis2D6.GetItem("Origin")
circle2D5.CenterPoint = point2D9
circle2D5.ReportName = 3

Dim constraints6 As Constraints
Set constraints6 = sketch6.Constraints
Dim reference29 As Reference
Set reference29 = part2.CreateReferenceFromObject(circle2D5)
Dim constraint13 As Constraint
Set constraint13 = constraints6.AddMonoEltCst(catCstTypeRadius, reference29)
constraint13.Mode = catCstModeDrivingDimension
Dim length12 As Length
Set length12 = constraint13.Dimension
length12.Value = d_ / 2

sketch6.CloseEdition
part2.InWorkObject = sketch6
part2.Update

Dim pocket2 As Pocket
Set pocket2 = shapeFactory2.AddNewPocket(sketch6, B / 2)
Dim limit4 As Limit
Set limit4 = pocket2.FirstLimit
limit4.LimitMode = catUpThruNextLimit
part2.Update

'Chaflán
Dim reference30 As Reference
Set reference30 = part2.CreateReferenceFromName("")
Dim constRadEdgeFillet2 As ConstRadEdgeFillet
Set constRadEdgeFillet2 = shapeFactory2.AddNewSolidEdgeFilletWithConstantRadius(reference30, catTangencyFilletEdgePropagation, r1)

Dim reference31 As Reference
Set reference31 = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;2);None:();Cf11:());Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket1)
constRadEdgeFillet2.AddObjectToFillet reference31
constRadEdgeFillet2.EdgePropagation = catTangencyFilletEdgePropagation
Dim reference32 As Reference
Set reference32 = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());Face:(Brp:(Pad.1;1);None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket1)
constRadEdgeFillet2.AddObjectToFillet reference32
constRadEdgeFillet2.EdgePropagation = catTangencyFilletEdgePropagation

part2.Update

'SLOT
Dim reference33 As Reference
Set reference33 = originElements2.PlaneZX
Dim sketch7 As Sketch
Set sketch7 = sketches2.Add(reference33)

Dim arrayOfVariantOfDouble7(8)
arrayOfVariantOfDouble7(0) = 0#
arrayOfVariantOfDouble7(1) = 0#
arrayOfVariantOfDouble7(2) = 0#
arrayOfVariantOfDouble7(3) = -1#
arrayOfVariantOfDouble7(4) = 0#
arrayOfVariantOfDouble7(5) = 0#
arrayOfVariantOfDouble7(6) = 0#
arrayOfVariantOfDouble7(7) = -0#
arrayOfVariantOfDouble7(8) = 1#
Set sketch7Variant = sketch7
```

```
sketch7Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble7
part2.InWorkObject = sketch7

Dim factory2D7 As Factory2D
Set factory2D7 = sketch7.OpenEdition()
Dim geometricElements7 As GeometricElements
Set geometricElements7 =
sketch7.GeometricElements
Dim axis2D7 As Axis2D
Set axis2D7 =
geometricElements7.Item("AbsoluteAxis")
Dim line2D17 As Line2D
Set line2D17 = axis2D7.GetItem("HDirection")
line2D17.ReportName = 1
Dim line2D18 As Line2D
Set line2D18 = axis2D7.GetItem("VDirection")
line2D18.ReportName = 2

'dibujamos un rectangulo
Dim point2D10 As Point2D
Set point2D10 = factory2D7.CreatePoint(-B / 4, F /
2)
point2D10.ReportName = 3
Dim point2D11 As Point2D
Set point2D11 = factory2D7.CreatePoint(B / 4, F / 2)
point2D11.ReportName = 4
Dim line2D19 As Line2D
Set line2D19 = factory2D7.CreateLine(-B / 4, F / 2,
B / 4, F / 2)
line2D19.ReportName = 5
line2D19.StartPoint = point2D10
line2D19.EndPoint = point2D11

Dim constraints7 As Constraints
Set constraints7 = sketch7.Constraints
Dim reference34 As Reference
Set reference34 =
part2.CreateReferenceFromObject(line2D19)
Dim reference35 As Reference
Set reference35 =
part2.CreateReferenceFromObject(line2D17)
Dim constraint14 As Constraint
Set constraint14 =
constraints7.AddBiEltCst(catCstTypeHorizontality,
reference34, reference35)
constraint14.Mode = catCstModeDrivingDimension

Dim reference36 As Reference
Set reference36 =
part2.CreateReferenceFromObject(line2D19)
Dim constraint15 As Constraint
Set constraint15 =
constraints7.AddMonoEltCst(catCstTypeLength,
reference36)
constraint15.Mode = catCstModeDrivingDimension
Dim length13 As Length
Set length13 = constraint15.Dimension
length13.Value = B / 2

Dim point2D12 As Point2D
Set point2D12 = factory2D7.CreatePoint(B / 4, E / 2)
point2D12.ReportName = 6
Dim line2D20 As Line2D
Set line2D20 = factory2D7.CreateLine(B / 4, F / 2, B
/ 4, E / 2)
line2D20.ReportName = 7
line2D20.StartPoint = point2D11
line2D20.EndPoint = point2D12

Dim reference37 As Reference
Set reference37 =
part2.CreateReferenceFromObject(line2D20)
Dim reference38 As Reference
Set reference38 =
part2.CreateReferenceFromObject(line2D18)
Dim constraint16 As Constraint
Set constraint16 =
constraints7.AddBiEltCst(catCstTypeVerticality,
reference37, reference38)
constraint16.Mode = catCstModeDrivingDimension

Dim reference39 As Reference
Set reference39 =
part2.CreateReferenceFromObject(line2D20)
Dim constraint17 As Constraint
Set constraint17 =
constraints7.AddMonoEltCst(catCstTypeLength,
reference39)
constraint17.Mode = catCstModeDrivingDimension
Dim length14 As Length
Set length14 = constraint17.Dimension
length14.Value = (E - F) / 2

Dim point2D13 As Point2D
Set point2D13 = factory2D7.CreatePoint(-B / 4, E /
2)
point2D13.ReportName = 8
Dim line2D21 As Line2D
Set line2D21 = factory2D7.CreateLine(B / 4, E / 2, -
B / 4, E / 2)
line2D21.ReportName = 9
line2D21.StartPoint = point2D12
line2D21.EndPoint = point2D13

Dim reference40 As Reference
Set reference40 =
part2.CreateReferenceFromObject(line2D21)
Dim reference41 As Reference
Set reference41 =
part2.CreateReferenceFromObject(line2D13)
Dim constraint18 As Constraint
Set constraint18 =
constraints7.AddBiEltCst(catCstTypeHorizontality,
reference40, reference41)
constraint18.Mode = catCstModeDrivingDimension

Dim line2D22 As Line2D
Set line2D22 = factory2D7.CreateLine(-B / 4, E / 2, -
B / 4, F / 2)
line2D22.ReportName = 10
line2D22.StartPoint = point2D13
line2D22.EndPoint = point2D10

Dim reference42 As Reference
Set reference42 =
part2.CreateReferenceFromObject(line2D22)
Dim reference43 As Reference
Set reference43 =
part2.CreateReferenceFromObject(line2D18)
Dim constraint19 As Constraint
Set constraint19 =
constraints7.AddBiEltCst(catCstTypeVerticality,
reference42, reference43)
```

```
constraint19.Mode = catCstModeDrivingDimension

Dim reference44 As Reference
Set reference44 = part2.CreateReferenceFromObject(line2D22)
Dim reference45 As Reference
Set reference45 = part2.CreateReferenceFromObject(line2D18)
Dim constraint20 As Constraint
Set constraint20 = constraints7.AddBiEltCst(catCstTypeDistance, reference44, reference45)
constraint20.Mode = catCstModeDrivingDimension
Dim length15 As Length
Set length15 = constraint20.Dimension
length15.Value = B / 4

Dim reference46 As Reference
Set reference46 = part2.CreateReferenceFromObject(line2D19)
Dim reference47 As Reference
Set reference47 = part2.CreateReferenceFromObject(line2D17)
Dim constraint21 As Constraint
Set constraint21 = constraints7.AddBiEltCst(catCstTypeDistance, reference46, reference47)
constraint21.Mode = catCstModeDrivingDimension
Dim length16 As Length
Set length16 = constraint21.Dimension
length16.Value = F / 2

sketch7.CloseEdition
part2.InWorkObject = sketch7
part2.Update

Dim sketch8 As Sketch
Set sketch8 = sketches2.Add(reference26)

Dim arrayOfVariantOfDouble8(8)
arrayOfVariantOfDouble8(0) = 0#
arrayOfVariantOfDouble8(1) = 0#
arrayOfVariantOfDouble8(2) = 0#
arrayOfVariantOfDouble8(3) = 0#
arrayOfVariantOfDouble8(4) = 1#
arrayOfVariantOfDouble8(5) = 0#
arrayOfVariantOfDouble8(6) = 0#
arrayOfVariantOfDouble8(7) = 0#
arrayOfVariantOfDouble8(8) = 1#
Set sketch8Variant = sketch8
sketch8Variant.SetAbsoluteAxisData arrayOfVariantOfDouble8

part2.InWorkObject = sketch8

Dim factory2D8 As Factory2D
Set factory2D8 = sketch8.OpenEdition()
Dim geometricElements8 As GeometricElements
Set geometricElements8 = sketch8.GeometricElements
Dim axis2D8 As Axis2D
Set axis2D8 = geometricElements8.Item("AbsoluteAxis")
Dim line2D23 As Line2D
Set line2D23 = axis2D8.GetItem("HDirection")
line2D23.ReportName = 1
Dim line2D24 As Line2D
Set line2D24 = axis2D8.GetItem("VDirection")
line2D24.ReportName = 2

Dim circle2D6 As Circle2D
Set circle2D6 = factory2D8.CreateClosedCircle(0#, 0#, F / 2 + (E - F) / 4)
Dim point2D14 As Point2D
Set point2D14 = axis2D8.GetItem("Origin")
circle2D6.CenterPoint = point2D14
circle2D6.ReportName = 3

Dim constraints8 As Constraints
Set constraints8 = sketch8.Constraints

Dim reference48 As Reference
Set reference48 = part2.CreateReferenceFromObject(circle2D6)
Dim constraint22 As Constraint
Set constraint22 = constraints8.AddMonoEltCst(catCstTypeRadius, reference48)
constraint22.Mode = catCstModeDrivingDimension
Dim length17 As Length
Set length17 = constraint22.Dimension
length17.Value = F / 2 + (E - F) / 4

sketch8.CloseEdition
part2.InWorkObject = sketch8
part2.Update

Dim slot2 As Slot
Set slot2 = shapeFactory2.AddNewSlotFromRef(Nothing, Nothing)

Dim reference49 As Reference
Set reference49 = part2.CreateReferenceFromObject(sketch7)
slot2.SetProfileElement reference49
Dim reference50 As Reference
Set reference50 = part2.CreateReferenceFromObject(sketch8)
slot2.CenterCurveElement = reference50
part2.Update

Dim product2 As Product
Set product2 = partDocument2.GetItem("Part2")
product2.PartNumber = "arointerno"
partDocument2.SaveAs objPath & "\aroint.CATPart"
partDocument2.Close
'=====================================
'----------------------------------------Rodillos---------------
'=====================================

Dim documents3 As Documents
Set documents3 = CATIA.Documents
Dim partDocument3 As PartDocument
Set partDocument3 = documents3.Add("Part")
Dim part3 As Part
Set part3 = partDocument3.Part
Dim bodies3 As Bodies
Set bodies3 = part3.Bodies
Dim body3 As Body
Set body3 = bodies3.Item("PartBody")
Dim sketches3 As Sketches
Set sketches3 = body3.Sketches
Dim originElements3 As OriginElements
```

```
Set originElements3 = part3.OriginElements
Dim reference51 As Reference
Set reference51 = originElements3.PlaneYZ
Dim sketch9 As Sketch
Set sketch9 = sketches3.Add(reference51)

Dim arrayOfVariantOfDouble9(8)
arrayOfVariantOfDouble9(0) = 0#
arrayOfVariantOfDouble9(1) = 0#
arrayOfVariantOfDouble9(2) = 0#
arrayOfVariantOfDouble9(3) = 0#
arrayOfVariantOfDouble9(4) = 1#
arrayOfVariantOfDouble9(5) = 0#
arrayOfVariantOfDouble9(6) = 0#
arrayOfVariantOfDouble9(7) = 0#
arrayOfVariantOfDouble9(8) = 1#
Set sketch9Variant = sketch9
sketch9Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble9

part3.InWorkObject = sketch9

Dim factory2D9 As Factory2D
Set factory2D9 = sketch9.OpenEdition()
Dim geometricElements9 As GeometricElements
Set geometricElements9 =
sketch9.GeometricElements

Dim axis2D9 As Axis2D
Set axis2D9 =
geometricElements9.Item("AbsoluteAxis")
Dim line2D25 As Line2D
Set line2D25 = axis2D9.GetItem("HDirection")
line2D25.ReportName = 1
Dim line2D26 As Line2D
Set line2D26 = axis2D9.GetItem("VDirection")
line2D26.ReportName = 2

Dim point2D15 As Point2D
Set point2D15 = factory2D9.CreatePoint(0#, F / 2 +
(E - F) / 4)
point2D15.ReportName = 3
Dim circle2D7 As Circle2D
Set circle2D7 = factory2D9.CreateClosedCircle(0#, F
/ 2 + (E - F) / 4, (E - F) / 4)
circle2D7.CenterPoint = point2D15
circle2D7.ReportName = 4

Dim constraints9 As Constraints
Set constraints9 = sketch9.Constraints
Dim reference52 As Reference
Set reference52 =
part3.CreateReferenceFromObject(point2D15)
Dim reference53 As Reference
Set reference53 =
part3.CreateReferenceFromObject(line2D26)
Dim constraint23 As Constraint
Set constraint23 =
constraints9.AddBiEltCst(catCstTypeOn,
reference52, reference53)
constraint23.Mode = catCstModeDrivingDimension
Dim reference54 As Reference
Set reference54 =
part3.CreateReferenceFromObject(circle2D7)
Dim constraint24 As Constraint
Set constraint24 =
constraints9.AddMonoEltCst(catCstTypeRadius,
reference54)
constraint24.Mode = catCstModeDrivingDimension
Dim length18 As Length
Set length18 = constraint24.Dimension
length18.Value = (E - F) / 4

Dim reference55 As Reference
Set reference55 =
part3.CreateReferenceFromObject(line2D25)
Dim reference56 As Reference
Set reference56 =
part3.CreateReferenceFromObject(point2D15)
Dim constraint25 As Constraint
Set constraint25 =
constraints9.AddBiEltCst(catCstTypeDistance,
reference55, reference56)
constraint25.Mode = catCstModeDrivingDimension
Dim length19 As Length
Set length19 = constraint25.Dimension
length19.Value = F / 2 + (E - F) / 4

sketch9.CloseEdition
part3.InWorkObject = sketch9
part3.Update


Dim shapeFactory3 As ShapeFactory
Set shapeFactory3 = part3.ShapeFactory

Dim pad3 As Pad
Set pad3 = shapeFactory3.AddNewPad(sketch9, B /
4)
Dim limit5 As Limit
Set limit5 = pad3.FirstLimit
Dim length20 As Length
Set length20 = limit5.Dimension
length20.Value = B / 4
pad3.IsSymmetric = True

part3.Update

'CHAFLÁN
Dim reference3b As Reference
Set reference3b =
part3.CreateReferenceFromName("")
Dim constRadEdgeFillet3 As ConstRadEdgeFillet
Set constRadEdgeFillet3 =
shapeFactory3.AddNewSolidEdgeFilletWithConstant
Radius(reference3b,
catTangencyFilletEdgePropagation, r1)
Dim reference4b As Reference
Set reference4b =
part3.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;4)));None:();Cf
11:());Face:(Brp:(Pad.1;2);None:();Cf11:());None:(Li
mits1:();Limits2:());Cf11:());WithTemporaryBody;W
ithoutBuildError;WithSelectingFeatureSupport;MFB
RepVersion_CXR15)", pad1)
constRadEdgeFillet3.AddObjectToFillet reference4b
constRadEdgeFillet3.EdgePropagation =
catTangencyFilletEdgePropagation
Dim reference5b As Reference
Set reference5b =
part3.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;1);None:();Cf11:());Face:(Brp:(P
```

```
ad.1;0:(Brp:(Sketch.1;4)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pad1)
constRadEdgeFillet3.AddObjectToFillet reference5b
constRadEdgeFillet3.EdgePropagation = catTangencyFilletEdgePropagation
part3.Update

'Patrón Circular
Dim referenceb As Reference
Set referenceb = part3.CreateReferenceFromName("")
Dim referenceb1 As Reference
Set referenceb1 = part3.CreateReferenceFromName("")
Dim circPatternb As CircPattern
Set circPatternb = shapeFactory3.AddNewCircPattern(Nothing, 1, 2, 20#, 45#, 1, 1, referenceb, referenceb1, True, 0#, True)
circPatternb.CircularPatternParameters = catInstancesandAngularSpacing
Dim angularRepartitionb As angularRepartition
Set angularRepartitionb = circPatternb.angularRepartition
'cálculo del número de agujeros para ello definimos primero el número pi
Const PIb = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es F / 2 + (E - F) / 4
Dim longitudb As Double
longitudb = 2 * PIb * (F / 2 + (E - F) / 4)
'crearemos una variable auxiliar que usara el comando matemático Round para redondear el numero de huecos que tendrá el separador.
'para calcular este número hacemos la hipótesis de que la distancia entre rodillos será el diametro del rodillo
Dim variableauxb As Double
variableauxb = Math.Round(longitudb / (3 * (E - F) / 4))
'ya tenemos el numero de huecos del separador
Dim intParamb As intParam
Set intParamb = angularRepartitionb.InstancesCount
intParamb.Value = variableauxb
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angleb As angle
Set angleb = angularRepartitionb.AngularSpacing
angleb.Value = 360 / variableauxb
circPatternb.SetRotationAxis reference51

part3.UpdateObject circPatternb
part3.Update

Dim product3 As Product
Set product3 = partDocument3.GetItem("Part3")
product3.PartNumber = "rodillos"

partDocument3.SaveAs objPath & "\rodillos.CATPart"
partDocument3.Close

'=====================================
'----------Separador----------
'=====================================
Dim documents4 As Documents
Set documents4 = CATIA.Documents
Dim partDocument4 As PartDocument
Set partDocument4 = documents4.Add("Part")
Dim part4 As Part
Set part4 = partDocument4.Part
Dim bodies4 As Bodies
Set bodies4 = part4.Bodies
Dim body4 As Body
Set body4 = bodies4.Item("PartBody")
Dim sketches4 As Sketches
Set sketches4 = body4.Sketches
Dim originElements4 As OriginElements
Set originElements4 = part4.OriginElements
Dim reference57 As Reference
Set reference57 = originElements4.PlaneYZ
Dim sketch10 As Sketch
Set sketch10 = sketches4.Add(reference57)

Dim arrayOfVariantOfDouble10(8)
arrayOfVariantOfDouble10(0) = 0#
arrayOfVariantOfDouble10(1) = 0#
arrayOfVariantOfDouble10(2) = 0#
arrayOfVariantOfDouble10(3) = 0#
arrayOfVariantOfDouble10(4) = 1#
arrayOfVariantOfDouble10(5) = 0#
arrayOfVariantOfDouble10(6) = 0#
arrayOfVariantOfDouble10(7) = 0#
arrayOfVariantOfDouble10(8) = 1#
Set sketch10Variant = sketch10
sketch10Variant.SetAbsoluteAxisData arrayOfVariantOfDouble10
part4.InWorkObject = sketch10

'PAD
Dim factory2D10 As Factory2D
Set factory2D10 = sketch10.OpenEdition()
Dim geometricElements10 As GeometricElements
Set geometricElements10 = sketch10.GeometricElements

Dim axis2D10 As Axis2D
Set axis2D10 = geometricElements10.Item("AbsoluteAxis")
Dim line2D27 As Line2D
Set line2D27 = axis2D10.GetItem("HDirection")
line2D27.ReportName = 1
Dim line2D28 As Line2D
Set line2D28 = axis2D10.GetItem("VDirection")
line2D28.ReportName = 2

Dim circle2D8 As Circle2D
Set circle2D8 = factory2D10.CreateClosedCircle(0#, 0#, F / 2 + (E - F) / 4 + r1)
Dim point2D16 As Point2D
Set point2D16 = axis2D10.GetItem("Origin")
circle2D8.CenterPoint = point2D16
circle2D8.ReportName = 3

Dim constraints10 As Constraints
Set constraints10 = sketch10.Constraints

Dim reference58 As Reference
Set reference58 = part4.CreateReferenceFromObject(circle2D8)
Dim constraint26 As Constraint
```

```
Set constraint26 = constraints10.AddMonoEltCst(catCstTypeRadius, reference58)
constraint26.Mode = catCstModeDrivingDimension
Dim length21 As Length
Set length21 = constraint26.Dimension
length21.Value = F / 2 + (E - F) / 4 + r1

sketch10.CloseEdition
part4.InWorkObject = sketch10
part4.Update

Dim shapeFactory4 As ShapeFactory
Set shapeFactory4 = part4.ShapeFactory
Dim pad4 As Pad
Set pad4 = shapeFactory4.AddNewPad(sketch10, B / 2)
Dim limit6 As Limit
Set limit6 = pad4.FirstLimit
Dim length22 As Length
Set length22 = limit6.Dimension
length22.Value = B / 2
pad4.IsSymmetric = True

part4.Update

'POCKET
Dim reference59 As Reference
Set reference59 = part4.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")

Dim sketch11 As Sketch
Set sketch11 = sketches4.Add(reference59)

Dim arrayOfVariantOfDouble11(8)
arrayOfVariantOfDouble11(0) = B / 2
arrayOfVariantOfDouble11(1) = 0#
arrayOfVariantOfDouble11(2) = 0#
arrayOfVariantOfDouble11(3) = 0#
arrayOfVariantOfDouble11(4) = 1#
arrayOfVariantOfDouble11(5) = 0#
arrayOfVariantOfDouble11(6) = 0#
arrayOfVariantOfDouble11(7) = 0#
arrayOfVariantOfDouble11(8) = 1#
Set sketch11Variant = sketch11
sketch11Variant.SetAbsoluteAxisData arrayOfVariantOfDouble11

part4.InWorkObject = sketch11

Dim factory2D11 As Factory2D
Set factory2D11 = sketch11.OpenEdition()
Dim geometricElements11 As GeometricElements
Set geometricElements11 = sketch11.GeometricElements
Dim axis2D11 As Axis2D
Set axis2D11 = geometricElements11.Item("AbsoluteAxis")
Dim line2D29 As Line2D
Set line2D29 = axis2D11.GetItem("HDirection")
line2D29.ReportName = 1
Dim line2D30 As Line2D
Set line2D30 = axis2D11.GetItem("VDirection")
line2D30.ReportName = 2

Dim circle2D9 As Circle2D
Set circle2D9 = factory2D11.CreateClosedCircle(0#, 0#, F / 2 + (E - F) / 4 - r1)
Dim point2D17 As Point2D
Set point2D17 = axis2D11.GetItem("Origin")
circle2D9.CenterPoint = point2D17
circle2D9.ReportName = 3

Dim constraints11 As Constraints
Set constraints11 = sketch11.Constraints
Dim reference61 As Reference
Set reference61 = part4.CreateReferenceFromObject(circle2D9)
Dim constraint27 As Constraint
Set constraint27 = constraints11.AddMonoEltCst(catCstTypeRadius, reference61)
constraint27.Mode = catCstModeDrivingDimension
Dim length23 As Length
Set length23 = constraint27.Dimension
length23.Value = F / 2 + (E - F) / 4 - r1
sketch11.CloseEdition
part4.InWorkObject = sketch11
part4.Update

Dim pocket3 As Pocket
Set pocket3 = shapeFactory4.AddNewPocket(sketch11, B / 2)
Dim limit7 As Limit
Set limit7 = pocket3.FirstLimit
limit7.LimitMode = catUpThruNextLimit

part4.Update

'HUECO PARA LOS RODILLOS
Dim reference62 As Reference
Set reference62 = originElements4.PlaneXY
Dim sketch12 As Sketch
Set sketch12 = sketches4.Add(reference62)

Dim arrayOfVariantOfDouble12(8)
arrayOfVariantOfDouble12(0) = 0#
arrayOfVariantOfDouble12(1) = 0#
arrayOfVariantOfDouble12(2) = 0#
arrayOfVariantOfDouble12(3) = 1#
arrayOfVariantOfDouble12(4) = 0#
arrayOfVariantOfDouble12(5) = 0#
arrayOfVariantOfDouble12(6) = 0#
arrayOfVariantOfDouble12(7) = 1#
arrayOfVariantOfDouble12(8) = 0#
Set sketch12Variant = sketch12
sketch12Variant.SetAbsoluteAxisData arrayOfVariantOfDouble12

part4.InWorkObject = sketch12

Dim factory2D12 As Factory2D
Set factory2D12 = sketch12.OpenEdition()
Dim geometricElements12 As GeometricElements
Set geometricElements12 = sketch12.GeometricElements
Dim axis2D12 As Axis2D
Set axis2D12 = geometricElements12.Item("AbsoluteAxis")
Dim line2D31 As Line2D
Set line2D31 = axis2D12.GetItem("HDirection")
line2D31.ReportName = 1
```

```
Dim line2D32 As Line2D
Set line2D32 = axis2D12.GetItem("VDirection")
line2D32.ReportName = 2

Dim point2D18 As Point2D
Set point2D18 = factory2D12.CreatePoint(-B / 4, -(E
- F) / 4)
point2D18.ReportName = 3

Dim point2D19 As Point2D
Set point2D19 = factory2D12.CreatePoint(B / 4, -(E -
F) / 4)
point2D19.ReportName = 4

Dim line2D33 As Line2D
Set line2D33 = factory2D12.CreateLine(-B / 4, -(E -
F) / 4, B / 4, -(E - F) / 4)
line2D33.ReportName = 5
line2D33.StartPoint = point2D18
line2D33.EndPoint = point2D19

Dim constraints12 As Constraints
Set constraints12 = sketch12.Constraints

Dim reference63 As Reference
Set reference63 =
part4.CreateReferenceFromObject(line2D33)
Dim reference64 As Reference
Set reference64 =
part4.CreateReferenceFromObject(line2D31)
Dim constraint28 As Constraint
Set constraint28 =
constraints12.AddBiEltCst(catCstTypeHorizontality,
reference63, reference64)
constraint28.Mode = catCstModeDrivingDimension

Dim reference65 As Reference
Set reference65 =
part4.CreateReferenceFromObject(line2D33)
Dim constraint29 As Constraint
Set constraint29 =
constraints12.AddMonoEltCst(catCstTypeLength,
reference65)
constraint29.Mode = catCstModeDrivingDimension
Dim length24 As Length
Set length24 = constraint29.Dimension
length24.Value = B / 2

Dim point2D20 As Point2D
Set point2D20 = factory2D12.CreatePoint(B / 4, (E -
F) / 4)
point2D20.ReportName = 6

Dim line2D34 As Line2D
Set line2D34 = factory2D12.CreateLine(B / 4, -(E -
F) / 4, B / 4, (E - F) / 4)
line2D34.ReportName = 7
line2D34.StartPoint = point2D19
line2D34.EndPoint = point2D20

Dim reference67 As Reference
Set reference67 =
part4.CreateReferenceFromObject(line2D34)
Dim reference68 As Reference
Set reference68 =
part4.CreateReferenceFromObject(line2D32)
Dim constraint30 As Constraint
Set constraint30 =
constraints12.AddBiEltCst(catCstTypeVerticality,
reference67, reference68)
constraint30.Mode = catCstModeDrivingDimension

Dim reference69 As Reference
Set reference69 =
part4.CreateReferenceFromObject(line2D34)
Dim constraint31 As Constraint
Set constraint31 =
constraints12.AddMonoEltCst(catCstTypeLength,
reference69)
constraint31.Mode = catCstModeDrivingDimension
Dim length25 As Length
Set length25 = constraint31.Dimension
length25.Value = (E - F) / 2

Dim point2D21 As Point2D
Set point2D21 = factory2D12.CreatePoint(-B / 4, (E -
F) / 4)
point2D21.ReportName = 8

Dim line2D35 As Line2D
Set line2D35 = factory2D12.CreateLine(B / 4, (E - F)
/ 4, -B / 4, (E - F) / 4)
line2D35.ReportName = 9
line2D35.StartPoint = point2D20
line2D35.EndPoint = point2D21

Dim reference70 As Reference
Set reference70 =
part4.CreateReferenceFromObject(line2D35)
Dim reference71 As Reference
Set reference71 =
part4.CreateReferenceFromObject(line2D31)
Dim constraint32 As Constraint
Set constraint32 =
constraints12.AddBiEltCst(catCstTypeHorizontality,
reference70, reference71)
constraint32.Mode = catCstModeDrivingDimension

Dim line2D36 As Line2D
Set line2D36 = factory2D12.CreateLine(-B / 4, (E -
F) / 4, -B / 4, -(E - F) / 4)
line2D36.ReportName = 10
line2D36.StartPoint = point2D21
line2D36.EndPoint = point2D18

Dim reference72 As Reference
Set reference72 =
part4.CreateReferenceFromObject(line2D36)
Dim reference73 As Reference
Set reference73 =
part4.CreateReferenceFromObject(line2D32)
Dim constraint33 As Constraint
Set constraint33 =
constraints12.AddBiEltCst(catCstTypeVerticality,
reference72, reference73)
constraint33.Mode = catCstModeDrivingDimension

Dim reference74 As Reference
Set reference74 =
part4.CreateReferenceFromObject(line2D32)
Dim reference75 As Reference
Set reference75 =
part4.CreateReferenceFromObject(line2D34)
Dim constraint34 As Constraint
```

```
Set constraint34 = constraints12.AddBiEltCst(catCstTypeDistance, reference74, reference75)
constraint34.Mode = catCstModeDrivingDimension
Dim length26 As Length
Set length26 = constraint34.Dimension
length26.Value = B / 4

Dim reference76 As Reference
Set reference76 = part4.CreateReferenceFromObject(line2D31)
Dim reference77 As Reference
Set reference77 = part4.CreateReferenceFromObject(line2D33)
Dim constraint35 As Constraint
Set constraint35 = constraints12.AddBiEltCst(catCstTypeDistance, reference76, reference77)
constraint35.Mode = catCstModeDrivingDimension
Dim length27 As Length
Set length27 = constraint35.Dimension
length27.Value = (E - F) / 4

sketch12.CloseEdition
part4.InWorkObject = sketch12
part4.Update

Dim pocket4 As Pocket
Set pocket4 = shapeFactory4.AddNewPocket(sketch12, 20#)
Dim limit8 As Limit
Set limit8 = pocket4.FirstLimit
pocket4.DirectionOrientation = catRegularOrientation
limit8.LimitMode = catUpToLastLimit

part4.Update

'PATRÓN CIRCULAR
Dim reference78 As Reference
Set reference78 = part4.CreateReferenceFromName("")
Dim reference79 As Reference
Set reference79 = part4.CreateReferenceFromName("")

Dim circPattern1 As CircPattern
Set circPattern1 = shapeFactory4.AddNewCircPattern(pocket4, 1, 2, 20#, 45#, 1, 1, reference78, reference79, True, 0#, True)
circPattern1.CircularPatternParameters = catInstancesandAngularSpacing

Dim angularRepartition As angularRepartition
Set angularRepartition = circPattern1.angularRepartition

'calculo del número de agujeros para ello definimos primero el número pi
Const PI = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es F / 2 + (E - F) / 4
Dim longitud As Double
longitud = 2 * PI * (F / 2 + (E - F) / 4)
'crearemos una variable auxiliar que usara el comando matemático Round para redondear el numero de huecos que tendrá el separador.
'para calcular este número hacemos la hipótesis de que la distancia  entre rodillos será el diametro del rodillo
Dim variableaux As Double
variableaux = Math.Round(longitud / (3 * (E - F) / 4))
'ya tenemos el numero de huecos del separador
Dim intParam As intParam
Set intParam = angularRepartition.InstancesCount
intParam.Value = variableaux
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angle As angle
Set angle = angularRepartition.AngularSpacing
angle.Value = 360 / variableaux

Dim reference80 As Reference
Set reference80 = part4.CreateReferenceFromObject(line2D31)
circPattern1.SetRotationAxis reference80

part4.UpdateObject circPattern1
part4.Update

Dim product4 As Product
Set product4 = partDocument4.GetItem("Part4")
product4.PartNumber = "jaula"

partDocument4.SaveAs objPath & "\jaula.CATPart"
partDocument4.Close

'=====================================
'----------------------------------ENSAMBLAJE-----
'=====================================


Dim documents5 As Documents
Set documents5 = CATIA.Documents
Dim productDocument1 As ProductDocument
Set productDocument1 = documents5.Add("Product")
Dim product5 As Product
Set product5 = productDocument1.Product
Dim products1 As Products
Set products1 = product5.Products

Dim arrayOfVariantOfBSTR1(0)
arrayOfVariantOfBSTR1(0) = objPath & "\aroext.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR1, "All"

Dim constraints13 As Constraints
Set constraints13 = product5.Connections("CATIAConstraints")
'Fix del aro externo
Dim reference81 As Reference
Set reference81 = product5.CreateReferenceFromName("Product5/aroexterno.1/!Product5/aroexterno.1/")
Dim constraint36 As Constraint
Set constraint36 = constraints13.AddMonoEltCst(catCstTypeReference, reference81)
```

```
Dim arrayOfVariantOfBSTR2(0)
arrayOfVariantOfBSTR2(0) = objPath &
"\aroint.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR2, "All"

'coincidencia de ejes de los aros externo e interno
Dim reference82 As Reference
Set reference82 =
product5.CreateReferenceFromName("Product5/aroi
nterno.1/!Axis:(Selection_RSur:(Face:(Brp:(Pocket.1
;0:(Brp:(Sketch.2;3)));None:();Cf11:());Slot.1_Result
OUT;Z0;G3055))")
Dim reference83 As Reference
Set reference83 =
product5.CreateReferenceFromName("Product5/aroe
xterno.1/!Axis:(Selection_RSur:(Face:(Brp:(Pad.1;0:(
Brp:(Sketch.1;3)));None:();Cf11:());Pocket.1_Result
OUT;Z0;G3055))")
Dim constraint37 As Constraint
Set constraint37 =
constraints13.AddBiEltCst(catCstTypeOn,
reference82, reference83)

Dim arrayOfVariantOfBSTR3(0)
arrayOfVariantOfBSTR3(0) = objPath &
"\rodillos.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR3, "All"

Dim arrayOfVariantOfBSTR4(0)
arrayOfVariantOfBSTR4(0) = objPath &
"\jaula.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR4, "All"

'coincidencia de ejes aro externo y jaula
Dim reference84 As Reference
Set reference84 =
product5.CreateReferenceFromName("Product5/jaula
.1/!Axis:(Selection_RSur:(Face:(Brp:(Pocket.1;0:(Br
p:(Sketch.2;3)));None:();Cf11:());CircPattern.1_Resu
ltOUT;Z0;G3055))")
Dim constraint38 As Constraint
Set constraint38 =
constraints13.AddBiEltCst(catCstTypeOn,
reference83, reference84)

'fijamos la distancia entre la cara lateral del aro
externo y el interno
Dim reference89 As Reference
Set reference89 =
product5.CreateReferenceFromName("Product5/aroe
xterno.1/!Selection_RSur:(Face:(Brp:(Pad.1;2);None:
();Cf11:());Pocket.1_ResultOUT;Z0;G3055)")
Dim reference90 As Reference
Set reference90 =
product5.CreateReferenceFromName("Product5/aroi
nterno.1/!Selection_RSur:(Face:(Brp:(Pad.1;2);None:
();Cf11:());Slot.1_ResultOUT;Z0;G3055)")
Dim constraint39 As Constraint
Set constraint39 =
constraints13.AddBiEltCst(catCstTypeDistance,
reference89, reference90)
Dim length28 As Length
Set length28 = constraint39.Dimension
length28.Value = 0#
constraint39.Orientation = catCstOrientSame

'idem para el aro externo y la jaula
Dim reference91 As Reference
Set reference91 =
product5.CreateReferenceFromName("Product5/jaula
.1/!Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf1
1:());CircPattern.1_ResultOUT;Z0;G3055)")
Dim constraint40 As Constraint
Set constraint40 =
constraints13.AddBiEltCst(catCstTypeDistance,
reference89, reference91)
Dim length29 As Length
Set length29 = constraint40.Dimension
length29.Value = 0#
constraint40.Orientation = catCstOrientSame
product5.Update

'Fijamos los rodillos a los huecos de la jaula.
Dim referencec1 As Reference
Set referencec1 =
product5.CreateReferenceFromName("Product5/rodil
los.1/!zx plane")
Dim referencec2 As Reference
Set referencec2 =
product5.CreateReferenceFromName("Product5/jaula
.1/!zx plane")
Dim constraintc1 As Constraint
Set constraintc1 =
constraints13.AddBiEltCst(catCstTypeDistance,
referencec1, referencec2)

Dim referencec3 As Reference
Set referencec3 =
product5.CreateReferenceFromName("Product5/rodil
los.1/!yz plane")
Dim referencec4 As Reference
Set referencec4 =
product5.CreateReferenceFromName("Product5/jaula
.1/!yz plane")
Dim constraintc2 As Constraint
Set constraintc2 =
constraints13.AddBiEltCst(catCstTypeDistance,
referencec3, referencec4)

Dim referencec5 As Reference
Set referencec5 =
product5.CreateReferenceFromName("Product5/rodil
los.1/!xy plane")
Dim referencec6 As Reference
Set referencec6 =
product5.CreateReferenceFromName("Product5/jaula
.1/!xy plane")
Dim constraintc3 As Constraint
Set constraintc3 =
constraints13.AddBiEltCst(catCstTypeDistance,
referencec5, referencec6)

product5.Update
productDocument1.SaveAs objPath &
"\RodillosNUP.CATProduct"
```

```
'productDocument1.Close
End Sub
```

- **Rodamiento de rodillos cilíndricos tipo NU.**

Ídem al Rodamiento tipo NUP salvo el anillo interno.

```
'====================================
'----------------------------------aro interno----------------
'====================================
Dim documents2 As Documents
Set documents2 = CATIA.Documents
Dim partDocument2 As PartDocument
Set partDocument2 = documents2.Add("Part")
Dim part2 As Part
Set part2 = partDocument2.Part
Dim bodies2 As Bodies
Set bodies2 = part2.Bodies
Dim body2 As Body
Set body2 = bodies2.Item("PartBody")
Dim sketches2 As Sketches
Set sketches2 = body2.Sketches
Dim originElements2 As OriginElements
Set originElements2 = part2.OriginElements
Dim reference26 As Reference
Set reference26 = originElements2.PlaneYZ
Dim sketch5 As Sketch
Set sketch5 = sketches2.Add(reference26)

Dim arrayOfVariantOfDouble5(8)
arrayOfVariantOfDouble5(0) = 0#
arrayOfVariantOfDouble5(1) = 0#
arrayOfVariantOfDouble5(2) = 0#
arrayOfVariantOfDouble5(3) = 0#
arrayOfVariantOfDouble5(4) = 1#
arrayOfVariantOfDouble5(5) = 0#
arrayOfVariantOfDouble5(6) = 0#
arrayOfVariantOfDouble5(7) = 0#
arrayOfVariantOfDouble5(8) = 1#
Set sketch5Variant = sketch5
sketch5Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble5

part2.InWorkObject = sketch5

'PAD
Dim factory2D5 As Factory2D
Set factory2D5 = sketch5.OpenEdition()
Dim geometricElements5 As GeometricElements
Set geometricElements5 = sketch5.GeometricElements

Dim axis2D5 As Axis2D
Set axis2D5 = geometricElements5.Item("AbsoluteAxis")
Dim line2D13 As Line2D
Set line2D13 = axis2D5.GetItem("HDirection")
line2D13.ReportName = 1
Dim line2D14 As Line2D
Set line2D14 = axis2D5.GetItem("VDirection")
line2D14.ReportName = 2

Dim circle2D4 As Circle2D
Set circle2D4 = factory2D5.CreateClosedCircle(0#, 0#, F / 2)
Dim point2D8 As Point2D
Set point2D8 = axis2D5.GetItem("Origin")
circle2D4.CenterPoint = point2D8
circle2D4.ReportName = 3

Dim constraints5 As Constraints
Set constraints5 = sketch5.Constraints

Dim reference27 As Reference
Set reference27 = part2.CreateReferenceFromObject(circle2D4)
Dim constraint12 As Constraint
Set constraint12 = constraints5.AddMonoEltCst(catCstTypeRadius, reference27)
constraint12.Mode = catCstModeDrivingDimension

Dim length10 As Length
Set length10 = constraint12.Dimension
length10.Value = F / 2

sketch5.CloseEdition
part2.InWorkObject = sketch5
part2.Update

Dim shapeFactory2 As ShapeFactory
Set shapeFactory2 = part2.ShapeFactory
Dim pad2 As Pad
Set pad2 = shapeFactory2.AddNewPad(sketch5, B / 2)
Dim limit3 As Limit
Set limit3 = pad2.FirstLimit
Dim length11 As Length
Set length11 = limit3.Dimension
length11.Value = B / 2
pad2.IsSymmetric = True

part2.Update

'POCKET
Dim reference28 As Reference
Set reference28 = part2.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")

Dim sketch6 As Sketch
Set sketch6 = sketches2.Add(reference28)

Dim arrayOfVariantOfDouble6(8)
arrayOfVariantOfDouble6(0) = B / 2
arrayOfVariantOfDouble6(1) = 0#
arrayOfVariantOfDouble6(2) = 0#
arrayOfVariantOfDouble6(3) = 0#
arrayOfVariantOfDouble6(4) = 1#
arrayOfVariantOfDouble6(5) = 0#
arrayOfVariantOfDouble6(6) = 0#
arrayOfVariantOfDouble6(7) = 0#
arrayOfVariantOfDouble6(8) = 1#
Set sketch6Variant = sketch6
sketch6Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble6

part2.InWorkObject = sketch6

Dim factory2D6 As Factory2D
Set factory2D6 = sketch6.OpenEdition()
Dim geometricElements6 As GeometricElements
Set geometricElements6 = sketch6.GeometricElements
```

```
Dim axis2D6 As Axis2D
Set axis2D6 = geometricElements6.Item("AbsoluteAxis")
Dim line2D15 As Line2D
Set line2D15 = axis2D6.GetItem("HDirection")
line2D15.ReportName = 1
Dim line2D16 As Line2D
Set line2D16 = axis2D6.GetItem("VDirection")
line2D16.ReportName = 2

Dim circle2D5 As Circle2D
Set circle2D5 = factory2D6.CreateClosedCircle(0#, 0#, d_ / 2)
Dim point2D9 As Point2D
Set point2D9 = axis2D6.GetItem("Origin")
circle2D5.CenterPoint = point2D9
circle2D5.ReportName = 3

Dim constraints6 As Constraints
Set constraints6 = sketch6.Constraints

Dim reference29 As Reference
Set reference29 = part2.CreateReferenceFromObject(circle2D5)
Dim constraint13 As Constraint
Set constraint13 = constraints6.AddMonoEltCst(catCstTypeRadius, reference29)
constraint13.Mode = catCstModeDrivingDimension
Dim length12 As Length
Set length12 = constraint13.Dimension
length12.Value = d_ / 2

sketch6.CloseEdition
part2.InWorkObject = sketch6
part2.Update

Dim pocket2 As Pocket
Set pocket2 = shapeFactory2.AddNewPocket(sketch6, B / 2)
Dim limit4 As Limit
Set limit4 = pocket2.FirstLimit
limit4.LimitMode = catUpThruNextLimit

part2.Update

'Chaflán
Dim reference30 As Reference
Set reference30 = part2.CreateReferenceFromName("")
Dim constRadEdgeFillet2 As ConstRadEdgeFillet
Set constRadEdgeFillet2 = shapeFactory2.AddNewSolidEdgeFilletWithConstantRadius(reference30, catTangencyFilletEdgePropagation, r1)

Dim reference31 As Reference
Set reference31 = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;2);None:();Cf11:());Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket1)
constRadEdgeFillet2.AddObjectToFillet reference31
constRadEdgeFillet2.EdgePropagation = catTangencyFilletEdgePropagation

Dim reference32 As Reference
Set reference32 = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());Face:(Brp:(Pad.1;1);None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket1)
constRadEdgeFillet2.AddObjectToFillet reference32
constRadEdgeFillet2.EdgePropagation = catTangencyFilletEdgePropagation
part2.Update

Dim product2 As Product
Set product2 = partDocument2.GetItem("Part2")
product2.PartNumber = "arointerno"
partDocument2.SaveAs objPath & "\aroint.CATPart"
partDocument2.Close
```

- **Rodamiento de rodillos cilíndricos tipo N.**

Ídem al rodamiento tipo NUP salvo el aro externo.

```
'====================================
'----------------------------------Aro externo-------------
'====================================
Dim documents1 As Documents
Set documents1 = CATIA.Documents
Dim partDocument1 As PartDocument
Set partDocument1 = documents1.Add("Part")
Dim part1 As Part
Set part1 = partDocument1.Part
Dim bodies1 As Bodies
Set bodies1 = part1.Bodies
Dim body1 As Body
Set body1 = bodies1.Item("PartBody")
Dim sketches1 As Sketches
Set sketches1 = body1.Sketches
Dim originElements1 As OriginElements
Set originElements1 = part1.OriginElements
Dim reference1 As Reference
Set reference1 = originElements1.PlaneYZ
Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0#
arrayOfVariantOfDouble1(1) = 0#
arrayOfVariantOfDouble1(2) = 0#
arrayOfVariantOfDouble1(3) = 0#
arrayOfVariantOfDouble1(4) = 1#
arrayOfVariantOfDouble1(5) = 0#
arrayOfVariantOfDouble1(6) = 0#
arrayOfVariantOfDouble1(7) = 0#
arrayOfVariantOfDouble1(8) = 1#
Set sketch1Variant = sketch1
sketch1Variant.SetAbsoluteAxisData arrayOfVariantOfDouble1
part1.InWorkObject = sketch1

Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()
Dim geometricElements1 As GeometricElements
Set geometricElements1 = sketch1.GeometricElements
Dim axis2D1 As Axis2D
Set axis2D1 = geometricElements1.Item("AbsoluteAxis")
```

```
Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2

Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateClosedCircle(0#, 0#, D / 2)
Dim point2D1 As Point2D
Set point2D1 = axis2D1.GetItem("Origin")
circle2D1.CenterPoint = point2D1
circle2D1.ReportName = 3

Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints
Dim reference2 As Reference
Set reference2 = part1.CreateReferenceFromObject(circle2D1)
Dim constraint1 As Constraint
Set constraint1 = constraints1.AddMonoEltCst(catCstTypeRadius, reference2)
constraint1.Mode = catCstModeDrivingDimension
Dim length1 As Length
Set length1 = constraint1.Dimension
length1.Value = D / 2

sketch1.CloseEdition
part1.InWorkObject = sketch1
part1.Update

'PAD
Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory
Dim pad1 As Pad
Set pad1 = shapeFactory1.AddNewPad(sketch1, B / 2)
pad1.IsSymmetric = True
Dim limit1 As Limit
Set limit1 = pad1.FirstLimit
Dim length2 As Length
Set length2 = limit1.Dimension
length2.Value = B / 2
part1.Update

'CHAFLÁN
Dim reference3 As Reference
Set reference3 = part1.CreateReferenceFromName("")
Dim constRadEdgeFillet1 As ConstRadEdgeFillet
Set constRadEdgeFillet1 = shapeFactory1.AddNewSolidEdgeFilletWithConstantRadius(reference3, catTangencyFilletEdgePropagation, r1)

Dim reference4 As Reference
Set reference4 = part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());Face:(Brp:(Pad.1;2);None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pad1)
constRadEdgeFillet1.AddObjectToFillet reference4
constRadEdgeFillet1.EdgePropagation = catTangencyFilletEdgePropagation
Dim reference5 As Reference
Set reference5 = part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;1);None:();Cf11:());Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pad1)
constRadEdgeFillet1.AddObjectToFillet reference5
constRadEdgeFillet1.EdgePropagation = catTangencyFilletEdgePropagation

part1.Update

'POCKET
Dim reference6 As Reference
Set reference6 = part1.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055)")

Dim sketch2 As Sketch
Set sketch2 = sketches1.Add(reference6)
Dim arrayOfVariantOfDouble2(8)
arrayOfVariantOfDouble2(0) = B / 2
arrayOfVariantOfDouble2(1) = 0#
arrayOfVariantOfDouble2(2) = 0#
arrayOfVariantOfDouble2(3) = 0#
arrayOfVariantOfDouble2(4) = 1#
arrayOfVariantOfDouble2(5) = 0#
arrayOfVariantOfDouble2(6) = 0#
arrayOfVariantOfDouble2(7) = 0#
arrayOfVariantOfDouble2(8) = 1#
Set sketch2Variant = sketch2
sketch2Variant.SetAbsoluteAxisData arrayOfVariantOfDouble2
part1.InWorkObject = sketch2

Dim factory2D2 As Factory2D
Set factory2D2 = sketch2.OpenEdition()
Dim geometricElements2 As GeometricElements
Set geometricElements2 = sketch2.GeometricElements
Dim axis2D2 As Axis2D
Set axis2D2 = geometricElements2.Item("AbsoluteAxis")
Dim line2D3 As Line2D
Set line2D3 = axis2D2.GetItem("HDirection")
line2D3.ReportName = 1
Dim line2D4 As Line2D
Set line2D4 = axis2D2.GetItem("VDirection")
line2D4.ReportName = 2

Dim circle2D2 As Circle2D
Set circle2D2 = factory2D2.CreateClosedCircle(0#, 0#, E / 2)
Dim point2D2 As Point2D
Set point2D2 = axis2D2.GetItem("Origin")
circle2D2.CenterPoint = point2D2
circle2D2.ReportName = 3

Dim constraints2 As Constraints
Set constraints2 = sketch2.Constraints
Dim reference7 As Reference
```

```
Set reference7 = part1.CreateReferenceFromObject(circle2D2)
Dim constraint2 As Constraint
Set constraint2 = constraints2.AddMonoEltCst(catCstTypeRadius, reference7)
constraint2.Mode = catCstModeDrivingDimension
Dim length4 As Length
Set length4 = constraint2.Dimension
length4.Value = E / 2

sketch2.CloseEdition
part1.InWorkObject = sketch2
part1.Update

Dim pocket1 As Pocket
Set pocket1 = shapeFactory1.AddNewPocket(sketch2, B / 2)
Dim limit2 As Limit
Set limit2 = pocket1.FirstLimit
limit2.LimitMode = catUpToLastLimit
part1.Update

Dim product1 As Product
Set product1 = partDocument1.GetItem("Part1")
product1.PartNumber = "aroexterno"
partDocument1.SaveAs objPath & "\aroext.catpart"
partDocument1.Close
```

- **Rodamiento de rodillos cilíndricos tipo NJ.**

Ídem al rodamiento tipo NUP salvo el anillo interno.

```
'====================================
'----------------------------------anillo interno----------
'====================================
Dim documents2 As Documents
Set documents2 = CATIA.Documents
Dim partDocument2 As PartDocument
Set partDocument2 = documents2.Add("Part")
Dim part2 As Part
Set part2 = partDocument2.Part
Dim bodies2 As Bodies
Set bodies2 = part2.Bodies
Dim body2 As Body
Set body2 = bodies2.Item("PartBody")
Dim sketches2 As Sketches
Set sketches2 = body2.Sketches
Dim originElements2 As OriginElements
Set originElements2 = part2.OriginElements
Dim reference26 As Reference
Set reference26 = originElements2.PlaneYZ
Dim sketch5 As Sketch
Set sketch5 = sketches2.Add(reference26)

Dim arrayOfVariantOfDouble5(8)
arrayOfVariantOfDouble5(0) = 0#
arrayOfVariantOfDouble5(1) = 0#
arrayOfVariantOfDouble5(2) = 0#
arrayOfVariantOfDouble5(3) = 0#
arrayOfVariantOfDouble5(4) = 1#
arrayOfVariantOfDouble5(5) = 0#
arrayOfVariantOfDouble5(6) = 0#
arrayOfVariantOfDouble5(7) = 0#
arrayOfVariantOfDouble5(8) = 1#
Set sketch5Variant = sketch5
sketch5Variant.SetAbsoluteAxisData arrayOfVariantOfDouble5
part2.InWorkObject = sketch5

'PAD
Dim factory2D5 As Factory2D
Set factory2D5 = sketch5.OpenEdition()
Dim geometricElements5 As GeometricElements
Set geometricElements5 = sketch5.GeometricElements

Dim axis2D5 As Axis2D
Set axis2D5 = geometricElements5.Item("AbsoluteAxis")
Dim line2D13 As Line2D
Set line2D13 = axis2D5.GetItem("HDirection")
line2D13.ReportName = 1
Dim line2D14 As Line2D
Set line2D14 = axis2D5.GetItem("VDirection")
line2D14.ReportName = 2

Dim circle2D4 As Circle2D
Set circle2D4 = factory2D5.CreateClosedCircle(0#, 0#, d_1 / 2)
Dim point2D8 As Point2D
Set point2D8 = axis2D5.GetItem("Origin")
circle2D4.CenterPoint = point2D8
circle2D4.ReportName = 3

Dim constraints5 As Constraints
Set constraints5 = sketch5.Constraints

Dim reference27 As Reference
Set reference27 = part2.CreateReferenceFromObject(circle2D4)
Dim constraint12 As Constraint
Set constraint12 = constraints5.AddMonoEltCst(catCstTypeRadius, reference27)
constraint12.Mode = catCstModeDrivingDimension

Dim length10 As Length
Set length10 = constraint12.Dimension
length10.Value = d_1 / 2

sketch5.CloseEdition
part2.InWorkObject = sketch5
part2.Update

Dim shapeFactory2 As ShapeFactory
Set shapeFactory2 = part2.ShapeFactory
Dim pad2 As Pad
Set pad2 = shapeFactory2.AddNewPad(sketch5, B / 2)
Dim limit3 As Limit
Set limit3 = pad2.FirstLimit
Dim length11 As Length
Set length11 = limit3.Dimension
length11.Value = B / 2
pad2.IsSymmetric = True

part2.Update

'POCKET
Dim reference28 As Reference
Set reference28 = part2.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")
```

```
Dim sketch6 As Sketch
Set sketch6 = sketches2.Add(reference28)

Dim arrayOfVariantOfDouble6(8)
arrayOfVariantOfDouble6(0) = B / 2
arrayOfVariantOfDouble6(1) = 0#
arrayOfVariantOfDouble6(2) = 0#
arrayOfVariantOfDouble6(3) = 0#
arrayOfVariantOfDouble6(4) = 1#
arrayOfVariantOfDouble6(5) = 0#
arrayOfVariantOfDouble6(6) = 0#
arrayOfVariantOfDouble6(7) = 0#
arrayOfVariantOfDouble6(8) = 1#
Set sketch6Variant = sketch6
sketch6Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble6

part2.InWorkObject = sketch6

Dim factory2D6 As Factory2D
Set factory2D6 = sketch6.OpenEdition()
Dim geometricElements6 As GeometricElements
Set geometricElements6 =
sketch6.GeometricElements

Dim axis2D6 As Axis2D
Set axis2D6 =
geometricElements6.Item("AbsoluteAxis")
Dim line2D15 As Line2D
Set line2D15 = axis2D6.GetItem("HDirection")
line2D15.ReportName = 1
Dim line2D16 As Line2D
Set line2D16 = axis2D6.GetItem("VDirection")
line2D16.ReportName = 2

Dim circle2D5 As Circle2D
Set circle2D5 = factory2D6.CreateClosedCircle(0#,
0#, d_ / 2)
Dim point2D9 As Point2D
Set point2D9 = axis2D6.GetItem("Origin")
circle2D5.CenterPoint = point2D9
circle2D5.ReportName = 3

Dim constraints6 As Constraints
Set constraints6 = sketch6.Constraints

Dim reference29 As Reference
Set reference29 =
part2.CreateReferenceFromObject(circle2D5)
Dim constraint13 As Constraint
Set constraint13 =
constraints6.AddMonoEltCst(catCstTypeRadius,
reference29)
constraint13.Mode = catCstModeDrivingDimension
Dim length12 As Length
Set length12 = constraint13.Dimension
length12.Value = d_ / 2

sketch6.CloseEdition
part2.InWorkObject = sketch6
part2.Update

Dim pocket2 As Pocket
Set pocket2 =
shapeFactory2.AddNewPocket(sketch6, B / 2)
Dim limit4 As Limit
Set limit4 = pocket2.FirstLimit
limit4.LimitMode = catUpThruNextLimit

part2.Update

'Chaflán
Dim reference30 As Reference
Set reference30 =
part2.CreateReferenceFromName("")
Dim constRadEdgeFillet2 As ConstRadEdgeFillet
Set constRadEdgeFillet2 =
shapeFactory2.AddNewSolidEdgeFilletWithConstant
Radius(reference30,
catTangencyFilletEdgePropagation, r1)

Dim reference31 As Reference
Set reference31 =
part2.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;2);None:();Cf11:());Face:(Brp:(P
ocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());None:
(Limits1:();Limits2:());Cf11:());WithTemporaryBody
;WithoutBuildError;WithSelectingFeatureSupport;M
FBRepVersion_CXR15)", pocket1)
constRadEdgeFillet2.AddObjectToFillet reference31
constRadEdgeFillet2.EdgePropagation =
catTangencyFilletEdgePropagation
Dim reference32 As Reference
Set reference32 =
part2.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();
Cf11:());Face:(Brp:(Pad.1;1);None:();Cf11:());None:(
Limits1:();Limits2:());Cf11:());WithTemporaryBody;
WithoutBuildError;WithSelectingFeatureSupport;MF
BRepVersion_CXR15)", pocket1)
constRadEdgeFillet2.AddObjectToFillet reference32
constRadEdgeFillet2.EdgePropagation =
catTangencyFilletEdgePropagation
part2.Update

'SLOT
Dim reference33 As Reference
Set reference33 = originElements2.PlaneZX
Dim sketch7 As Sketch
Set sketch7 = sketches2.Add(reference33)

Dim arrayOfVariantOfDouble7(8)
arrayOfVariantOfDouble7(0) = 0#
arrayOfVariantOfDouble7(1) = 0#
arrayOfVariantOfDouble7(2) = 0#
arrayOfVariantOfDouble7(3) = -1#
arrayOfVariantOfDouble7(4) = 0#
arrayOfVariantOfDouble7(5) = 0#
arrayOfVariantOfDouble7(6) = 0#
arrayOfVariantOfDouble7(7) = -0#
arrayOfVariantOfDouble7(8) = 1#
Set sketch7Variant = sketch7
sketch7Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble7
part2.InWorkObject = sketch7

Dim factory2D7 As Factory2D
Set factory2D7 = sketch7.OpenEdition()
Dim geometricElements7 As GeometricElements
Set geometricElements7 =
sketch7.GeometricElements
Dim axis2D7 As Axis2D
```

```
Set axis2D7 = geometricElements7.Item("AbsoluteAxis")
Dim line2D17 As Line2D
Set line2D17 = axis2D7.GetItem("HDirection")
line2D17.ReportName = 1
Dim line2D18 As Line2D
Set line2D18 = axis2D7.GetItem("VDirection")
line2D18.ReportName = 2

'dibujamos un rectangulo
Dim point2D10 As Point2D
Set point2D10 = factory2D7.CreatePoint(-B / 4, F / 2)
point2D10.ReportName = 3
Dim point2D11 As Point2D
Set point2D11 = factory2D7.CreatePoint(B / 2, F / 2)
point2D11.ReportName = 4
Dim line2D19 As Line2D
Set line2D19 = factory2D7.CreateLine(-B / 4, F / 2, B / 2, F / 2)
line2D19.ReportName = 5
line2D19.StartPoint = point2D10
line2D19.EndPoint = point2D11

Dim constraints7 As Constraints
Set constraints7 = sketch7.Constraints

Dim reference34 As Reference
Set reference34 = part2.CreateReferenceFromObject(line2D19)
Dim reference35 As Reference
Set reference35 = part2.CreateReferenceFromObject(line2D17)
Dim constraint14 As Constraint
Set constraint14 = constraints7.AddBiEltCst(catCstTypeHorizontality, reference34, reference35)
constraint14.Mode = catCstModeDrivingDimension

Dim reference36 As Reference
Set reference36 = part2.CreateReferenceFromObject(line2D19)
Dim constraint15 As Constraint
Set constraint15 = constraints7.AddMonoEltCst(catCstTypeLength, reference36)
constraint15.Mode = catCstModeDrivingDimension
Dim length13 As Length
Set length13 = constraint15.Dimension
length13.Value = 3 * B / 2

Dim point2D12 As Point2D
Set point2D12 = factory2D7.CreatePoint(B / 2, E / 2)
point2D12.ReportName = 6

Dim line2D20 As Line2D
Set line2D20 = factory2D7.CreateLine(B / 2, F / 2, B / 2, E / 2)
line2D20.ReportName = 7
line2D20.StartPoint = point2D11
line2D20.EndPoint = point2D12

Dim reference37 As Reference
Set reference37 = part2.CreateReferenceFromObject(line2D20)
Dim reference38 As Reference
Set reference38 = part2.CreateReferenceFromObject(line2D18)
Dim constraint16 As Constraint
Set constraint16 = constraints7.AddBiEltCst(catCstTypeVerticality, reference37, reference38)
constraint16.Mode = catCstModeDrivingDimension

Dim reference39 As Reference
Set reference39 = part2.CreateReferenceFromObject(line2D20)
Dim constraint17 As Constraint
Set constraint17 = constraints7.AddMonoEltCst(catCstTypeLength, reference39)
constraint17.Mode = catCstModeDrivingDimension
Dim length14 As Length
Set length14 = constraint17.Dimension
length14.Value = (E - F) / 2

Dim point2D13 As Point2D
Set point2D13 = factory2D7.CreatePoint(-B / 4, E / 2)
point2D13.ReportName = 8

Dim line2D21 As Line2D
Set line2D21 = factory2D7.CreateLine(B / 2, E / 2, -B / 4, E / 2)
line2D21.ReportName = 9
line2D21.StartPoint = point2D12
line2D21.EndPoint = point2D13

Dim reference40 As Reference
Set reference40 = part2.CreateReferenceFromObject(line2D21)
Dim reference41 As Reference
Set reference41 = part2.CreateReferenceFromObject(line2D13)
Dim constraint18 As Constraint
Set constraint18 = constraints7.AddBiEltCst(catCstTypeHorizontality, reference40, reference41)
constraint18.Mode = catCstModeDrivingDimension

Dim line2D22 As Line2D
Set line2D22 = factory2D7.CreateLine(-B / 4, E / 2, -B / 4, F / 2)
line2D22.ReportName = 10
line2D22.StartPoint = point2D13
line2D22.EndPoint = point2D10

Dim reference42 As Reference
Set reference42 = part2.CreateReferenceFromObject(line2D22)
Dim reference43 As Reference
Set reference43 = part2.CreateReferenceFromObject(line2D18)
Dim constraint19 As Constraint
Set constraint19 = constraints7.AddBiEltCst(catCstTypeVerticality, reference42, reference43)
constraint19.Mode = catCstModeDrivingDimension

Dim reference44 As Reference
Set reference44 = part2.CreateReferenceFromObject(line2D22)
Dim reference45 As Reference
```

```
Set reference45 = part2.CreateReferenceFromObject(line2D18)
Dim constraint20 As Constraint
Set constraint20 = constraints7.AddBiEltCst(catCstTypeDistance, reference44, reference45)
constraint20.Mode = catCstModeDrivingDimension
Dim length15 As Length
Set length15 = constraint20.Dimension
length15.Value = B / 4

Dim reference46 As Reference
Set reference46 = part2.CreateReferenceFromObject(line2D19)
Dim reference47 As Reference
Set reference47 = part2.CreateReferenceFromObject(line2D17)
Dim constraint21 As Constraint
Set constraint21 = constraints7.AddBiEltCst(catCstTypeDistance, reference46, reference47)
constraint21.Mode = catCstModeDrivingDimension
Dim length16 As Length
Set length16 = constraint21.Dimension
length16.Value = F / 2

sketch7.CloseEdition
part2.InWorkObject = sketch7
part2.Update

Dim sketch8 As Sketch
Set sketch8 = sketches2.Add(reference26)

Dim arrayOfVariantOfDouble8(8)
arrayOfVariantOfDouble8(0) = 0#
arrayOfVariantOfDouble8(1) = 0#
arrayOfVariantOfDouble8(2) = 0#
arrayOfVariantOfDouble8(3) = 0#
arrayOfVariantOfDouble8(4) = 1#
arrayOfVariantOfDouble8(5) = 0#
arrayOfVariantOfDouble8(6) = 0#
arrayOfVariantOfDouble8(7) = 0#
arrayOfVariantOfDouble8(8) = 1#
Set sketch8Variant = sketch8
sketch8Variant.SetAbsoluteAxisData arrayOfVariantOfDouble8

part2.InWorkObject = sketch8

Dim factory2D8 As Factory2D
Set factory2D8 = sketch8.OpenEdition()
Dim geometricElements8 As GeometricElements
Set geometricElements8 = sketch8.GeometricElements
Dim axis2D8 As Axis2D
Set axis2D8 = geometricElements8.Item("AbsoluteAxis")
Dim line2D23 As Line2D
Set line2D23 = axis2D8.GetItem("HDirection")
line2D23.ReportName = 1
Dim line2D24 As Line2D
Set line2D24 = axis2D8.GetItem("VDirection")
line2D24.ReportName = 2

Dim circle2D6 As Circle2D
Set circle2D6 = factory2D8.CreateClosedCircle(0#, 0#, F / 2 + (E - F) / 4)
Dim point2D14 As Point2D
Set point2D14 = axis2D8.GetItem("Origin")
circle2D6.CenterPoint = point2D14
circle2D6.ReportName = 3

Dim constraints8 As Constraints
Set constraints8 = sketch8.Constraints

Dim reference48 As Reference
Set reference48 = part2.CreateReferenceFromObject(circle2D6)
Dim constraint22 As Constraint
Set constraint22 = constraints8.AddMonoEltCst(catCstTypeRadius, reference48)
constraint22.Mode = catCstModeDrivingDimension
Dim length17 As Length
Set length17 = constraint22.Dimension
length17.Value = F / 2 + (E - F) / 4

sketch8.CloseEdition
part2.InWorkObject = sketch8
part2.Update

Dim slot2 As Slot
Set slot2 = shapeFactory2.AddNewSlotFromRef(Nothing, Nothing)

Dim reference49 As Reference
Set reference49 = part2.CreateReferenceFromObject(sketch7)
slot2.SetProfileElement reference49
Dim reference50 As Reference
Set reference50 = part2.CreateReferenceFromObject(sketch8)
slot2.CenterCurveElement = reference50

part2.Update

Dim product2 As Product
Set product2 = partDocument2.GetItem("Part2")
product2.PartNumber = "arointerno"
partDocument2.SaveAs objPath & "\aroint.CATPart"
partDocument2.Close
```

- **Corona de rodillos cilíndricos.**

```
Private Sub CommandButton1_Click()
'=====================================
'----------------------------------jaula----------------------
'=====================================
Dim documents1 As Documents
Set documents1 = CATIA.Documents
Dim partDocument1 As PartDocument
Set partDocument1 = documents1.Add("Part")
Dim part1 As Part
Set part1 = partDocument1.Part
Dim bodies1 As Bodies
Set bodies1 = part1.Bodies
Dim body1 As Body
Set body1 = bodies1.Item("PartBody")
Dim sketches1 As Sketches
Set sketches1 = body1.Sketches
Dim originElements1 As OriginElements
Set originElements1 = part1.OriginElements
Dim reference1 As Reference
Set reference1 = originElements1.PlaneYZ
```

```
Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0#
arrayOfVariantOfDouble1(1) = 0#
arrayOfVariantOfDouble1(2) = 0#
arrayOfVariantOfDouble1(3) = 0#
arrayOfVariantOfDouble1(4) = 1#
arrayOfVariantOfDouble1(5) = 0#
arrayOfVariantOfDouble1(6) = 0#
arrayOfVariantOfDouble1(7) = 0#
arrayOfVariantOfDouble1(8) = 1#
Set sketch1Variant = sketch1
sketch1Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble1
part1.InWorkObject = sketch1

Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()
Dim geometricElements1 As GeometricElements
Set geometricElements1 =
sketch1.GeometricElements
Dim axis2D1 As Axis2D
Set axis2D1 =
geometricElements1.Item("AbsoluteAxis")
Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2
'PAD
Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateClosedCircle(0#,
0#, D / 2)
Dim point2D1 As Point2D
Set point2D1 = axis2D1.GetItem("Origin")
circle2D1.CenterPoint = point2D1
circle2D1.ReportName = 3

Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints
Dim reference2 As Reference
Set reference2 =
part1.CreateReferenceFromObject(circle2D1)
Dim constraint1 As Constraint
Set constraint1 =
constraints1.AddMonoEltCst(catCstTypeRadius,
reference2)
constraint1.Mode = catCstModeDrivingDimension
Dim length1 As Length
Set length1 = constraint1.Dimension
length1.Value = D / 2

sketch1.CloseEdition
part1.InWorkObject = sketch1
part1.UpdateObject sketch1

Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory
Dim pad1 As Pad
Set pad1 = shapeFactory1.AddNewPad(sketch1, B /
3)
pad1.IsSymmetric = True
Dim limit1 As Limit
Set limit1 = pad1.FirstLimit
Dim length2 As Length
Set length2 = limit1.Dimension
length2.Value = B / 3
part1.UpdateObject pad1

Dim reference3 As Reference
Set reference3 =
part1.CreateReferenceFromName("Selection_RSur:(
Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultO
UT;Z0;G3055)")
Dim sketch2 As Sketch
Set sketch2 = sketches1.Add(reference3)
Dim arrayOfVariantOfDouble2(8)
arrayOfVariantOfDouble2(0) = B / 3
arrayOfVariantOfDouble2(1) = 0#
arrayOfVariantOfDouble2(2) = 0#
arrayOfVariantOfDouble2(3) = 0#
arrayOfVariantOfDouble2(4) = 1#
arrayOfVariantOfDouble2(5) = 0#
arrayOfVariantOfDouble2(6) = 0#
arrayOfVariantOfDouble2(7) = 0#
arrayOfVariantOfDouble2(8) = 1#
Set sketch2Variant = sketch2
sketch2Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble2
part1.InWorkObject = sketch2

'POCKET
Dim factory2D2 As Factory2D
Set factory2D2 = sketch2.OpenEdition()
Dim geometricElements2 As GeometricElements
Set geometricElements2 =
sketch2.GeometricElements
Dim axis2D2 As Axis2D
Set axis2D2 =
geometricElements2.Item("AbsoluteAxis")
Dim line2D3 As Line2D
Set line2D3 = axis2D2.GetItem("HDirection")
line2D3.ReportName = 1
Dim line2D4 As Line2D
Set line2D4 = axis2D2.GetItem("VDirection")
line2D4.ReportName = 2

Dim circle2D2 As Circle2D
Set circle2D2 = factory2D2.CreateClosedCircle(0#,
0#, d_ / 2)
Dim point2D2 As Point2D
Set point2D2 = axis2D2.GetItem("Origin")
circle2D2.CenterPoint = point2D2
circle2D2.ReportName = 3

Dim constraints2 As Constraints
Set constraints2 = sketch2.Constraints
Dim reference4 As Reference
Set reference4 =
part1.CreateReferenceFromObject(circle2D2)
Dim constraint2 As Constraint
Set constraint2 =
constraints2.AddMonoEltCst(catCstTypeRadius,
reference4)
constraint2.Mode = catCstModeDrivingDimension
Dim length3 As Length
Set length3 = constraint2.Dimension
length3.Value = d_ / 2
sketch2.CloseEdition
part1.InWorkObject = sketch2
part1.UpdateObject sketch2
```

```
Dim pocket1 As Pocket
Set pocket1 = shapeFactory1.AddNewPocket(sketch2, B / 3)
Dim limit2 As Limit
Set limit2 = pocket1.FirstLimit
limit2.LimitMode = catUpToLastLimit
part1.UpdateObject pocket1

'Huecos para los rodillos
Dim reference5 As Reference
Set reference5 = part1.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pocket.1_ResultOUT;Z0;G3055)")
Dim sketch3 As Sketch
Set sketch3 = sketches1.Add(reference5)

Dim arrayOfVariantOfDouble3(8)
arrayOfVariantOfDouble3(0) = B / 3
arrayOfVariantOfDouble3(1) = 0#
arrayOfVariantOfDouble3(2) = 0#
arrayOfVariantOfDouble3(3) = 0#
arrayOfVariantOfDouble3(4) = 1#
arrayOfVariantOfDouble3(5) = 0#
arrayOfVariantOfDouble3(6) = 0#
arrayOfVariantOfDouble3(7) = 0#
arrayOfVariantOfDouble3(8) = 1#
Set sketch3Variant = sketch3
sketch3Variant.SetAbsoluteAxisData arrayOfVariantOfDouble3
part1.InWorkObject = sketch3

Dim factory2D3 As Factory2D
Set factory2D3 = sketch3.OpenEdition()
Dim geometricElements3 As GeometricElements
Set geometricElements3 = sketch3.GeometricElements
Dim axis2D3 As Axis2D
Set axis2D3 = geometricElements3.Item("AbsoluteAxis")
Dim line2D5 As Line2D
Set line2D5 = axis2D3.GetItem("HDirection")
line2D5.ReportName = 1
Dim line2D6 As Line2D
Set line2D6 = axis2D3.GetItem("VDirection")
line2D6.ReportName = 2

Dim point2D3 As Point2D
Set point2D3 = factory2D3.CreatePoint(-Dw / 2, Eb / 2)
point2D3.ReportName = 3
Dim point2D4 As Point2D
Set point2D4 = factory2D3.CreatePoint(Dw / 2, Eb / 2)
point2D4.ReportName = 4
Dim line2D7 As Line2D
Set line2D7 = factory2D3.CreateLine(-Dw / 2, Eb / 2, Dw / 2, Eb / 2)
line2D7.ReportName = 5
line2D7.StartPoint = point2D3
line2D7.EndPoint = point2D4
Dim point2D5 As Point2D
Set point2D5 = factory2D3.CreatePoint(Dw / 2, Ea / 2)
point2D5.ReportName = 6
Dim line2D8 As Line2D
Set line2D8 = factory2D3.CreateLine(Dw / 2, Eb / 2, Dw / 2, Ea / 2)
line2D8.ReportName = 7
line2D8.EndPoint = point2D4
line2D8.StartPoint = point2D5
Dim point2D6 As Point2D
Set point2D6 = factory2D3.CreatePoint(-Dw / 2, Ea / 2)
point2D6.ReportName = 8
Dim line2D9 As Line2D
Set line2D9 = factory2D3.CreateLine(Dw / 2, Ea / 2, -Dw / 2, Ea / 2)
line2D9.ReportName = 9
line2D9.StartPoint = point2D5
line2D9.EndPoint = point2D6
Dim line2D10 As Line2D
Set line2D10 = factory2D3.CreateLine(-Dw / 2, Ea / 2, -Dw / 2, Eb / 2)
line2D10.ReportName = 10
line2D10.EndPoint = point2D6
line2D10.StartPoint = point2D3

Dim constraints3 As Constraints
Set constraints3 = sketch3.Constraints
Dim reference6 As Reference
Set reference6 = part1.CreateReferenceFromObject(line2D7)
Dim reference7 As Reference
Set reference7 = part1.CreateReferenceFromObject(line2D5)
Dim constraint3 As Constraint
Set constraint3 = constraints3.AddBiEltCst(catCstTypeHorizontality, reference6, reference7)
constraint3.Mode = catCstModeDrivingDimension

Dim reference8 As Reference
Set reference8 = part1.CreateReferenceFromObject(line2D9)
Dim reference9 As Reference
Set reference9 = part1.CreateReferenceFromObject(line2D5)
Dim constraint4 As Constraint
Set constraint4 = constraints3.AddBiEltCst(catCstTypeHorizontality, reference8, reference9)
constraint4.Mode = catCstModeDrivingDimension

Dim reference10 As Reference
Set reference10 = part1.CreateReferenceFromObject(line2D8)
Dim reference11 As Reference
Set reference11 = part1.CreateReferenceFromObject(line2D6)
Dim constraint5 As Constraint
Set constraint5 = constraints3.AddBiEltCst(catCstTypeVerticality, reference10, reference11)
constraint5.Mode = catCstModeDrivingDimension

Dim reference12 As Reference
Set reference12 = part1.CreateReferenceFromObject(line2D10)
Dim reference13 As Reference
Set reference13 = part1.CreateReferenceFromObject(line2D6)
Dim constraint6 As Constraint
```

```
Set constraint6 = constraints3.AddBiEltCst(catCstTypeVerticality, reference12, reference13)
constraint6.Mode = catCstModeDrivingDimension

Dim reference14 As Reference
Set reference14 = part1.CreateReferenceFromObject(line2D8)
Dim reference15 As Reference
Set reference15 = part1.CreateReferenceFromObject(line2D10)
Dim constraint7 As Constraint
Set constraint7 = constraints3.AddBiEltCst(catCstTypeDistance, reference14, reference15)
constraint7.Mode = catCstModeDrivingDimension

Dim length4 As Length
Set length4 = constraint7.Dimension
length4.Value = Dw

Dim reference16 As Reference
Set reference16 = part1.CreateReferenceFromObject(line2D8)
Dim reference17 As Reference
Set reference17 = part1.CreateReferenceFromObject(line2D6)
Dim constraint8 As Constraint
Set constraint8 = constraints3.AddBiEltCst(catCstTypeDistance, reference16, reference17)
constraint8.Mode = catCstModeDrivingDimension

Dim length5 As Length
Set length5 = constraint8.Dimension
length5.Value = Dw / 2

Dim reference18 As Reference
Set reference18 = part1.CreateReferenceFromObject(line2D9)
Dim reference19 As Reference
Set reference19 = part1.CreateReferenceFromObject(line2D5)
Dim constraint9 As Constraint
Set constraint9 = constraints3.AddBiEltCst(catCstTypeDistance, reference18, reference19)
constraint9.Mode = catCstModeDrivingDimension
Dim length6 As Length
Set length6 = constraint9.Dimension
length6.Value = Ea / 2

Dim reference20 As Reference
Set reference20 = part1.CreateReferenceFromObject(line2D7)
Dim reference21 As Reference
Set reference21 = part1.CreateReferenceFromObject(line2D5)
Dim constraint10 As Constraint
Set constraint10 = constraints3.AddBiEltCst(catCstTypeDistance, reference20, reference21)
constraint10.Mode = catCstModeDrivingDimension
Dim length7 As Length
Set length7 = constraint10.Dimension
length7.Value = Eb / 2
sketch3.CloseEdition

part1.InWorkObject = sketch3
part1.UpdateObject sketch3

Dim pocket2 As Pocket
Set pocket2 = shapeFactory1.AddNewPocket(sketch3, B / 2)
Dim limit3 As Limit
Set limit3 = pocket2.FirstLimit
limit3.LimitMode = catUpToLastLimit
part1.UpdateObject pocket2

Dim reference22 As Reference
Set reference22 = part1.CreateReferenceFromName("")
Dim reference23 As Reference
Set reference23 = part1.CreateReferenceFromName("")
Dim circPattern1 As CircPattern
Set circPattern1 = shapeFactory1.AddNewCircPattern(pocket2, 1, 2, 20#, 45#, 1, 1, reference22, reference23, True, 0#, True)
circPattern1.CircularPatternParameters = catInstancesandAngularSpacing

Dim angularRepartition1 As angularRepartition
Set angularRepartition1 = circPattern1.angularRepartition

'calculo del número de agujeros para ello definimos primero el número pi
Const PI = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de los rodillos que en este caso es (D+d)/4
Dim longitud As Double
longitud = 2 * PI * ((D - d_) / 4 + d_ / 2)
'crearemos una variable auxiliar que usara el comando matemático Round para redondear el numero de huecos que tendrá el separador.
'para calcular este número hacemos la hipótesis de que la distancia entre rodillos será de Dw/2
Dim variableaux As Double
variableaux = Math.Round(longitud / (3 / 2 * Dw))
'ya tenemos el numero de huecos del separador
Dim intParam As intParam
Set intParam = angularRepartition1.InstancesCount
intParam.Value = variableaux
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angularRepartition2 As angularRepartition
Set angularRepartition2 = circPattern1.angularRepartition
Dim angle As angle
Set angle = angularRepartition2.AngularSpacing
angle.Value = 360 / variableaux

Dim refrot7 As Reference
Set refrot7 = part1.CreateReferenceFromObject(reference1)
circPattern1.SetRotationAxis refrot7

part1.UpdateObject circPattern1
part1.Update
Dim product1 As Product
Set product1 = partDocument1.GetItem("Part1")
product1.PartNumber = "jaula"
```

```
partDocument1.SaveAs objPath & "\jaula.catpart"
partDocument1.Close

'=====================================
'--------------------------------Rodillo--------------------
'=====================================
Dim documents2 As Documents
Set documents2 = CATIA.Documents
Dim partDocument2 As PartDocument
Set partDocument2 = documents2.Add("Part")
Dim part2 As Part
Set part2 = partDocument2.Part
Dim bodies2 As Bodies
Set bodies2 = part2.Bodies
Dim body2 As Body
Set body2 = bodies2.Item("PartBody")
Dim sketches2 As Sketches
Set sketches2 = body2.Sketches
Dim originElements2 As OriginElements
Set originElements2 = part2.OriginElements
Dim reference24 As Reference
Set reference24 = originElements2.PlaneYZ

Dim sketch4 As Sketch
Set sketch4 = sketches2.Add(reference24)

Dim arrayOfVariantOfDouble4(8)
arrayOfVariantOfDouble4(0) = 0#
arrayOfVariantOfDouble4(1) = 0#
arrayOfVariantOfDouble4(2) = 0
arrayOfVariantOfDouble4(3) = 0#
arrayOfVariantOfDouble4(4) = 1#
arrayOfVariantOfDouble4(5) = 0#
arrayOfVariantOfDouble4(6) = 0#
arrayOfVariantOfDouble4(7) = 0#
arrayOfVariantOfDouble4(8) = 1#
Set sketch4Variant = sketch4
sketch4Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble4
part2.InWorkObject = sketch4

Dim factory2D4 As Factory2D
Set factory2D4 = sketch4.OpenEdition()
Dim geometricElements4 As GeometricElements
Set geometricElements4 = sketch4.GeometricElements

Dim axis2D4 As Axis2D
Set axis2D4 = geometricElements4.Item("AbsoluteAxis")
Dim line2D11 As Line2D
Set line2D11 = axis2D4.GetItem("HDirection")
line2D11.ReportName = 1
Dim line2D12 As Line2D
Set line2D12 = axis2D4.GetItem("VDirection")
line2D12.ReportName = 2

Dim point2D1a As Point2D
Set point2D1a = factory2D4.CreatePoint(0#, Eb / 2)
point2D1a.ReportName = 3
Dim constraints4 As Constraints
Set constraints4 = sketch4.Constraints
Dim reference2a As Reference
Set reference2a = part2.CreateReferenceFromObject(point2D1a)
Dim reference3a As Reference
Set reference3a = part2.CreateReferenceFromObject(line2D12)
Dim constraint1a As Constraint
Set constraint1a = constraints4.AddBiEltCst(catCstTypeOn, reference2a, reference3a)
constraint1a.Mode = catCstModeDrivingDimension

Dim point2D2a As Point2D
Set point2D2a = factory2D4.CreatePoint(Dw / 2, Eb / 2)
point2D2a.ReportName = 4
Dim line2D3a As Line2D
Set line2D3a = factory2D4.CreateLine(0, Eb / 2, Dw / 2, Eb / 2)
line2D3a.ReportName = 5
line2D3a.StartPoint = point2D1a
line2D3a.EndPoint = point2D2a

Dim point2D3a As Point2D
Set point2D3a = factory2D4.CreatePoint(Dw / 2, Ea / 2)
point2D3a.ReportName = 6
Dim line2D4a As Line2D
Set line2D4a = factory2D4.CreateLine(Dw / 2, Eb / 2, Dw / 2, Ea / 2)
line2D4a.ReportName = 7
line2D4a.EndPoint = point2D2a
line2D4a.StartPoint = point2D3a

Dim point2D4a As Point2D
Set point2D4a = factory2D4.CreatePoint(0, Ea / 2)
point2D4a.ReportName = 8
Dim line2D5a As Line2D
Set line2D5a = factory2D4.CreateLine(Dw / 2, Ea / 2, 0, Ea / 2)
line2D5a.ReportName = 9
line2D5a.StartPoint = point2D3a
line2D5a.EndPoint = point2D4a

Dim line2D6a As Line2D
Set line2D6a = factory2D4.CreateLine(0, Ea / 2, 0, Eb / 2)
line2D6a.ReportName = 10
line2D6a.EndPoint = point2D4a
line2D6a.StartPoint = point2D1a

Dim reference4a As Reference
Set reference4a = part2.CreateReferenceFromObject(line2D3a)
Dim reference5a As Reference
Set reference5a = part2.CreateReferenceFromObject(line2D11)
Dim constraint2a As Constraint
Set constraint2a = constraints4.AddBiEltCst(catCstTypeHorizontality, reference4a, reference5a)
constraint2a.Mode = catCstModeDrivingDimension

Dim reference6a As Reference
Set reference6a = part2.CreateReferenceFromObject(line2D5a)
Dim reference7a As Reference
Set reference7a = part2.CreateReferenceFromObject(line2D11)
Dim constraint3a As Constraint
```

```
Set constraint3a = constraints4.AddBiEltCst(catCstTypeHorizontality, reference6a, reference7a)
constraint3a.Mode = catCstModeDrivingDimension

Dim reference8a As Reference
Set reference8a = part2.CreateReferenceFromObject(line2D4a)
Dim reference9a As Reference
Set reference9a = part2.CreateReferenceFromObject(line2D12)
Dim constrainta4 As Constraint
Set constraint4a = constraints4.AddBiEltCst(catCstTypeVerticality, reference8a, reference9a)
constraint4a.Mode = catCstModeDrivingDimension

Dim reference10a As Reference
Set reference10a = part2.CreateReferenceFromObject(line2D6a)
Dim reference11a As Reference
Set reference11a = part2.CreateReferenceFromObject(line2D12)
Dim constraint5a As Constraint
Set constraint5a = constraints4.AddBiEltCst(catCstTypeVerticality, reference10a, reference11a)
constraint5a.Mode = catCstModeDrivingDimension

Dim reference12a As Reference
Set reference12a = part2.CreateReferenceFromObject(line2D5a)
Dim reference13a As Reference
Set reference13a = part2.CreateReferenceFromObject(line2D11)
Dim constraint6a As Constraint
Set constraint6a = constraints4.AddBiEltCst(catCstTypeDistance, reference12a, reference13a)
constraint6a.Mode = catCstModeDrivingDimension
Dim length1a As Length
Set length1a = constraint6a.Dimension
length1a.Value = Ea / 2

Dim reference14a As Reference
Set reference14a = part2.CreateReferenceFromObject(line2D4a)
Dim reference15a As Reference
Set reference15a = part2.CreateReferenceFromObject(line2D12)
Dim constraint7a As Constraint
Set constraint7a = constraints4.AddBiEltCst(catCstTypeDistance, reference14a, reference15a)
constraint7a.Mode = catCstModeDrivingDimension
Dim length2a As Length
Set length2a = constraint7a.Dimension
length2a.Value = Dw / 2

Dim reference16a As Reference
Set reference16a = part2.CreateReferenceFromObject(line2D4a)
Dim constraint8a As Constraint
Set constraint8a = constraints4.AddMonoEltCst(catCstTypeLength, reference16a)
constraint8a.Mode = catCstModeDrivingDimension
Dim length3a As Length
Set length3a = constraint8a.Dimension
length3a.Value = (Eb - Ea) / 2
sketch4.CloseEdition
part2.InWorkObject = sketch4
part2.UpdateObject sketch4

Dim shapeFactory2 As ShapeFactory
Set shapeFactory2 = part2.ShapeFactory
Dim shaft1 As Shaft
Set shaft1 = shapeFactory2.AddNewShaft(sketch4)
Dim reference19a As Reference
Set reference19a = part2.CreateReferenceFromObject(line2D12)
shaft1.RevoluteAxis = reference19a
part2.UpdateObject shaft1

Dim reference1b As Reference
Set reference1b = part2.CreateReferenceFromName("")
Dim constRadEdgeFillet1b As ConstRadEdgeFillet
Set constRadEdgeFillet1b = shapeFactory2.AddNewSolidEdgeFilletWithConstantRadius(reference1b, catTangencyFilletEdgePropagation, 0.5)
Dim reference2b As Reference
Set reference2b = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Shaft.1;0:(Brp:(Sketch.1;7)));None:();Cf11:());Face:(Brp:(Shaft.1;0:(Brp:(Sketch.1;5)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", shaft1)
constRadEdgeFillet1b.AddObjectToFillet reference2b
constRadEdgeFillet1b.EdgePropagation = catTangencyFilletEdgePropagation
Dim reference3b As Reference
Set reference3b = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Shaft.1;0:(Brp:(Sketch.1;9)));None:();Cf11:());Face:(Brp:(Shaft.1;0:(Brp:(Sketch.1;7)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", shaft1)
constRadEdgeFillet1b.AddObjectToFillet reference3b
constRadEdgeFillet1b.EdgePropagation = catTangencyFilletEdgePropagation
part2.UpdateObject constRadEdgeFillet1b

Dim referenceb As Reference
Set referenceb = part2.CreateReferenceFromName("")
Dim referenceb1 As Reference
Set referenceb1 = part2.CreateReferenceFromName("")
Dim circPatternb As CircPattern
Set circPatternb = shapeFactory2.AddNewCircPattern(Nothing, 1, 2, 20#, 45#, 1, 1, referenceb, referenceb1, True, 0#, True)
circPatternb.CircularPatternParameters = catInstancesandAngularSpacing
Dim angularRepartitionb As angularRepartition
Set angularRepartitionb = circPatternb.angularRepartition
```

```
'calculo del número de agujeros para ello definimos primero el número pi
Const PIb = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es (D+d)/4
Dim longitudb As Double
longitudb = 2 * PIb * ((D - d_) / 4 + d_ / 2)
'crearemos una variable auxiliar que usara el comando matemático Round para redondear el numero de huecos que tendrá el separador.
'para calcular este número hacemos la hipótesis de que la distancia entre rodillos será de Dw/2
Dim variableauxb As Double
variableauxb = Math.Round(longitudb / (3 / 2 * Dw))
'ya tenemos el numero de huecos del separador
Dim intParamb As intParam
Set intParamb = angularRepartitionb.InstancesCount
intParamb.Value = variableauxb
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angleb As angle
Set angleb = angularRepartitionb.AngularSpacing
angleb.Value = 360 / variableauxb

circPatternb.SetRotationAxis reference24

part2.Update

Dim product2 As Product
Set product2 = partDocument2.GetItem("Part2")
product2.PartNumber = "rodillos"
partDocument2.SaveAs objPath & "\rodillos.catpart"
partDocument2.Close

'=====================================
'----------------------------------ENSAMBLAJE----
'=====================================
Dim documents5 As Documents
Set documents5 = CATIA.Documents
Dim productDocument1 As ProductDocument
Set productDocument1 = documents5.Add("Product")
Dim product5 As Product
Set product5 = productDocument1.Product
Dim products1 As Products
Set products1 = product5.Products

Dim arrayOfVariantOfBSTR3(0)
arrayOfVariantOfBSTR3(0) = objPath & "\rodillos.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR3, "All"

Dim arrayOfVariantOfBSTR4(0)
arrayOfVariantOfBSTR4(0) = objPath & "\jaula.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR4, "All"

Dim constraints5 As Constraints
Set constraints5 = product5.Connections("CATIAConstraints")
'Fijamos los rodillos a los huecos de la jaula.
Dim referencec1 As Reference
Set referencec1 = product5.CreateReferenceFromName("Product5/rodillos.1/!zx plane")
Dim referencec2 As Reference
Set referencec2 = product5.CreateReferenceFromName("Product5/jaula.1/!zx plane")
Dim constraintc1 As Constraint
Set constraintc1 = constraints5.AddBiEltCst(catCstTypeDistance, referencec1, referencec2)

Dim referencec3 As Reference
Set referencec3 = product5.CreateReferenceFromName("Product5/rodillos.1/!yz plane")
Dim referencec4 As Reference
Set referencec4 = product5.CreateReferenceFromName("Product5/jaula.1/!yz plane")
Dim constraintc2 As Constraint
Set constraintc2 = constraints5.AddBiEltCst(catCstTypeDistance, referencec3, referencec4)

Dim referencec5 As Reference
Set referencec5 = product5.CreateReferenceFromName("Product5/rodillos.1/!xy plane")
Dim referencec6 As Reference
Set referencec6 = product5.CreateReferenceFromName("Product5/jaula.1/!xy plane")
Dim constraintc3 As Constraint
Set constraintc3 = constraints5.AddBiEltCst(catCstTypeDistance, referencec5, referencec6)

product5.Update
productDocument1.SaveAs objPath & "\CoronaRodillos.CATProduct"
End Sub
```

- **Rodamiento axial de rodillos cilíndricos.**

Incluye una corona de rodillos cilíndricos y dos arandelas.

```
'=====================================
'----------------------------------Arandela eje----------
'=====================================

Dim documents3 As Documents
Set documents3 = CATIA.Documents
Dim partDocument3 As PartDocument
Set partDocument3 = documents3.Add("Part")
Dim part3 As Part
Set part3 = partDocument3.Part
Dim bodies3 As Bodies
Set bodies3 = part3.Bodies
Dim body3 As Body
Set body3 = bodies3.Item("PartBody")
Dim sketches3 As Sketches
Set sketches3 = body3.Sketches
Dim originElements3 As OriginElements
Set originElements3 = part3.OriginElements
Dim reference31 As Reference
Set reference31 = originElements3.PlaneYZ
Dim sketch5 As Sketch
```

```
Set sketch5 = sketches3.Add(reference31)

Dim arrayOfVariantOfDouble5(8)
arrayOfVariantOfDouble5(0) = 0#
arrayOfVariantOfDouble5(1) = 0#
arrayOfVariantOfDouble5(2) = 0#
arrayOfVariantOfDouble5(3) = 0#
arrayOfVariantOfDouble5(4) = 1#
arrayOfVariantOfDouble5(5) = 0#
arrayOfVariantOfDouble5(6) = 0#
arrayOfVariantOfDouble5(7) = 0#
arrayOfVariantOfDouble5(8) = 1#
Set sketch5Variant = sketch5
sketch5Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble5
part3.InWorkObject = sketch5

Dim factory2D5 As Factory2D
Set factory2D5 = sketch5.OpenEdition()
Dim geometricElements5 As GeometricElements
Set geometricElements5 =
sketch5.GeometricElements

Dim axis2D5 As Axis2D
Set axis2D5 =
geometricElements5.Item("AbsoluteAxis")
Dim line2D13 As Line2D
Set line2D13 = axis2D5.GetItem("HDirection")
line2D13.ReportName = 1
Dim line2D14 As Line2D
Set line2D14 = axis2D5.GetItem("VDirection")
line2D14.ReportName = 2

Dim circle2D4 As Circle2D
Set circle2D4 = factory2D5.CreateClosedCircle(0#,
0#, d_1 / 2)
Dim point2D8 As Point2D
Set point2D8 = axis2D5.GetItem("Origin")
circle2D4.CenterPoint = point2D8
circle2D4.ReportName = 3

Dim constraints5 As Constraints
Set constraints5 = sketch5.Constraints
Dim reference32 As Reference
Set reference32 =
part3.CreateReferenceFromObject(circle2D4)
Dim constraint12 As Constraint
Set constraint12 =
constraints5.AddMonoEltCst(catCstTypeRadius,
reference32)
constraint12.Mode = catCstModeDrivingDimension
Dim length10 As Length
Set length10 = constraint12.Dimension
length10.Value = d_1 / 2

sketch5.CloseEdition
part3.InWorkObject = sketch5
part3.UpdateObject sketch5

Dim shapeFactory3 As ShapeFactory
Set shapeFactory3 = part3.ShapeFactory
Dim pad3 As Pad
Set pad3 = shapeFactory3.AddNewPad(sketch5, B /
2)
pad3.IsSymmetric = True
Dim limit5 As Limit
Set limit5 = pad3.FirstLimit
Dim length11 As Length
Set length11 = limit5.Dimension
length11.Value = B / 2

part3.UpdateObject pad3

Dim reference33 As Reference
Set reference33 =
part3.CreateReferenceFromName("Selection_RSur:(
Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultO
UT;Z0;G3055)")

Dim sketch6 As Sketch
Set sketch6 = sketches3.Add(reference33)

Dim arrayOfVariantOfDouble6(8)
arrayOfVariantOfDouble6(0) = B / 2
arrayOfVariantOfDouble6(1) = 0#
arrayOfVariantOfDouble6(2) = 0#
arrayOfVariantOfDouble6(3) = 0#
arrayOfVariantOfDouble6(4) = 1#
arrayOfVariantOfDouble6(5) = 0#
arrayOfVariantOfDouble6(6) = 0#
arrayOfVariantOfDouble6(7) = 0#
arrayOfVariantOfDouble6(8) = 1#
Set sketch6Variant = sketch6
sketch6Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble6
part3.InWorkObject = sketch6

Dim factory2D6 As Factory2D
Set factory2D6 = sketch6.OpenEdition()
Dim geometricElements6 As GeometricElements
Set geometricElements6 =
sketch6.GeometricElements

Dim axis2D6 As Axis2D
Set axis2D6 =
geometricElements6.Item("AbsoluteAxis")
Dim line2D15 As Line2D
Set line2D15 = axis2D6.GetItem("HDirection")
line2D15.ReportName = 1
Dim line2D16 As Line2D
Set line2D16 = axis2D6.GetItem("VDirection")
line2D16.ReportName = 2

Dim circle2D5 As Circle2D
Set circle2D5 = factory2D6.CreateClosedCircle(0#,
0#, d_ / 2)
Dim point2D9 As Point2D
Set point2D9 = axis2D6.GetItem("Origin")
circle2D5.CenterPoint = point2D9
circle2D5.ReportName = 3

Dim constraints6 As Constraints
Set constraints6 = sketch6.Constraints
Dim reference34 As Reference
Set reference34 =
part3.CreateReferenceFromObject(circle2D5)
Dim constraint13 As Constraint
Set constraint13 =
constraints6.AddMonoEltCst(catCstTypeRadius,
reference34)
constraint13.Mode = catCstModeDrivingDimension
Dim length12 As Length
Set length12 = constraint13.Dimension
length12.Value = d_ / 2
```

```
sketch6.CloseEdition
part3.InWorkObject = sketch6
part3.UpdateObject sketch6

Dim pocket3 As Pocket
Set pocket3 =
shapeFactory3.AddNewPocket(sketch6, B / 2)
Dim limit6 As Limit
Set limit6 = pocket3.FirstLimit
limit6.LimitMode = catUpToLastLimit
part3.UpdateObject pocket3

Dim reference35 As Reference
Set reference35 =
part3.CreateReferenceFromName("")
Dim constRadEdgeFillet2 As ConstRadEdgeFillet
Set constRadEdgeFillet2 =
shapeFactory3.AddNewSolidEdgeFilletWithConstant
Radius(reference35,
catTangencyFilletEdgePropagation, r)
Dim reference36 As Reference
Set reference36 =
part3.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;1);None:();Cf11:());Face:(Brp:(P
ad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());None:(Li
mits1:();Limits2:());Cf11:());WithTemporaryBody;W
ithoutBuildError;WithSelectingFeatureSupport;MFB
RepVersion_CXR15)", pocket3)
constRadEdgeFillet2.AddObjectToFillet reference36
constRadEdgeFillet2.EdgePropagation =
catTangencyFilletEdgePropagation
Dim reference37 As Reference
Set reference37 =
part3.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();
Cf11:());Face:(Brp:(Pad.1;1);None:();Cf11:());None:(
Limits1:();Limits2:());Cf11:());WithTemporaryBody;
WithoutBuildError;WithSelectingFeatureSupport;MF
BRepVersion_CXR15)", pocket3)
constRadEdgeFillet2.AddObjectToFillet reference37
constRadEdgeFillet2.EdgePropagation =
catTangencyFilletEdgePropagation
part3.UpdateObject constRadEdgeFillet2
part3.Update

Dim product3 As Product
Set product3 = partDocument3.GetItem("Part3")
product3.PartNumber = "arandelaeje"
partDocument3.SaveAs objPath &
"\arandelaeje.catpart"
partDocument3.Close

'====================================
'----------------------------------Arandela de soporte
'====================================
Dim documents4 As Documents
Set documents4 = CATIA.Documents
Dim partDocument4 As PartDocument
Set partDocument4 = documents4.Add("Part")
Dim part4 As Part
Set part4 = partDocument4.Part
Dim bodies4 As Bodies
Set bodies4 = part4.Bodies
Dim body4 As Body
Set body4 = bodies4.Item("PartBody")
Dim sketches4 As Sketches
Set sketches4 = body4.Sketches
Dim originElements4 As OriginElements
Set originElements4 = part4.OriginElements
Dim reference38 As Reference
Set reference38 = originElements4.PlaneYZ
Dim sketch7 As Sketch
Set sketch7 = sketches4.Add(reference38)

Dim arrayOfVariantOfDouble7(8)
arrayOfVariantOfDouble7(0) = 0#
arrayOfVariantOfDouble7(1) = 0#
arrayOfVariantOfDouble7(2) = 0#
arrayOfVariantOfDouble7(3) = 0#
arrayOfVariantOfDouble7(4) = 1#
arrayOfVariantOfDouble7(5) = 0#
arrayOfVariantOfDouble7(6) = 0#
arrayOfVariantOfDouble7(7) = 0#
arrayOfVariantOfDouble7(8) = 1#
Set sketch7Variant = sketch7
sketch7Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble7
part4.InWorkObject = sketch7

Dim factory2D7 As Factory2D
Set factory2D7 = sketch7.OpenEdition()
Dim geometricElements7 As GeometricElements
Set geometricElements7 =
sketch7.GeometricElements

Dim axis2D7 As Axis2D
Set axis2D7 =
geometricElements7.Item("AbsoluteAxis")
Dim line2D17 As Line2D
Set line2D17 = axis2D7.GetItem("HDirection")
line2D17.ReportName = 1
Dim line2D18 As Line2D
Set line2D18 = axis2D7.GetItem("VDirection")
line2D18.ReportName = 2

Dim circle2D6 As Circle2D
Set circle2D6 = factory2D7.CreateClosedCircle(0#,
0#, D / 2)
Dim point2D10 As Point2D
Set point2D10 = axis2D7.GetItem("Origin")
circle2D6.CenterPoint = point2D10
circle2D6.ReportName = 3

Dim constraints7 As Constraints
Set constraints7 = sketch7.Constraints
Dim reference39 As Reference
Set reference39 =
part4.CreateReferenceFromObject(circle2D6)
Dim constraint14 As Constraint
Set constraint14 =
constraints7.AddMonoEltCst(catCstTypeRadius,
reference39)
constraint14.Mode = catCstModeDrivingDimension
Dim length13 As Length
Set length13 = constraint14.Dimension
length13.Value = D / 2

sketch7.CloseEdition
part4.InWorkObject = sketch7
part4.UpdateObject sketch7

Dim shapeFactory4 As ShapeFactory
Set shapeFactory4 = part4.ShapeFactory
```

```
Dim pad4 As Pad
Set pad4 = shapeFactory4.AddNewPad(sketch7, B / 2)
pad4.IsSymmetric = True
Dim limit7 As Limit
Set limit7 = pad4.FirstLimit
Dim length14 As Length
Set length14 = limit7.Dimension
length14.Value = B / 2
part4.UpdateObject pad4

Dim reference40 As Reference
Set reference40 = part4.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")

Dim sketch8 As Sketch
Set sketch8 = sketches4.Add(reference40)
Dim arrayOfVariantOfDouble8(8)
arrayOfVariantOfDouble8(0) = B / 2
arrayOfVariantOfDouble8(1) = 0#
arrayOfVariantOfDouble8(2) = 0#
arrayOfVariantOfDouble8(3) = 0#
arrayOfVariantOfDouble8(4) = 1#
arrayOfVariantOfDouble8(5) = 0#
arrayOfVariantOfDouble8(6) = 0#
arrayOfVariantOfDouble8(7) = 0#
arrayOfVariantOfDouble8(8) = 1#
Set sketch8Variant = sketch8
sketch8Variant.SetAbsoluteAxisData arrayOfVariantOfDouble8
part4.InWorkObject = sketch8

Dim factory2D8 As Factory2D
Set factory2D8 = sketch8.OpenEdition()
Dim geometricElements8 As GeometricElements
Set geometricElements8 = sketch8.GeometricElements

Dim axis2D8 As Axis2D
Set axis2D8 = geometricElements8.Item("AbsoluteAxis")
Dim line2D19 As Line2D
Set line2D19 = axis2D8.GetItem("HDirection")
line2D19.ReportName = 1
Dim line2D20 As Line2D
Set line2D20 = axis2D8.GetItem("VDirection")
line2D20.ReportName = 2

Dim circle2D7 As Circle2D
Set circle2D7 = factory2D8.CreateClosedCircle(0#, 0#, D1 / 2)
Dim point2D11 As Point2D
Set point2D11 = axis2D8.GetItem("Origin")
circle2D7.CenterPoint = point2D11
circle2D7.ReportName = 3

Dim constraints8 As Constraints
Set constraints8 = sketch8.Constraints
Dim reference41 As Reference
Set reference41 = part4.CreateReferenceFromObject(circle2D7)
Dim constraint15 As Constraint
Set constraint15 = constraints8.AddMonoEltCst(catCstTypeRadius, reference41)
constraint15.Mode = catCstModeDrivingDimension
Dim length15 As Length
Set length15 = constraint15.Dimension
length15.Value = D1 / 2
sketch8.CloseEdition
part4.InWorkObject = sketch8
part4.UpdateObject sketch8

Dim pocket4 As Pocket
Set pocket4 = shapeFactory4.AddNewPocket(sketch8, B / 2)
Dim limit8 As Limit
Set limit8 = pocket4.FirstLimit
limit8.LimitMode = catUpToLastLimit
part4.UpdateObject pocket4

Dim reference42 As Reference
Set reference42 = part4.CreateReferenceFromName("")
Dim constRadEdgeFillet3 As ConstRadEdgeFillet
Set constRadEdgeFillet3 = shapeFactory4.AddNewSolidEdgeFilletWithConstantRadius(reference42, catTangencyFilletEdgePropagation, r)
Dim reference43 As Reference
Set reference43 = part4.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());Face:(Brp:(Pad.1;2);None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket4)
constRadEdgeFillet3.AddObjectToFillet reference43
constRadEdgeFillet3.EdgePropagation = catTangencyFilletEdgePropagation
Dim reference44 As Reference
Set reference44 = part4.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;2);None:();Cf11:());Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket4)
constRadEdgeFillet3.AddObjectToFillet reference44
constRadEdgeFillet3.EdgePropagation = catTangencyFilletEdgePropagation
part4.UpdateObject constRadEdgeFillet3
part4.Update

Dim product4 As Product
Set product4 = partDocument4.GetItem("Part4")
product4.PartNumber = "arandelasoporte"
partDocument4.SaveAs objPath & "\arandelasoporte.catpart"
partDocument4.Close
'====================================
'----------------------------------ENSAMBLAJE---------
'====================================
Dim documents5 As Documents
Set documents5 = CATIA.Documents
Dim productDocument1 As ProductDocument
Set productDocument1 = documents5.Add("Product")
Dim product5 As Product
Set product5 = productDocument1.Product

Dim products1 As Products
```

```
Set products1 = product5.Products

Dim arrayOfVariantOfBSTR1(0)
arrayOfVariantOfBSTR1(0) = objPath &
"\arandelasoporte.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR1, "All"

Dim constraints9 As Constraints
Set constraints9 =
product5.Connections("CATIAConstraints")
'fijamos el aro1
Dim reference45 As Reference
Set reference45 =
product5.CreateReferenceFromName("Product5/aran
delasoporte.1/!Product5/arandelasoporte.1/")
Dim constraint16 As Constraint
Set constraint16 =
constraints9.AddMonoEltCst(catCstTypeReference,
reference45)

Dim arrayOfVariantOfBSTR2(0)
arrayOfVariantOfBSTR2(0) = objPath &
"\arandelaeje.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR2, "All"

Dim reference46 As Reference
Set reference46 =
product5.CreateReferenceFromName("Product5/aran
delasoporte.1/!Axis:(Selection_RSur:(Face:(Brp:(Poc
ket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());EdgeFill
et.1_ResultOUT;Z0;G3055))")
Dim reference47 As Reference
Set reference47 =
product5.CreateReferenceFromName("Product5/aran
delaeje.1/!Axis:(Selection_RSur:(Face:(Brp:(Pocket.
1;0:(Brp:(Sketch.2;3)));None:();Cf11:());EdgeFillet.1
_ResultOUT;Z0;G3055))")
Dim constraint17 As Constraint
Set constraint17 =
constraints9.AddBiEltCst(catCstTypeOn,
reference46, reference47)

Dim reference48 As Reference
Set reference48 =
product5.CreateReferenceFromName("Product5/aran
delasoporte.1/!Selection_RSur:(Face:(Brp:(Pad.1;2);
None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055
)")
Dim reference49 As Reference
Set reference49 =
product5.CreateReferenceFromName("Product5/aran
delaeje.1/!Selection_RSur:(Face:(Brp:(Pad.1;1);None
:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055)")
Dim constraint18 As Constraint
Set constraint18 =
constraints9.AddBiEltCst(catCstTypeDistance,
reference48, reference49)
Dim length16 As Length
Set length16 = constraint18.Dimension
length16.Value = -H
constraint18.Orientation = catCstOrientOpposite

Dim arrayOfVariantOfBSTR3(0)
arrayOfVariantOfBSTR3(0) = objPath &
"\rodillos.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR3, "All"

Dim arrayOfVariantOfBSTR4(0)
arrayOfVariantOfBSTR4(0) = objPath &
"\jaula.catpart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles
arrayOfVariantOfBSTR4, "All"

Dim reference50 As Reference
Set reference50 =
product5.CreateReferenceFromName("Product5/aran
delasoporte.1/!Axis:(Selection_RSur:(Face:(Brp:(Poc
ket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());EdgeFill
et.1_ResultOUT;Z0;G3055))")
Dim reference51 As Reference
Set reference51 =
product5.CreateReferenceFromName("Product5/jaula
.1/!Axis:(Selection_RSur:(Face:(Brp:(Pocket.1;0:(Br
p:(Sketch.2;3)));None:();Cf11:());CircPattern.1_Resu
ltOUT;Z0;G3055))")
Dim constraint19 As Constraint
Set constraint19 =
constraints9.AddBiEltCst(catCstTypeOn,
reference50, reference51)

Dim reference52 As Reference
Set reference52 =
product5.CreateReferenceFromName("Product5/aran
delasoporte.1/!Selection_RSur:(Face:(Brp:(Pad.1;2);
None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055
)")
Dim reference53 As Reference
Set reference53 =
product5.CreateReferenceFromName("Product5/jaula
.1/!yz plane")
Dim constraint20 As Constraint
Set constraint20 =
constraints9.AddBiEltCst(catCstTypeDistance,
reference52, reference53)
Dim length17 As Length
Set length17 = constraint20.Dimension
length17.Value = -H / 2
constraint20.Orientation = catCstOrientSame
```

- **Rodamiento de agujas sin aro interno.**

```
Private Sub CommandButton1_Click()
'====================================
'----------------------------------Anillo externo----------
'====================================
Dim documents1 As Documents
Set documents1 = CATIA.Documents
Dim partDocument1 As PartDocument
Set partDocument1 = documents1.Add("Part")
Dim part1 As Part
Set part1 = partDocument1.Part
Dim bodies1 As Bodies
Set bodies1 = part1.Bodies
Dim body1 As Body
Set body1 = bodies1.Item("PartBody")
Dim sketches1 As Sketches
Set sketches1 = body1.Sketches
Dim originElements1 As OriginElements
```

```
Set originElements1 = part1.OriginElements
Dim reference1 As Reference
Set reference1 = originElements1.PlaneYZ
Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0#
arrayOfVariantOfDouble1(1) = 0#
arrayOfVariantOfDouble1(2) = 0#
arrayOfVariantOfDouble1(3) = 0#
arrayOfVariantOfDouble1(4) = 1#
arrayOfVariantOfDouble1(5) = 0#
arrayOfVariantOfDouble1(6) = 0#
arrayOfVariantOfDouble1(7) = 0#
arrayOfVariantOfDouble1(8) = 1#
Set sketch1Variant = sketch1
sketch1Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble1
part1.InWorkObject = sketch1

Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()
Dim geometricElements1 As GeometricElements
Set geometricElements1 =
sketch1.GeometricElements
Dim axis2D1 As Axis2D
Set axis2D1 =
geometricElements1.Item("AbsoluteAxis")
Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2

Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateClosedCircle(0#,
0#, D / 2)
Dim point2D1 As Point2D
Set point2D1 = axis2D1.GetItem("Origin")
circle2D1.CenterPoint = point2D1
circle2D1.ReportName = 3

Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints
Dim reference2 As Reference
Set reference2 =
part1.CreateReferenceFromObject(circle2D1)
Dim constraint1 As Constraint
Set constraint1 =
constraints1.AddMonoEltCst(catCstTypeRadius,
reference2)
constraint1.Mode = catCstModeDrivingDimension
Dim length1 As Length
Set length1 = constraint1.Dimension
length1.Value = D / 2

sketch1.CloseEdition
part1.InWorkObject = sketch1
part1.Update

'PAD
Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory
Dim pad1 As Pad
Set pad1 = shapeFactory1.AddNewPad(sketch1, B /
2)
pad1.IsSymmetric = True
Dim limit1 As Limit
Set limit1 = pad1.FirstLimit
Dim length2 As Length
Set length2 = limit1.Dimension
length2.Value = B / 2
part1.Update

'CHAFLÁN
Dim reference3 As Reference
Set reference3 =
part1.CreateReferenceFromName("")
Dim constRadEdgeFillet1 As ConstRadEdgeFillet
Set constRadEdgeFillet1 =
shapeFactory1.AddNewSolidEdgeFilletWithConstant
Radius(reference3,
catTangencyFilletEdgePropagation, r)

Dim reference4 As Reference
Set reference4 =
part1.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf
11:());Face:(Brp:(Pad.1;2);None:();Cf11:());None:(Li
mits1:();Limits2:());Cf11:());WithTemporaryBody;W
ithoutBuildError;WithSelectingFeatureSupport;MFB
RepVersion_CXR15)", pad1)
constRadEdgeFillet1.AddObjectToFillet reference4
constRadEdgeFillet1.EdgePropagation =
catTangencyFilletEdgePropagation
Dim reference5 As Reference
Set reference5 =
part1.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;1);None:();Cf11:());Face:(Brp:(P
ad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());None:(Li
mits1:();Limits2:());Cf11:());WithTemporaryBody;W
ithoutBuildError;WithSelectingFeatureSupport;MFB
RepVersion_CXR15)", pad1)
constRadEdgeFillet1.AddObjectToFillet reference5
constRadEdgeFillet1.EdgePropagation =
catTangencyFilletEdgePropagation

part1.Update

'POCKET
Dim reference6 As Reference
Set reference6 =
part1.CreateReferenceFromName("Selection_RSur:(
Face:(Brp:(Pad.1;2);None:();Cf11:());EdgeFillet.1_R
esultOUT;Z0;G3055)")

Dim sketch2 As Sketch
Set sketch2 = sketches1.Add(reference6)

Dim arrayOfVariantOfDouble2(8)
arrayOfVariantOfDouble2(0) = B / 2
arrayOfVariantOfDouble2(1) = 0#
arrayOfVariantOfDouble2(2) = 0#
arrayOfVariantOfDouble2(3) = 0#
arrayOfVariantOfDouble2(4) = 1#
arrayOfVariantOfDouble2(5) = 0#
arrayOfVariantOfDouble2(6) = 0#
arrayOfVariantOfDouble2(7) = 0#
arrayOfVariantOfDouble2(8) = 1#
Set sketch2Variant = sketch2
```

```
sketch2Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble2
part1.InWorkObject = sketch2

Dim factory2D2 As Factory2D
Set factory2D2 = sketch2.OpenEdition()
Dim geometricElements2 As GeometricElements
Set geometricElements2 =
sketch2.GeometricElements
Dim axis2D2 As Axis2D
Set axis2D2 =
geometricElements2.Item("AbsoluteAxis")
Dim line2D3 As Line2D
Set line2D3 = axis2D2.GetItem("HDirection")
line2D3.ReportName = 1
Dim line2D4 As Line2D
Set line2D4 = axis2D2.GetItem("VDirection")
line2D4.ReportName = 2

Dim circle2D2 As Circle2D
Set circle2D2 = factory2D2.CreateClosedCircle(0#,
0#, D1 / 2)
Dim point2D2 As Point2D
Set point2D2 = axis2D2.GetItem("Origin")
circle2D2.CenterPoint = point2D2
circle2D2.ReportName = 3

Dim constraints2 As Constraints
Set constraints2 = sketch2.Constraints
Dim reference7 As Reference
Set reference7 =
part1.CreateReferenceFromObject(circle2D2)
Dim constraint2 As Constraint
Set constraint2 =
constraints2.AddMonoEltCst(catCstTypeRadius,
reference7)
constraint2.Mode = catCstModeDrivingDimension
Dim length4 As Length
Set length4 = constraint2.Dimension
length4.Value = D1 / 2

sketch2.CloseEdition
part1.InWorkObject = sketch2
part1.Update

Dim pocket1 As Pocket
Set pocket1 =
shapeFactory1.AddNewPocket(sketch2, B / 2)
Dim limit2 As Limit
Set limit2 = pocket1.FirstLimit
limit2.LimitMode = catUpToLastLimit

part1.Update

'SLOT para el canal de los rodillos
Dim reference8 As Reference
Set reference8 = originElements1.PlaneZX
Dim sketch3 As Sketch
Set sketch3 = sketches1.Add(reference8)

Dim arrayOfVariantOfDouble3(8)
arrayOfVariantOfDouble3(0) = 0#
arrayOfVariantOfDouble3(1) = 0#
arrayOfVariantOfDouble3(2) = 0#
arrayOfVariantOfDouble3(3) = -1#
arrayOfVariantOfDouble3(4) = 0#
arrayOfVariantOfDouble3(5) = 0#
arrayOfVariantOfDouble3(6) = 0#
arrayOfVariantOfDouble3(7) = -0#
arrayOfVariantOfDouble3(8) = 1#
Set sketch3Variant = sketch3
sketch3Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble3
part1.InWorkObject = sketch3

Dim factory2D3 As Factory2D
Set factory2D3 = sketch3.OpenEdition()
Dim geometricElements3 As GeometricElements
Set geometricElements3 =
sketch3.GeometricElements
Dim axis2D3 As Axis2D
Set axis2D3 =
geometricElements3.Item("AbsoluteAxis")
Dim line2D5 As Line2D
Set line2D5 = axis2D3.GetItem("HDirection")
line2D5.ReportName = 1
Dim line2D6 As Line2D
Set line2D6 = axis2D3.GetItem("VDirection")
line2D6.ReportName = 2

'dibujamos un rectangulo
Dim point2D3 As Point2D
Set point2D3 = factory2D3.CreatePoint(-0.4 * B, F /
2)
point2D3.ReportName = 3

Dim point2D4 As Point2D
Set point2D4 = factory2D3.CreatePoint(0.4 * B, F /
2)
point2D4.ReportName = 4

Dim line2D7 As Line2D
Set line2D7 = factory2D3.CreateLine(-0.4 * B, F / 2,
0.4 * B, F / 2)
line2D7.ReportName = 5
line2D7.StartPoint = point2D3
line2D7.EndPoint = point2D4

Dim constraints3 As Constraints
Set constraints3 = sketch3.Constraints

Dim reference9 As Reference
Set reference9 =
part1.CreateReferenceFromObject(line2D7)
Dim reference10 As Reference
Set reference10 =
part1.CreateReferenceFromObject(line2D5)
Dim constraint3 As Constraint
Set constraint3 =
constraints3.AddBiEltCst(catCstTypeHorizontality,
reference9, reference10)
constraint3.Mode = catCstModeDrivingDimension

Dim reference11 As Reference
Set reference11 =
part1.CreateReferenceFromObject(line2D7)
Dim constraint4 As Constraint
Set constraint4 =
constraints3.AddMonoEltCst(catCstTypeLength,
reference11)
constraint4.Mode = catCstModeDrivingDimension
Dim length5 As Length
Set length5 = constraint4.Dimension
length5.Value = 0.8 * B
```

```
Dim point2D5 As Point2D
Set point2D5 = factory2D3.CreatePoint(0.4 * B, F / 2
+ Da)
point2D5.ReportName = 6

Dim line2D8 As Line2D
Set line2D8 = factory2D3.CreateLine(0.4 * B, F / 2,
0.4 * B, F / 2 + Da)
line2D8.ReportName = 7
line2D8.StartPoint = point2D4
line2D8.EndPoint = point2D5

Dim reference12 As Reference
Set reference12 =
part1.CreateReferenceFromObject(line2D8)
Dim reference13 As Reference
Set reference13 =
part1.CreateReferenceFromObject(line2D6)
Dim constraint5 As Constraint
Set constraint5 =
constraints3.AddBiEltCst(catCstTypeVerticality,
reference12, reference13)
constraint5.Mode = catCstModeDrivingDimension

Dim reference14 As Reference
Set reference14 =
part1.CreateReferenceFromObject(line2D8)
Dim constraint6 As Constraint
Set constraint6 =
constraints3.AddMonoEltCst(catCstTypeLength,
reference14)
constraint6.Mode = catCstModeDrivingDimension
Dim length6 As Length
Set length6 = constraint6.Dimension
length6.Value = Da

Dim point2D6 As Point2D
Set point2D6 = factory2D3.CreatePoint(-0.4 * B, F /
2 + Da)
point2D6.ReportName = 8

Dim line2D9 As Line2D
Set line2D9 = factory2D3.CreateLine(0.4 * B, F / 2 +
Da, -0.4 * B, F / 2 + Da)
line2D9.ReportName = 9
line2D9.StartPoint = point2D5
line2D9.EndPoint = point2D6

Dim reference15 As Reference
Set reference15 =
part1.CreateReferenceFromObject(line2D9)
Dim reference16 As Reference
Set reference16 =
part1.CreateReferenceFromObject(line2D5)
Dim constraint7 As Constraint
Set constraint7 =
constraints3.AddBiEltCst(catCstTypeHorizontality,
reference15, reference16)
constraint7.Mode = catCstModeDrivingDimension

Dim line2D10 As Line2D
Set line2D10 = factory2D3.CreateLine(-0.4 * B, F / 2
+ Da, -0.4 * B, F / 2)
line2D10.ReportName = 10
line2D10.StartPoint = point2D6
line2D10.EndPoint = point2D3

Dim reference17 As Reference
Set reference17 =
part1.CreateReferenceFromObject(line2D10)
Dim reference18 As Reference
Set reference18 =
part1.CreateReferenceFromObject(line2D6)
Dim constraint8 As Constraint
Set constraint8 =
constraints3.AddBiEltCst(catCstTypeVerticality,
reference17, reference18)
constraint8.Mode = catCstModeDrivingDimension

Dim reference19 As Reference
Set reference19 =
part1.CreateReferenceFromObject(line2D7)
Dim reference20 As Reference
Set reference20 =
part1.CreateReferenceFromObject(line2D5)
Dim constraint9 As Constraint
Set constraint9 =
constraints3.AddBiEltCst(catCstTypeDistance,
reference19, reference20)
constraint9.Mode = catCstModeDrivingDimension
Dim length7 As Length
Set length7 = constraint9.Dimension
length7.Value = F / 2

Dim reference21 As Reference
Set reference21 =
part1.CreateReferenceFromObject(line2D8)
Dim reference22 As Reference
Set reference22 =
part1.CreateReferenceFromObject(line2D6)
Dim constraint10 As Constraint
Set constraint10 =
constraints3.AddBiEltCst(catCstTypeDistance,
reference21, reference22)
constraint10.Mode = catCstModeDrivingDimension
Dim length8 As Length
Set length8 = constraint10.Dimension
length8.Value = 0.4 * B

sketch3.CloseEdition
part1.InWorkObject = sketch3
part1.Update
Dim sketch4 As Sketch
Set sketch4 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble4(8)
arrayOfVariantOfDouble4(0) = 0#
arrayOfVariantOfDouble4(1) = 0#
arrayOfVariantOfDouble4(2) = 0#
arrayOfVariantOfDouble4(3) = 0#
arrayOfVariantOfDouble4(4) = 1#
arrayOfVariantOfDouble4(5) = 0#
arrayOfVariantOfDouble4(6) = 0#
arrayOfVariantOfDouble4(7) = 0#
arrayOfVariantOfDouble4(8) = 1#
Set sketch4Variant = sketch4
sketch4Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble4
part1.InWorkObject = sketch4

Dim factory2D4 As Factory2D
Set factory2D4 = sketch4.OpenEdition()
Dim geometricElements4 As GeometricElements
```

```
Set geometricElements4 = sketch4.GeometricElements
Dim axis2D4 As Axis2D
Set axis2D4 = geometricElements4.Item("AbsoluteAxis")
Dim line2D11 As Line2D
Set line2D11 = axis2D4.GetItem("HDirection")
line2D11.ReportName = 1
Dim line2D12 As Line2D
Set line2D12 = axis2D4.GetItem("VDirection")
line2D12.ReportName = 2

Dim circle2D3 As Circle2D
Set circle2D3 = factory2D4.CreateClosedCircle(0#, 0#, F / 2 + Da / 2)
Dim point2D7 As Point2D
Set point2D7 = axis2D4.GetItem("Origin")
circle2D3.CenterPoint = point2D7
circle2D3.ReportName = 3


Dim constraints4 As Constraints
Set constraints4 = sketch4.Constraints

Dim reference23 As Reference
Set reference23 = part1.CreateReferenceFromObject(circle2D3)
Dim constraint11 As Constraint
Set constraint11 = constraints4.AddMonoEltCst(catCstTypeRadius, reference23)
constraint11.Mode = catCstModeDrivingDimension
Dim length9 As Length
Set length9 = constraint11.Dimension
length9.Value = F / 2 + Da / 2

sketch4.CloseEdition
part1.InWorkObject = sketch4
part1.Update

Dim slot1 As Slot
Set slot1 = shapeFactory1.AddNewSlotFromRef(Nothing, Nothing)
Dim reference24 As Reference
Set reference24 = part1.CreateReferenceFromObject(sketch3)
slot1.SetProfileElement reference24
Dim reference25 As Reference
Set reference25 = part1.CreateReferenceFromObject(sketch4)
slot1.CenterCurveElement = reference25
part1.Update

Dim product1 As Product
Set product1 = partDocument1.GetItem("Part1")
product1.PartNumber = "aroexterno"
partDocument1.SaveAs objPath & "\aroext.catpart"
partDocument1.Close

'=====================================
'----------------------------------Aguja----------------
'=====================================
Dim documents2 As Documents
Set documents2 = CATIA.Documents
Dim partDocument2 As PartDocument
Set partDocument2 = documents2.Add("Part")
Dim part2 As Part
Set part2 = partDocument2.Part
Dim bodies2 As Bodies
Set bodies2 = part2.Bodies
Dim body2 As Body
Set body2 = bodies2.Item("PartBody")
Dim sketches2 As Sketches
Set sketches2 = body2.Sketches
Dim originElements2 As OriginElements
Set originElements2 = part2.OriginElements
Dim reference26 As Reference
Set reference26 = originElements2.PlaneYZ
Dim sketch5 As Sketch
Set sketch5 = sketches2.Add(reference26)

Dim arrayOfVariantOfDouble5(8)
arrayOfVariantOfDouble5(0) = 0#
arrayOfVariantOfDouble5(1) = 0#
arrayOfVariantOfDouble5(2) = 0#
arrayOfVariantOfDouble5(3) = 0#
arrayOfVariantOfDouble5(4) = 1#
arrayOfVariantOfDouble5(5) = 0#
arrayOfVariantOfDouble5(6) = 0#
arrayOfVariantOfDouble5(7) = 0#
arrayOfVariantOfDouble5(8) = 1#
Set sketch5Variant = sketch5
sketch5Variant.SetAbsoluteAxisData arrayOfVariantOfDouble5

part2.InWorkObject = sketch5

Dim factory2D5 As Factory2D
Set factory2D5 = sketch5.OpenEdition()
Dim geometricElements5 As GeometricElements
Set geometricElements5 = sketch5.GeometricElements

Dim axis2D5 As Axis2D
Set axis2D5 = geometricElements5.Item("AbsoluteAxis")
Dim line2D13 As Line2D
Set line2D13 = axis2D5.GetItem("HDirection")
line2D13.ReportName = 1
Dim line2D14 As Line2D
Set line2D14 = axis2D5.GetItem("VDirection")
line2D14.ReportName = 2

Dim point2D8 As Point2D
Set point2D8 = factory2D5.CreatePoint(0#, F / 2 + Da / 2)
point2D8.ReportName = 3
Dim circle2D4 As Circle2D
Set circle2D4 = factory2D5.CreateClosedCircle(0#, F / 2 + Da / 2, Da / 2)
circle2D4.CenterPoint = point2D8
circle2D4.ReportName = 4

Dim constraints5 As Constraints
Set constraints5 = sketch5.Constraints
Dim reference27 As Reference
Set reference27 = part2.CreateReferenceFromObject(point2D8)
Dim reference28 As Reference
Set reference28 = part2.CreateReferenceFromObject(line2D14)
Dim constraint12 As Constraint
```

```
Set constraint12 = constraints5.AddBiEltCst(catCstTypeOn, reference27, reference28)
constraint12.Mode = catCstModeDrivingDimension
Dim reference29 As Reference
Set reference29 = part2.CreateReferenceFromObject(circle2D4)
Dim constraint13 As Constraint
Set constraint13 = constraints5.AddMonoEltCst(catCstTypeRadius, reference29)
constraint13.Mode = catCstModeDrivingDimension
Dim length10 As Length
Set length10 = constraint13.Dimension
length10.Value = Da / 2

Dim reference30 As Reference
Set reference30 = part2.CreateReferenceFromObject(line2D13)
Dim reference31 As Reference
Set reference31 = part2.CreateReferenceFromObject(point2D8)
Dim constraint14 As Constraint
Set constraint14 = constraints5.AddBiEltCst(catCstTypeDistance, reference30, reference31)
constraint14.Mode = catCstModeDrivingDimension
Dim length11 As Length
Set length11 = constraint14.Dimension
length11.Value = F / 2 + Da / 2

sketch5.CloseEdition
part2.InWorkObject = sketch5
part2.Update
Dim shapeFactory2 As ShapeFactory
Set shapeFactory2 = part2.ShapeFactory

Dim pad2 As Pad
Set pad2 = shapeFactory2.AddNewPad(sketch5, 0.4 * B)
Dim limit3 As Limit
Set limit3 = pad2.FirstLimit
Dim length12 As Length
Set length12 = limit3.Dimension
length12.Value = 0.4 * B
pad2.IsSymmetric = True

Dim referenceb As Reference
Set referenceb = part2.CreateReferenceFromName("")
Dim referenceb1 As Reference
Set referenceb1 = part2.CreateReferenceFromName("")
Dim circPatternb As CircPattern
Set circPatternb = shapeFactory2.AddNewCircPattern(Nothing, 1, 2, 20#, 45#, 1, 1, referenceb, referenceb1, True, 0#, True)
circPatternb.CircularPatternParameters = catInstancesandAngularSpacing
Dim angularRepartitionb As angularRepartition
Set angularRepartitionb = circPatternb.angularRepartition

'calculo del número de agujeros para ello definimos primero el número pi
Const PIb = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es F/2+Da/2
Dim longitudb As Double
longitudb = 2 * PIb * (F / 2 + Da / 2)
'crearemos una variable auxiliar que usara el comando matemático Round para redondear el numero de huecos que tendrá el separador.
'para calcular este número hacemos la hipótesis de que la distancia entre agujas será del diámetro de las agujas
Dim variableauxb As Double
variableauxb = Math.Round(longitudb / (2 * Da))
'ya tenemos el numero de huecos del separador
Dim intParamb As intParam
Set intParamb = angularRepartitionb.InstancesCount
intParamb.Value = variableauxb
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angleb As angle
Set angleb = angularRepartitionb.AngularSpacing
angleb.Value = 360 / variableauxb

circPatternb.SetRotationAxis reference26
part2.Update

Dim product2 As Product
Set product2 = partDocument2.GetItem("Part2")
product2.PartNumber = "agujas"
partDocument2.SaveAs objPath & "\agujas.CATPart"
partDocument2.Close
'=====================================
'----------Separador----------
'=====================================
Dim documents3 As Documents
Set documents3 = CATIA.Documents
Dim partDocument3 As PartDocument
Set partDocument3 = documents3.Add("Part")
Dim part3 As Part
Set part3 = partDocument3.Part
Dim bodies3 As Bodies
Set bodies3 = part3.Bodies
Dim body3 As Body
Set body3 = bodies3.Item("PartBody")
Dim sketches3 As Sketches
Set sketches3 = body3.Sketches
Dim originElements3 As OriginElements
Set originElements3 = part3.OriginElements
Dim reference32 As Reference
Set reference32 = originElements3.PlaneYZ
Dim sketch6 As Sketch
Set sketch6 = sketches3.Add(reference32)

Dim arrayOfVariantOfDouble6(8)
arrayOfVariantOfDouble6(0) = 0#
arrayOfVariantOfDouble6(1) = 0#
arrayOfVariantOfDouble6(2) = 0#
arrayOfVariantOfDouble6(3) = 0#
arrayOfVariantOfDouble6(4) = 1#
arrayOfVariantOfDouble6(5) = 0#
arrayOfVariantOfDouble6(6) = 0#
arrayOfVariantOfDouble6(7) = 0#
arrayOfVariantOfDouble6(8) = 1#
Set sketch6Variant = sketch6
sketch6Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble6
part3.InWorkObject = sketch6
```

```
'PAD
Dim factory2D6 As Factory2D
Set factory2D6 = sketch6.OpenEdition()
Dim geometricElements6 As GeometricElements
Set geometricElements6 = sketch6.GeometricElements

Dim axis2D6 As Axis2D
Set axis2D6 = geometricElements6.Item("AbsoluteAxis")
Dim line2D15 As Line2D
Set line2D15 = axis2D6.GetItem("HDirection")
line2D15.ReportName = 1
Dim line2D16 As Line2D
Set line2D16 = axis2D6.GetItem("VDirection")
line2D16.ReportName = 2

Dim circle2D5 As Circle2D
Set circle2D5 = factory2D6.CreateClosedCircle(0#, 0#, D1 / 2 - r / 5)
Dim point2D9 As Point2D
Set point2D9 = axis2D6.GetItem("Origin")
circle2D5.CenterPoint = point2D9
circle2D5.ReportName = 3

Dim constraints6 As Constraints
Set constraints6 = sketch6.Constraints

Dim reference33 As Reference
Set reference33 = part3.CreateReferenceFromObject(circle2D5)
Dim constraint15 As Constraint
Set constraint15 = constraints6.AddMonoEltCst(catCstTypeRadius, reference33)
constraint15.Mode = catCstModeDrivingDimension
Dim length13 As Length
Set length13 = constraint15.Dimension
length13.Value = D1 / 2 - r / 5

sketch6.CloseEdition
part3.InWorkObject = sketch6
part3.Update

Dim shapeFactory3 As ShapeFactory
Set shapeFactory3 = part3.ShapeFactory

Dim pad3 As Pad
Set pad3 = shapeFactory3.AddNewPad(sketch6, B / 2 - r)
Dim limit4 As Limit
Set limit4 = pad3.FirstLimit
Dim length14 As Length
Set length14 = limit4.Dimension
length14.Value = B / 2 - r
pad3.IsSymmetric = True
part3.Update

'POCKET
Dim reference34 As Reference
Set reference34 = part3.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")

Dim sketch7 As Sketch
Set sketch7 = sketches3.Add(reference34)

Dim arrayOfVariantOfDouble7(8)
arrayOfVariantOfDouble7(0) = B / 2 - r
arrayOfVariantOfDouble7(1) = 0#
arrayOfVariantOfDouble7(2) = 0#
arrayOfVariantOfDouble7(3) = 0#
arrayOfVariantOfDouble7(4) = 1#
arrayOfVariantOfDouble7(5) = 0#
arrayOfVariantOfDouble7(6) = 0#
arrayOfVariantOfDouble7(7) = 0#
arrayOfVariantOfDouble7(8) = 1#
Set sketch7Variant = sketch7
sketch7Variant.SetAbsoluteAxisData arrayOfVariantOfDouble7

part3.InWorkObject = sketch7

Dim factory2D7 As Factory2D
Set factory2D7 = sketch7.OpenEdition()
Dim geometricElements7 As GeometricElements
Set geometricElements7 = sketch7.GeometricElements
Dim axis2D7 As Axis2D
Set axis2D7 = geometricElements7.Item("AbsoluteAxis")
Dim line2D17 As Line2D
Set line2D17 = axis2D7.GetItem("HDirection")
line2D17.ReportName = 1
Dim line2D18 As Line2D
Set line2D18 = axis2D7.GetItem("VDirection")
line2D18.ReportName = 2

Dim circle2D6 As Circle2D
Set circle2D6 = factory2D7.CreateClosedCircle(0#, 0#, F / 2 + 0.15 * Da)
Dim point2D10 As Point2D
Set point2D10 = axis2D7.GetItem("Origin")
circle2D6.CenterPoint = point2D10
circle2D6.ReportName = 3

Dim constraints7 As Constraints
Set constraints7 = sketch7.Constraints
Dim reference35 As Reference
Set reference35 = part3.CreateReferenceFromObject(circle2D6)
Dim constraint16 As Constraint
Set constraint16 = constraints7.AddMonoEltCst(catCstTypeRadius, reference35)
constraint16.Mode = catCstModeDrivingDimension
Dim length15 As Length
Set length15 = constraint16.Dimension
length15.Value = F / 2 + 0.15 * Da

sketch7.CloseEdition
part3.InWorkObject = sketch7
part3.Update

Dim pocket3 As Pocket
Set pocket3 = shapeFactory3.AddNewPocket(sketch7, B / 2 - r)
Dim limit5 As Limit
Set limit5 = pocket3.FirstLimit
limit5.LimitMode = catUpThruNextLimit

part3.Update
```

```
'HUECO PARA LAS AGUJAS
Dim reference36 As Reference
Set reference36 = originElements3.PlaneXY
Dim sketch8 As Sketch
Set sketch8 = sketches3.Add(reference36)

Dim arrayOfVariantOfDouble8(8)
arrayOfVariantOfDouble8(0) = 0#
arrayOfVariantOfDouble8(1) = 0#
arrayOfVariantOfDouble8(2) = 0#
arrayOfVariantOfDouble8(3) = 1#
arrayOfVariantOfDouble8(4) = 0#
arrayOfVariantOfDouble8(5) = 0#
arrayOfVariantOfDouble8(6) = 0#
arrayOfVariantOfDouble8(7) = 1#
arrayOfVariantOfDouble8(8) = 0#
Set sketch8Variant = sketch8
sketch8Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble8

part3.InWorkObject = sketch8

Dim factory2D8 As Factory2D
Set factory2D8 = sketch8.OpenEdition()
Dim geometricElements8 As GeometricElements
Set geometricElements8 =
sketch8.GeometricElements
Dim axis2D8 As Axis2D
Set axis2D8 =
geometricElements8.Item("AbsoluteAxis")
Dim line2D19 As Line2D
Set line2D19 = axis2D8.GetItem("HDirection")
line2D19.ReportName = 1
Dim line2D20 As Line2D
Set line2D20 = axis2D8.GetItem("VDirection")
line2D20.ReportName = 2

Dim point2D11 As Point2D
Set point2D11 = factory2D8.CreatePoint(-0.4 * B, -
Da / 2)
point2D11.ReportName = 3

Dim point2D12 As Point2D
Set point2D12 = factory2D8.CreatePoint(0.4 * B, -Da
/ 2)
point2D12.ReportName = 4

Dim line2D21 As Line2D
Set line2D21 = factory2D8.CreateLine(-0.4 * B, -Da
/ 2, 0.4 * B, -Da / 2)
line2D21.ReportName = 5
line2D21.StartPoint = point2D11
line2D21.EndPoint = point2D12

Dim constraints8 As Constraints
Set constraints8 = sketch8.Constraints

Dim reference37 As Reference
Set reference37 =
part3.CreateReferenceFromObject(line2D21)
Dim reference38 As Reference
Set reference38 =
part3.CreateReferenceFromObject(line2D19)
Dim constraint17 As Constraint
Set constraint17 =
constraints8.AddBiEltCst(catCstTypeHorizontality,
reference37, reference38)
constraint17.Mode = catCstModeDrivingDimension

Dim reference39 As Reference
Set reference39 =
part3.CreateReferenceFromObject(line2D21)
Dim constraint18 As Constraint
Set constraint18 =
constraints8.AddMonoEltCst(catCstTypeLength,
reference39)
constraint18.Mode = catCstModeDrivingDimension
Dim length16 As Length
Set length16 = constraint18.Dimension
length16.Value = 0.8 * B

Dim point2D13 As Point2D
Set point2D13 = factory2D8.CreatePoint(0.4 * B, Da
/ 2)
point2D13.ReportName = 6
Dim line2D22 As Line2D
Set line2D22 = factory2D8.CreateLine(0.4 * B, -Da /
2, 0.4 * B, Da / 2)
line2D22.ReportName = 7
line2D22.StartPoint = point2D12
line2D22.EndPoint = point2D13

Dim reference40 As Reference
Set reference40 =
part3.CreateReferenceFromObject(line2D22)
Dim reference41 As Reference
Set reference41 =
part3.CreateReferenceFromObject(line2D20)
Dim constraint19 As Constraint
Set constraint19 =
constraints8.AddBiEltCst(catCstTypeVerticality,
reference40, reference41)
constraint19.Mode = catCstModeDrivingDimension

Dim reference42 As Reference
Set reference42 =
part3.CreateReferenceFromObject(line2D22)
Dim constraint20 As Constraint
Set constraint20 =
constraints8.AddMonoEltCst(catCstTypeLength,
reference42)
constraint20.Mode = catCstModeDrivingDimension
Dim length17 As Length
Set length17 = constraint20.Dimension
length17.Value = Da

Dim point2D14 As Point2D
Set point2D14 = factory2D8.CreatePoint(-0.4 * B, Da
/ 2)
point2D14.ReportName = 8
Dim line2D23 As Line2D
Set line2D23 = factory2D8.CreateLine(0.4 * B, Da /
2, -0.4 * B, Da / 2)
line2D23.ReportName = 9
line2D23.StartPoint = point2D13
line2D23.EndPoint = point2D14

Dim reference43 As Reference
Set reference43 =
part3.CreateReferenceFromObject(line2D23)
Dim reference44 As Reference
```

```
Set reference44 = part3.CreateReferenceFromObject(line2D19)
Dim constraint21 As Constraint
Set constraint21 = constraints8.AddBiEltCst(catCstTypeHorizontality, reference43, reference44)
constraint21.Mode = catCstModeDrivingDimension

Dim line2D24 As Line2D
Set line2D24 = factory2D8.CreateLine(-0.4 * B, Da / 2, -0.4 * B, -Da / 2)
line2D24.ReportName = 10
line2D24.StartPoint = point2D14
line2D24.EndPoint = point2D11

Dim reference45 As Reference
Set reference45 = part3.CreateReferenceFromObject(line2D24)
Dim reference46 As Reference
Set reference46 = part3.CreateReferenceFromObject(line2D20)
Dim constraint22 As Constraint
Set constraint22 = constraints8.AddBiEltCst(catCstTypeVerticality, reference45, reference46)
constraint22.Mode = catCstModeDrivingDimension

Dim reference47 As Reference
Set reference47 = part3.CreateReferenceFromObject(line2D20)
Dim reference48 As Reference
Set reference48 = part3.CreateReferenceFromObject(line2D22)
Dim constraint23 As Constraint
Set constraint23 = constraints8.AddBiEltCst(catCstTypeDistance, reference47, reference48)
constraint23.Mode = catCstModeDrivingDimension
Dim length18 As Length
Set length18 = constraint23.Dimension
length18.Value = 0.4 * B

Dim reference49 As Reference
Set reference49 = part3.CreateReferenceFromObject(line2D19)
Dim reference50 As Reference
Set reference50 = part3.CreateReferenceFromObject(line2D21)
Dim constraint24 As Constraint
Set constraint24 = constraints8.AddBiEltCst(catCstTypeDistance, reference49, reference50)
constraint24.Mode = catCstModeDrivingDimension
Dim length19 As Length
Set length19 = constraint24.Dimension
length19.Value = Da / 2
sketch8.CloseEdition
part3.InWorkObject = sketch8
part3.Update

Dim pocket4 As Pocket
Set pocket4 = shapeFactory3.AddNewPocket(sketch8, 20#)
Dim limit6 As Limit
Set limit6 = pocket4.FirstLimit
pocket4.DirectionOrientation = catRegularOrientation
limit6.LimitMode = catUpToLastLimit
part3.Update

'PATRÓN CIRCULAR
Dim reference51 As Reference
Set reference51 = part3.CreateReferenceFromName("")
Dim reference52 As Reference
Set reference52 = part3.CreateReferenceFromName("")

Dim circPattern1 As CircPattern
Set circPattern1 = shapeFactory3.AddNewCircPattern(pocket4, 1, 2, 20#, 45#, 1, 1, reference51, reference52, True, 0#, True)
circPattern1.CircularPatternParameters = catInstancesandAngularSpacing

Dim angularRepartition As angularRepartition
Set angularRepartition = circPattern1.angularRepartition

'calculo del número de agujeros para ello definimos primero el número pi
Const PI = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es F / 2 + Da / 2
Dim longitud As Double
longitud = 2 * PI * (F / 2 + Da / 2)
'crearemos una variable auxiliar que usara el comando matemático Round para redondear el numero de huecos que tendrá el separador.
'para calcular este número hacemos la hipótesis de que la distancia entre agujas será del diámetro de las agujas
Dim variableaux As Double
variableaux = Math.Round(longitud / (Da + Da))
'ya tenemos el numero de huecos del separador
Dim intParam As intParam
Set intParam = angularRepartition.InstancesCount
intParam.Value = variableaux
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angle As angle
Set angle = angularRepartition.AngularSpacing
angle.Value = 360 / variableaux
Dim reference53 As Reference
Set reference53 = part3.CreateReferenceFromObject(line2D19)
circPattern1.SetRotationAxis reference53

part3.UpdateObject circPattern1
part3.Update
Dim product3 As Product
Set product3 = partDocument3.GetItem("Part4")
product3.PartNumber = "jaula"
partDocument3.SaveAs objPath & "\jaula.CATPart"
partDocument3.Close

'====================================
'----------------------------------ENSAMBLAJE--------
'====================================
Dim documents5 As Documents
Set documents5 = CATIA.Documents
Dim productDocument1 As ProductDocument
```

```
Set productDocument1 = documents5.Add("Product")
Dim product5 As Product
Set product5 = productDocument1.Product
Dim products1 As Products
Set products1 = product5.Products

Dim arrayOfVariantOfBSTR1(0)
arrayOfVariantOfBSTR1(0) = objPath & "\aroext.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR1, "All"

Dim constraints9 As Constraints
Set constraints9 = product5.Connections("CATIAConstraints")
'Fix del aro externo
Dim reference54 As Reference
Set reference54 = product5.CreateReferenceFromName("Product5/aroexterno.1/!Product5/aroexterno.1/")
Dim constraint25 As Constraint
Set constraint25 = constraints9.AddMonoEltCst(catCstTypeReference, reference54)

Dim arrayOfVariantOfBSTR2(0)
arrayOfVariantOfBSTR2(0) = objPath & "\agujas.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR2, "All"

Dim arrayOfVariantOfBSTR3(0)
arrayOfVariantOfBSTR3(0) = objPath & "\jaula.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR3, "All"

'coincidencia de ejes del aro externo y la jaula
Dim reference55 As Reference
Set reference55 = product5.CreateReferenceFromName("Product5/jaula.1/!Axis:(Selection_RSur:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());CircPattern.1_ResultOUT;Z0;G3055))")
Dim reference56 As Reference
Set reference56 = product5.CreateReferenceFromName("Product5/aroexterno.1/!Axis:(Selection_RSur:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());Slot.1_ResultOUT;Z0;G3055))")
Dim constraint26 As Constraint
Set constraint26 = constraints9.AddBiEltCst(catCstTypeOn, reference55, reference56)

'idem aro externo y jaula
Dim reference57 As Reference
Set reference57 = product5.CreateReferenceFromName("Product5/aroexterno.1/!yz plane")
Dim reference58 As Reference
Set reference58 = product5.CreateReferenceFromName("Product5/jaula.1/!yz plane")
Dim constraint27 As Constraint
Set constraint27 = constraints9.AddBiEltCst(catCstTypeDistance, reference57, reference58)
Dim length20 As Length
Set length20 = constraint27.Dimension
length20.Value = 0#
constraint27.Orientation = catCstOrientSame

'constrainst agujas y jaula
Dim reference69 As Reference
Set reference69 = product5.CreateReferenceFromName("Product5/agujas.1/!xy plane")
Dim reference70 As Reference
Set reference70 = product5.CreateReferenceFromName("Product5/jaula.1/!xy plane")
Dim constraint32 As Constraint
Set constraint32 = constraints9.AddBiEltCst(catCstTypeDistance, reference69, reference70)
Dim length25 As Length
Set length25 = constraint32.Dimension
length25.Value = 0#

Dim reference71 As Reference
Set reference71 = product5.CreateReferenceFromName("Product5/agujas.1/!yz plane")
Dim reference72 As Reference
Set reference72 = product5.CreateReferenceFromName("Product5/jaula.1/!yz plane")
Dim constraint33 As Constraint
Set constraint33 = constraints9.AddBiEltCst(catCstTypeDistance, reference71, reference72)
Dim length26 As Length
Set length26 = constraint33.Dimension
length26.Value = 0#

Dim reference73 As Reference
Set reference73 = product5.CreateReferenceFromName("Product5/agujas.1/!zx plane")
Dim reference74 As Reference
Set reference74 = product5.CreateReferenceFromName("Product5/jaula.1/!zx plane")
Dim constraint34 As Constraint
Set constraint34 = constraints9.AddBiEltCst(catCstTypeDistance, reference73, reference74)
Dim length27 As Length
Set length27 = constraint34.Dimension
length27.Value = 0#
constraint34.Orientation = catCstOrientSame

product5.Update
productDocument1.SaveAs objPath & "\AgujasSinAroInt.CATProduct"
```

- **Rodamiento de agujas con aro interno.**

```
Ídem al sin aro pero con un aro interno.
'=====================================
'----------------------------------anillo interno-------------
'=====================================
Dim documents4 As Documents
Set documents4 = CATIA.Documents
Dim partDocument4 As PartDocument
Set partDocument4 = documents4.Add("Part")
Dim part4 As Part
Set part4 = partDocument4.Part
Dim bodies4 As Bodies
Set bodies4 = part4.Bodies
Dim body4 As Body
Set body4 = bodies4.Item("PartBody")
Dim sketches4 As Sketches
Set sketches4 = body4.Sketches
Dim originElements4 As OriginElements
Set originElements4 = part4.OriginElements
Dim reference54 As Reference
Set reference54 = originElements4.PlaneYZ
Dim sketch9 As Sketch
Set sketch9 = sketches4.Add(reference54)

Dim arrayOfVariantOfDouble9(8)
arrayOfVariantOfDouble9(0) = 0#
arrayOfVariantOfDouble9(1) = 0#
arrayOfVariantOfDouble9(2) = 0#
arrayOfVariantOfDouble9(3) = 0#
arrayOfVariantOfDouble9(4) = 1#
arrayOfVariantOfDouble9(5) = 0#
arrayOfVariantOfDouble9(6) = 0#
arrayOfVariantOfDouble9(7) = 0#
arrayOfVariantOfDouble9(8) = 1#
Set sketch9Variant = sketch9
sketch9Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble9

part4.InWorkObject = sketch9

'PAD
Dim factory2D9 As Factory2D
Set factory2D9 = sketch9.OpenEdition()
Dim geometricElements9 As GeometricElements
Set geometricElements9 = sketch9.GeometricElements

Dim axis2D9 As Axis2D
Set axis2D9 = geometricElements9.Item("AbsoluteAxis")
Dim line2D25 As Line2D
Set line2D25 = axis2D9.GetItem("HDirection")
line2D25.ReportName = 1
Dim line2D26 As Line2D
Set line2D26 = axis2D9.GetItem("VDirection")
line2D26.ReportName = 2

Dim circle2D7 As Circle2D
Set circle2D7 = factory2D9.CreateClosedCircle(0#, 0#, F / 2)
Dim point2D15 As Point2D
Set point2D15 = axis2D9.GetItem("Origin")
circle2D7.CenterPoint = point2D15
circle2D7.ReportName = 3

Dim constraints9 As Constraints
Set constraints9 = sketch9.Constraints
Dim reference55 As Reference
Set reference55 = part4.CreateReferenceFromObject(circle2D7)
Dim constraint25 As Constraint
Set constraint25 = constraints9.AddMonoEltCst(catCstTypeRadius, reference55)
constraint25.Mode = catCstModeDrivingDimension
Dim length20 As Length
Set length20 = constraint25.Dimension
length20.Value = F / 2

sketch9.CloseEdition
part4.InWorkObject = sketch9
part4.Update

Dim shapeFactory4 As ShapeFactory
Set shapeFactory4 = part4.ShapeFactory
Dim pad4 As Pad
Set pad4 = shapeFactory4.AddNewPad(sketch9, B / 2)
Dim limit7 As Limit
Set limit7 = pad4.FirstLimit
Dim length21 As Length
Set length21 = limit7.Dimension
length21.Value = B / 2
pad4.IsSymmetric = True
part4.Update

'POCKET
Dim reference56 As Reference
Set reference56 = part4.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")

Dim sketch10 As Sketch
Set sketch10 = sketches4.Add(reference56)

Dim arrayOfVariantOfDouble10(8)
arrayOfVariantOfDouble10(0) = B / 2
arrayOfVariantOfDouble10(1) = 0#
arrayOfVariantOfDouble10(2) = 0#
arrayOfVariantOfDouble10(3) = 0#
arrayOfVariantOfDouble10(4) = 1#
arrayOfVariantOfDouble10(5) = 0#
arrayOfVariantOfDouble10(6) = 0#
arrayOfVariantOfDouble10(7) = 0#
arrayOfVariantOfDouble10(8) = 1#
Set sketch10Variant = sketch10
sketch10Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble10
part4.InWorkObject = sketch10

Dim factory2D10 As Factory2D
Set factory2D10 = sketch10.OpenEdition()
Dim geometricElements10 As GeometricElements
Set geometricElements10 = sketch10.GeometricElements

Dim axis2D10 As Axis2D
Set axis2D10 = geometricElements10.Item("AbsoluteAxis")
Dim line2D27 As Line2D
Set line2D27 = axis2D10.GetItem("HDirection")
line2D27.ReportName = 1
Dim line2D28 As Line2D
Set line2D28 = axis2D10.GetItem("VDirection")
```

```
line2D28.ReportName = 2

Dim circle2D8 As Circle2D
Set circle2D8 = factory2D10.CreateClosedCircle(0#,
0#, d_ / 2)
Dim point2D16 As Point2D
Set point2D16 = axis2D10.GetItem("Origin")
circle2D8.CenterPoint = point2D16
circle2D8.ReportName = 3

Dim constraints10 As Constraints
Set constraints10 = sketch10.Constraints

Dim reference57 As Reference
Set reference57 =
part4.CreateReferenceFromObject(circle2D8)
Dim constraint26 As Constraint
Set constraint26 =
constraints10.AddMonoEltCst(catCstTypeRadius,
reference57)
constraint26.Mode = catCstModeDrivingDimension
Dim length22 As Length
Set length22 = constraint26.Dimension
length22.Value = d_ / 2
sketch10.CloseEdition
part4.InWorkObject = sketch10
part4.Update

Dim pocket4 As Pocket
Set pocket4 =
shapeFactory4.AddNewPocket(sketch10, B / 2)
Dim limit8 As Limit
Set limit8 = pocket4.FirstLimit
limit8.LimitMode = catUpThruNextLimit
part4.Update

'Chaflán
Dim reference58 As Reference
Set reference58 =
part4.CreateReferenceFromName("")
Dim constRadEdgeFillet2 As ConstRadEdgeFillet
Set constRadEdgeFillet2 =
shapeFactory4.AddNewSolidEdgeFilletWithConstant
Radius(reference58,
catTangencyFilletEdgePropagation, r)

Dim reference59 As Reference
Set reference59 =
part4.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pad.1;2);None:();Cf11:());Face:(Brp:(P
ocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());None:
(Limits1:();Limits2:());Cf11:());WithTemporaryBody
;WithoutBuildError;WithSelectingFeatureSupport;M
FBRepVersion_CXR15)", pocket1)
constRadEdgeFillet2.AddObjectToFillet reference59
constRadEdgeFillet2.EdgePropagation =
catTangencyFilletEdgePropagation
Dim reference60 As Reference
Set reference60 =
part4.CreateReferenceFromBRepName("REdge:(Edg
e:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();
Cf11:());Face:(Brp:(Pad.1;1);None:();Cf11:());None:(
Limits1:();Limits2:());Cf11:());WithTemporaryBody;
WithoutBuildError;WithSelectingFeatureSupport;MF
BRepVersion_CXR15)", pocket1)
constRadEdgeFillet2.AddObjectToFillet reference60
constRadEdgeFillet2.EdgePropagation =
catTangencyFilletEdgePropagation
part4.Update

Dim product4 As Product
Set product4 = partDocument4.GetItem("Part4")
product4.PartNumber = "arointerno"
partDocument4.SaveAs objPath & "\aroint.CATPart"
partDocument4.Close
```

- **Corona de agujas.**

```
Private Sub CommandButton1_Click()
'====================================
'----------------------------------jaula----------------------
'====================================
Dim documents1 As Documents
Set documents1 = CATIA.Documents
Dim partDocument1 As PartDocument
Set partDocument1 = documents1.Add("Part")
Dim part1 As Part
Set part1 = partDocument1.Part
Dim bodies1 As Bodies
Set bodies1 = part1.Bodies
Dim body1 As Body
Set body1 = bodies1.Item("PartBody")
Dim sketches1 As Sketches
Set sketches1 = body1.Sketches
Dim originElements1 As OriginElements
Set originElements1 = part1.OriginElements
Dim reference1 As Reference
Set reference1 = originElements1.PlaneYZ
Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)

Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0#
arrayOfVariantOfDouble1(1) = 0#
arrayOfVariantOfDouble1(2) = 0#
arrayOfVariantOfDouble1(3) = 0#
arrayOfVariantOfDouble1(4) = 1#
arrayOfVariantOfDouble1(5) = 0#
arrayOfVariantOfDouble1(6) = 0#
arrayOfVariantOfDouble1(7) = 0#
arrayOfVariantOfDouble1(8) = 1#
Set sketch1Variant = sketch1
sketch1Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble1
part1.InWorkObject = sketch1

Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()
Dim geometricElements1 As GeometricElements
Set geometricElements1 =
sketch1.GeometricElements
Dim axis2D1 As Axis2D
Set axis2D1 =
geometricElements1.Item("AbsoluteAxis")
Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2
'PAD
Dim circle2D1 As Circle2D
Set circle2D1 = factory2D1.CreateClosedCircle(0#,
0#, Dc / 2)
```

```
Dim point2D1 As Point2D
Set point2D1 = axis2D1.GetItem("Origin")
circle2D1.CenterPoint = point2D1
circle2D1.ReportName = 3

Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints
Dim reference2 As Reference
Set reference2 = part1.CreateReferenceFromObject(circle2D1)
Dim constraint1 As Constraint
Set constraint1 = constraints1.AddMonoEltCst(catCstTypeRadius, reference2)
constraint1.Mode = catCstModeDrivingDimension
Dim length1 As Length
Set length1 = constraint1.Dimension
length1.Value = Dc / 2
sketch1.CloseEdition
part1.InWorkObject = sketch1
part1.UpdateObject sketch1

Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory
Dim pad1 As Pad
Set pad1 = shapeFactory1.AddNewPad(sketch1, Dw / 4)
pad1.IsSymmetric = True
Dim limit1 As Limit
Set limit1 = pad1.FirstLimit
Dim length2 As Length
Set length2 = limit1.Dimension
length2.Value = Dw / 4
part1.UpdateObject pad1

Dim reference3 As Reference
Set reference3 = part1.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")
Dim sketch2 As Sketch
Set sketch2 = sketches1.Add(reference3)
Dim arrayOfVariantOfDouble2(8)
arrayOfVariantOfDouble2(0) = Dw / 4
arrayOfVariantOfDouble2(1) = 0#
arrayOfVariantOfDouble2(2) = 0#
arrayOfVariantOfDouble2(3) = 0#
arrayOfVariantOfDouble2(4) = 1#
arrayOfVariantOfDouble2(5) = 0#
arrayOfVariantOfDouble2(6) = 0#
arrayOfVariantOfDouble2(7) = 0#
arrayOfVariantOfDouble2(8) = 1#
Set sketch2Variant = sketch2
sketch2Variant.SetAbsoluteAxisData arrayOfVariantOfDouble2
part1.InWorkObject = sketch2

'POCKET
Dim factory2D2 As Factory2D
Set factory2D2 = sketch2.OpenEdition()
Dim geometricElements2 As GeometricElements
Set geometricElements2 = sketch2.GeometricElements
Dim axis2D2 As Axis2D
Set axis2D2 = geometricElements2.Item("AbsoluteAxis")
Dim line2D3 As Line2D
Set line2D3 = axis2D2.GetItem("HDirection")
line2D3.ReportName = 1
Dim line2D4 As Line2D
Set line2D4 = axis2D2.GetItem("VDirection")
line2D4.ReportName = 2

Dim circle2D2 As Circle2D
Set circle2D2 = factory2D2.CreateClosedCircle(0#, 0#, d_ / 2)
Dim point2D2 As Point2D
Set point2D2 = axis2D2.GetItem("Origin")
circle2D2.CenterPoint = point2D2
circle2D2.ReportName = 3

Dim constraints2 As Constraints
Set constraints2 = sketch2.Constraints
Dim reference4 As Reference
Set reference4 = part1.CreateReferenceFromObject(circle2D2)
Dim constraint2 As Constraint
Set constraint2 = constraints2.AddMonoEltCst(catCstTypeRadius, reference4)
constraint2.Mode = catCstModeDrivingDimension
Dim length3 As Length
Set length3 = constraint2.Dimension
length3.Value = d_ / 2
sketch2.CloseEdition
part1.InWorkObject = sketch2
part1.UpdateObject sketch2

Dim pocket1 As Pocket
Set pocket1 = shapeFactory1.AddNewPocket(sketch2, Dw / 4)
Dim limit2 As Limit
Set limit2 = pocket1.FirstLimit
limit2.LimitMode = catUpToLastLimit
part1.UpdateObject pocket1

'Huecos para los rodillos
Dim reference5 As Reference
Set reference5 = part1.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pocket.1_ResultOUT;Z0;G3055)")
Dim sketch3 As Sketch
Set sketch3 = sketches1.Add(reference5)

Dim arrayOfVariantOfDouble3(8)
arrayOfVariantOfDouble3(0) = Dw / 4
arrayOfVariantOfDouble3(1) = 0#
arrayOfVariantOfDouble3(2) = 0#
arrayOfVariantOfDouble3(3) = 0#
arrayOfVariantOfDouble3(4) = 1#
arrayOfVariantOfDouble3(5) = 0#
arrayOfVariantOfDouble3(6) = 0#
arrayOfVariantOfDouble3(7) = 0#
arrayOfVariantOfDouble3(8) = 1#
Set sketch3Variant = sketch3
sketch3Variant.SetAbsoluteAxisData arrayOfVariantOfDouble3
part1.InWorkObject = sketch3

Dim factory2D3 As Factory2D
Set factory2D3 = sketch3.OpenEdition()
Dim geometricElements3 As GeometricElements
```

```
Set geometricElements3 = sketch3.GeometricElements
Dim axis2D3 As Axis2D
Set axis2D3 = geometricElements3.Item("AbsoluteAxis")
Dim line2D5 As Line2D
Set line2D5 = axis2D3.GetItem("HDirection")
line2D5.ReportName = 1
Dim line2D6 As Line2D
Set line2D6 = axis2D3.GetItem("VDirection")
line2D6.ReportName = 2

Dim point2D3 As Point2D
Set point2D3 = factory2D3.CreatePoint(-Dw / 2, Eb / 2)
point2D3.ReportName = 3
Dim point2D4 As Point2D
Set point2D4 = factory2D3.CreatePoint(Dw / 2, Eb / 2)
point2D4.ReportName = 4
Dim line2D7 As Line2D
Set line2D7 = factory2D3.CreateLine(-Dw / 2, Eb / 2, Dw / 2, Eb / 2)
line2D7.ReportName = 5
line2D7.StartPoint = point2D3
line2D7.EndPoint = point2D4

Dim point2D5 As Point2D
Set point2D5 = factory2D3.CreatePoint(Dw / 2, Ea / 2)
point2D5.ReportName = 6
Dim line2D8 As Line2D
Set line2D8 = factory2D3.CreateLine(Dw / 2, Eb / 2, Dw / 2, Ea / 2)
line2D8.ReportName = 7
line2D8.EndPoint = point2D4
line2D8.StartPoint = point2D5

Dim point2D6 As Point2D
Set point2D6 = factory2D3.CreatePoint(-Dw / 2, Ea / 2)
point2D6.ReportName = 8
Dim line2D9 As Line2D
Set line2D9 = factory2D3.CreateLine(Dw / 2, Ea / 2, -Dw / 2, Ea / 2)
line2D9.ReportName = 9
line2D9.StartPoint = point2D5
line2D9.EndPoint = point2D6

Dim line2D10 As Line2D
Set line2D10 = factory2D3.CreateLine(-Dw / 2, Ea / 2, -Dw / 2, Eb / 2)
line2D10.ReportName = 10
line2D10.EndPoint = point2D6
line2D10.StartPoint = point2D3

Dim constraints3 As Constraints
Set constraints3 = sketch3.Constraints
Dim reference6 As Reference
Set reference6 = part1.CreateReferenceFromObject(line2D7)
Dim reference7 As Reference
Set reference7 = part1.CreateReferenceFromObject(line2D5)
Dim constraint3 As Constraint
Set constraint3 = constraints3.AddBiEltCst(catCstTypeHorizontality, reference6, reference7)
constraint3.Mode = catCstModeDrivingDimension

Dim reference8 As Reference
Set reference8 = part1.CreateReferenceFromObject(line2D9)
Dim reference9 As Reference
Set reference9 = part1.CreateReferenceFromObject(line2D5)
Dim constraint4 As Constraint
Set constraint4 = constraints3.AddBiEltCst(catCstTypeHorizontality, reference8, reference9)
constraint4.Mode = catCstModeDrivingDimension

Dim reference10 As Reference
Set reference10 = part1.CreateReferenceFromObject(line2D8)
Dim reference11 As Reference
Set reference11 = part1.CreateReferenceFromObject(line2D6)
Dim constraint5 As Constraint
Set constraint5 = constraints3.AddBiEltCst(catCstTypeVerticality, reference10, reference11)
constraint5.Mode = catCstModeDrivingDimension

Dim reference12 As Reference
Set reference12 = part1.CreateReferenceFromObject(line2D10)
Dim reference13 As Reference
Set reference13 = part1.CreateReferenceFromObject(line2D6)
Dim constraint6 As Constraint
Set constraint6 = constraints3.AddBiEltCst(catCstTypeVerticality, reference12, reference13)
constraint6.Mode = catCstModeDrivingDimension

Dim reference14 As Reference
Set reference14 = part1.CreateReferenceFromObject(line2D8)
Dim reference15 As Reference
Set reference15 = part1.CreateReferenceFromObject(line2D10)
Dim constraint7 As Constraint
Set constraint7 = constraints3.AddBiEltCst(catCstTypeDistance, reference14, reference15)
constraint7.Mode = catCstModeDrivingDimension

Dim length4 As Length
Set length4 = constraint7.Dimension
length4.Value = Dw

Dim reference16 As Reference
Set reference16 = part1.CreateReferenceFromObject(line2D8)
Dim reference17 As Reference
Set reference17 = part1.CreateReferenceFromObject(line2D6)
Dim constraint8 As Constraint
Set constraint8 = constraints3.AddBiEltCst(catCstTypeDistance, reference16, reference17)
```

```
constraint8.Mode = catCstModeDrivingDimension

Dim length5 As Length
Set length5 = constraint8.Dimension
length5.Value = Dw / 2

Dim reference18 As Reference
Set reference18 = part1.CreateReferenceFromObject(line2D9)
Dim reference19 As Reference
Set reference19 = part1.CreateReferenceFromObject(line2D5)
Dim constraint9 As Constraint
Set constraint9 = constraints3.AddBiEltCst(catCstTypeDistance, reference18, reference19)
constraint9.Mode = catCstModeDrivingDimension
Dim length6 As Length
Set length6 = constraint9.Dimension
length6.Value = Ea / 2

Dim reference20 As Reference
Set reference20 = part1.CreateReferenceFromObject(line2D7)
Dim reference21 As Reference
Set reference21 = part1.CreateReferenceFromObject(line2D5)
Dim constraint10 As Constraint
Set constraint10 = constraints3.AddBiEltCst(catCstTypeDistance, reference20, reference21)
constraint10.Mode = catCstModeDrivingDimension
Dim length7 As Length
Set length7 = constraint10.Dimension
length7.Value = Eb / 2
sketch3.CloseEdition
part1.InWorkObject = sketch3
part1.UpdateObject sketch3

Dim pocket2 As Pocket
Set pocket2 = shapeFactory1.AddNewPocket(sketch3, Dw / 4)
Dim limit3 As Limit
Set limit3 = pocket2.FirstLimit
limit3.LimitMode = catUpToLastLimit
part1.UpdateObject pocket2

Dim reference22 As Reference
Set reference22 = part1.CreateReferenceFromName("")
Dim reference23 As Reference
Set reference23 = part1.CreateReferenceFromName("")
Dim circPattern1 As CircPattern
Set circPattern1 = shapeFactory1.AddNewCircPattern(pocket2, 1, 2, 20#, 45#, 1, 1, reference22, reference23, True, 0#, True)
circPattern1.CircularPatternParameters = catInstancesandAngularSpacing

Dim angularRepartition1 As angularRepartition
Set angularRepartition1 = circPattern1.angularRepartition

'calculo del número de agujeros para ello definimos primero el número pi
Const PI = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de los rodillos que en este caso es (D+d)/4
Dim longitud As Double
longitud = 2 * PI * ((Dc - d_) / 4 + d_ / 2)
'crearemos una variable auxiliar que usara el comando matemático Round para redondear el numero de huecos que tendrá el separador.
'para calcular este número hacemos la hipótesis de que la distancia entre rodillos será de Dw/2
Dim variableaux As Double
variableaux = Math.Round(longitud / (3 / 2 * Dw))
'ya tenemos el numero de huecos del separador
Dim intParam As intParam
Set intParam = angularRepartition1.InstancesCount
intParam.Value = variableaux
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angularRepartition2 As angularRepartition
Set angularRepartition2 = circPattern1.angularRepartition
Dim angle As angle
Set angle = angularRepartition2.AngularSpacing
angle.Value = 360 / variableaux
Dim refrot7 As Reference
Set refrot7 = part1.CreateReferenceFromObject(reference1)
circPattern1.SetRotationAxis refrot7

part1.UpdateObject circPattern1
part1.Update
Dim product1 As Product
Set product1 = partDocument1.GetItem("Part1")
product1.PartNumber = "jaula"
partDocument1.SaveAs objPath & "\jaula.CATPart"
partDocument1.Close
'=====================================
'----------------------------------Agujas--------------------
'=====================================
Dim documents2 As Documents
Set documents2 = CATIA.Documents
Dim partDocument2 As PartDocument
Set partDocument2 = documents2.Add("Part")
Dim part2 As Part
Set part2 = partDocument2.Part
Dim bodies2 As Bodies
Set bodies2 = part2.Bodies
Dim body2 As Body
Set body2 = bodies2.Item("PartBody")
Dim sketches2 As Sketches
Set sketches2 = body2.Sketches
Dim originElements2 As OriginElements
Set originElements2 = part2.OriginElements
Dim reference24 As Reference
Set reference24 = originElements2.PlaneYZ

Dim sketch4 As Sketch
Set sketch4 = sketches2.Add(reference24)
Dim arrayOfVariantOfDouble4(8)
arrayOfVariantOfDouble4(0) = 0#
arrayOfVariantOfDouble4(1) = 0#
arrayOfVariantOfDouble4(2) = 0
arrayOfVariantOfDouble4(3) = 0#
arrayOfVariantOfDouble4(4) = 1#
arrayOfVariantOfDouble4(5) = 0#
arrayOfVariantOfDouble4(6) = 0#
```

```
arrayOfVariantOfDouble4(7) = 0#
arrayOfVariantOfDouble4(8) = 1#
Set sketch4Variant = sketch4
sketch4Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble4
part2.InWorkObject = sketch4

Dim factory2D4 As Factory2D
Set factory2D4 = sketch4.OpenEdition()
Dim geometricElements4 As GeometricElements
Set geometricElements4 =
sketch4.GeometricElements

Dim axis2D4 As Axis2D
Set axis2D4 =
geometricElements4.Item("AbsoluteAxis")
Dim line2D11 As Line2D
Set line2D11 = axis2D4.GetItem("HDirection")
line2D11.ReportName = 1
Dim line2D12 As Line2D
Set line2D12 = axis2D4.GetItem("VDirection")
line2D12.ReportName = 2

Dim point2D1a As Point2D
Set point2D1a = factory2D4.CreatePoint(0#, Eb / 2)
point2D1a.ReportName = 3
Dim constraints4 As Constraints
Set constraints4 = sketch4.Constraints
Dim reference2a As Reference
Set reference2a =
part2.CreateReferenceFromObject(point2D1a)
Dim reference3a As Reference
Set reference3a =
part2.CreateReferenceFromObject(line2D12)
Dim constraint1a As Constraint
Set constraint1a =
constraints4.AddBiEltCst(catCstTypeOn,
reference2a, reference3a)
constraint1a.Mode = catCstModeDrivingDimension

Dim point2D2a As Point2D
Set point2D2a = factory2D4.CreatePoint(Dw / 2, Eb /
2)
point2D2a.ReportName = 4
Dim line2D3a As Line2D
Set line2D3a = factory2D4.CreateLine(0, Eb / 2, Dw
/ 2, Eb / 2)
line2D3a.ReportName = 5
line2D3a.StartPoint = point2D1a
line2D3a.EndPoint = point2D2a

Dim point2D3a As Point2D
Set point2D3a = factory2D4.CreatePoint(Dw / 2, Ea /
2)
point2D3a.ReportName = 6
Dim line2D4a As Line2D
Set line2D4a = factory2D4.CreateLine(Dw / 2, Eb /
2, Dw / 2, Ea / 2)
line2D4a.ReportName = 7
line2D4a.EndPoint = point2D2a
line2D4a.StartPoint = point2D3a

Dim point2D4a As Point2D
Set point2D4a = factory2D4.CreatePoint(0, Ea / 2)
point2D4a.ReportName = 8
Dim line2D5a As Line2D
Set line2D5a = factory2D4.CreateLine(Dw / 2, Ea / 2,
0, Ea / 2)
line2D5a.ReportName = 9
line2D5a.StartPoint = point2D3a
line2D5a.EndPoint = point2D4a

Dim line2D6a As Line2D
Set line2D6a = factory2D4.CreateLine(0, Ea / 2, 0,
Eb / 2)
line2D6a.ReportName = 10
line2D6a.EndPoint = point2D4a
line2D6a.StartPoint = point2D1a

Dim reference4a As Reference
Set reference4a =
part2.CreateReferenceFromObject(line2D3a)
Dim reference5a As Reference
Set reference5a =
part2.CreateReferenceFromObject(line2D11)
Dim constraint2a As Constraint
Set constraint2a =
constraints4.AddBiEltCst(catCstTypeHorizontality,
reference4a, reference5a)
constraint2a.Mode = catCstModeDrivingDimension

Dim reference6a As Reference
Set reference6a =
part2.CreateReferenceFromObject(line2D5a)
Dim reference7a As Reference
Set reference7a =
part2.CreateReferenceFromObject(line2D11)
Dim constraint3a As Constraint
Set constraint3a =
constraints4.AddBiEltCst(catCstTypeHorizontality,
reference6a, reference7a)
constraint3a.Mode = catCstModeDrivingDimension

Dim reference8a As Reference
Set reference8a =
part2.CreateReferenceFromObject(line2D4a)
Dim reference9a As Reference
Set reference9a =
part2.CreateReferenceFromObject(line2D12)
Dim constrainta4 As Constraint
Set constraint4a =
constraints4.AddBiEltCst(catCstTypeVerticality,
reference8a, reference9a)
constraint4a.Mode = catCstModeDrivingDimension

Dim reference10a As Reference
Set reference10a =
part2.CreateReferenceFromObject(line2D6a)
Dim reference11a As Reference
Set reference11a =
part2.CreateReferenceFromObject(line2D12)
Dim constraint5a As Constraint
Set constraint5a =
constraints4.AddBiEltCst(catCstTypeVerticality,
reference10a, reference11a)
constraint5a.Mode = catCstModeDrivingDimension

Dim reference12a As Reference
Set reference12a =
part2.CreateReferenceFromObject(line2D5a)
Dim reference13a As Reference
Set reference13a =
part2.CreateReferenceFromObject(line2D11)
```

```
Dim constraint6a As Constraint
Set constraint6a = constraints4.AddBiEltCst(catCstTypeDistance, reference12a, reference13a)
constraint6a.Mode = catCstModeDrivingDimension
Dim length1a As Length
Set length1a = constraint6a.Dimension
length1a.Value = Ea / 2

Dim reference14a As Reference
Set reference14a = part2.CreateReferenceFromObject(line2D4a)
Dim reference15a As Reference
Set reference15a = part2.CreateReferenceFromObject(line2D12)
Dim constraint7a As Constraint
Set constraint7a = constraints4.AddBiEltCst(catCstTypeDistance, reference14a, reference15a)
constraint7a.Mode = catCstModeDrivingDimension
Dim length2a As Length
Set length2a = constraint7a.Dimension
length2a.Value = Dw / 2

Dim reference16a As Reference
Set reference16a = part2.CreateReferenceFromObject(line2D4a)
Dim constraint8a As Constraint
Set constraint8a = constraints4.AddMonoEltCst(catCstTypeLength, reference16a)
constraint8a.Mode = catCstModeDrivingDimension
Dim length3a As Length
Set length3a = constraint8a.Dimension
length3a.Value = (Eb - Ea) / 2
sketch4.CloseEdition
part2.InWorkObject = sketch4
part2.UpdateObject sketch4

Dim shapeFactory2 As ShapeFactory
Set shapeFactory2 = part2.ShapeFactory
Dim shaft1 As Shaft
Set shaft1 = shapeFactory2.AddNewShaft(sketch4)
Dim reference19a As Reference
Set reference19a = part2.CreateReferenceFromObject(line2D12)
shaft1.RevoluteAxis = reference19a
part2.UpdateObject shaft1

Dim reference1b As Reference
Set reference1b = part2.CreateReferenceFromName("")
Dim constRadEdgeFillet1b As ConstRadEdgeFillet
Set constRadEdgeFillet1b = shapeFactory2.AddNewSolidEdgeFilletWithConstantRadius(reference1b, catTangencyFilletEdgePropagation, 0.5)
Dim reference2b As Reference
Set reference2b = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Shaft.1;0:(Brp:(Sketch.1;7)));None:();Cf11:());Face:(Brp:(Shaft.1;0:(Brp:(Sketch.1;5)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", shaft1)
constRadEdgeFillet1b.AddObjectToFillet reference2b
constRadEdgeFillet1b.EdgePropagation = catTangencyFilletEdgePropagation
Dim reference3b As Reference
Set reference3b = part2.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Shaft.1;0:(Brp:(Sketch.1;9)));None:();Cf11:());Face:(Brp:(Shaft.1;0:(Brp:(Sketch.1;7)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", shaft1)
constRadEdgeFillet1b.AddObjectToFillet reference3b
constRadEdgeFillet1b.EdgePropagation = catTangencyFilletEdgePropagation
part2.UpdateObject constRadEdgeFillet1b

Dim referenceb As Reference
Set referenceb = part2.CreateReferenceFromName("")
Dim referenceb1 As Reference
Set referenceb1 = part2.CreateReferenceFromName("")
Dim circPatternb As CircPattern
Set circPatternb = shapeFactory2.AddNewCircPattern(Nothing, 1, 2, 20#, 45#, 1, 1, referenceb, referenceb1, True, 0#, True)
circPatternb.CircularPatternParameters = catInstancesandAngularSpacing
Dim angularRepartitionb As angularRepartition
Set angularRepartitionb = circPatternb.angularRepartition

'calculo del número de agujeros para ello definimos primero el número pi
Const PIb = 3.14159265358979
'llamaremos longitud a la longitud de la circuferencia de radio la altura del centro de las bolas que en este caso es (D+d)/4
Dim longitudb As Double
longitudb = 2 * PIb * ((Dc - d_) / 4 + d_ / 2)
'crearemos una variable auxiliar que usara el comando matemático Round para redondear el numero de huecos que tendrá el separador.
'para calcular este número hacemos la hipótesis de que la distancia entre rodillos será de Dw/2
Dim variableauxb As Double
variableauxb = Math.Round(longitudb / (3 / 2 * Dw))
'ya tenemos el numero de huecos del separador
Dim intParamb As intParam
Set intParamb = angularRepartitionb.InstancesCount
intParamb.Value = variableauxb
'la separación entre huecos vendra dada simplemente diviendo los 360 grados entre el numero de huecos
Dim angleb As angle
Set angleb = angularRepartitionb.AngularSpacing
angleb.Value = 360 / variableauxb
circPatternb.SetRotationAxis reference24
part2.Update

Dim product2 As Product
Set product2 = partDocument2.GetItem("Part2")
product2.PartNumber = "agujas"

partDocument2.SaveAs objPath & "\agujas.catpart"
partDocument2.Close
'====================================
```

```
'-------------------------------------ENSAMBLAJE-------
'=====================================
Dim documents5 As Documents
Set documents5 = CATIA.Documents
Dim productDocument1 As ProductDocument
Set productDocument1 = documents5.Add("Product")
Dim product5 As Product
Set product5 = productDocument1.Product
Dim products1 As Products
Set products1 = product5.Products

Dim arrayOfVariantOfBSTR3(0)
arrayOfVariantOfBSTR3(0) = objPath & "\agujas.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR3, "All"

Dim arrayOfVariantOfBSTR4(0)
arrayOfVariantOfBSTR4(0) = objPath & "\jaula.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR4, "All"

Dim constraints5 As Constraints
Set constraints5 = product5.Connections("CATIAConstraints")
'Fijamos los rodillos a los huecos de la jaula.
Dim referencec1 As Reference
Set referencec1 = product5.CreateReferenceFromName("Product5/agujas.1/!zx plane")
Dim referencec2 As Reference
Set referencec2 = product5.CreateReferenceFromName("Product5/jaula.1/!zx plane")
Dim constraintc1 As Constraint
Set constraintc1 = constraints5.AddBiEltCst(catCstTypeDistance, referencec1, referencec2)

Dim referencec3 As Reference
Set referencec3 = product5.CreateReferenceFromName("Product5/agujas.1/!yz plane")
Dim referencec4 As Reference
Set referencec4 = product5.CreateReferenceFromName("Product5/jaula.1/!yz plane")
Dim constraintc2 As Constraint
Set constraintc2 = constraints5.AddBiEltCst(catCstTypeDistance, referencec3, referencec4)

Dim referencec5 As Reference
Set referencec5 = product5.CreateReferenceFromName("Product5/agujas.1/!xy plane")
Dim referencec6 As Reference
Set referencec6 = product5.CreateReferenceFromName("Product5/jaula.1/!xy plane")
Dim constraintc3 As Constraint
Set constraintc3 = constraints5.AddBiEltCst(catCstTypeDistance, referencec5, referencec6)

product5.Update
productDocument1.SaveAs objPath & "\CoronaAgujas.CATProduct"
End Sub
```

- **Rodamiento axial de agujas.**

Incluye una corona de agujas y dos arandelas

```
'=====================================
'---------------------------------Arandela de eje----------
'=====================================
Dim documents3 As Documents
Set documents3 = CATIA.Documents
Dim partDocument3 As PartDocument
Set partDocument3 = documents3.Add("Part")
Dim part3 As Part
Set part3 = partDocument3.Part
Dim bodies3 As Bodies
Set bodies3 = part3.Bodies
Dim body3 As Body
Set body3 = bodies3.Item("PartBody")
Dim sketches3 As Sketches
Set sketches3 = body3.Sketches
Dim originElements3 As OriginElements
Set originElements3 = part3.OriginElements
Dim reference31 As Reference
Set reference31 = originElements3.PlaneYZ
Dim sketch5 As Sketch
Set sketch5 = sketches3.Add(reference31)

Dim arrayOfVariantOfDouble5(8)
arrayOfVariantOfDouble5(0) = 0#
arrayOfVariantOfDouble5(1) = 0#
arrayOfVariantOfDouble5(2) = 0#
arrayOfVariantOfDouble5(3) = 0#
arrayOfVariantOfDouble5(4) = 1#
arrayOfVariantOfDouble5(5) = 0#
arrayOfVariantOfDouble5(6) = 0#
arrayOfVariantOfDouble5(7) = 0#
arrayOfVariantOfDouble5(8) = 1#
Set sketch5Variant = sketch5
sketch5Variant.SetAbsoluteAxisData arrayOfVariantOfDouble5
part3.InWorkObject = sketch5

Dim factory2D5 As Factory2D
Set factory2D5 = sketch5.OpenEdition()
Dim geometricElements5 As GeometricElements
Set geometricElements5 = sketch5.GeometricElements

Dim axis2D5 As Axis2D
Set axis2D5 = geometricElements5.Item("AbsoluteAxis")
Dim line2D13 As Line2D
Set line2D13 = axis2D5.GetItem("HDirection")
line2D13.ReportName = 1
Dim line2D14 As Line2D
Set line2D14 = axis2D5.GetItem("VDirection")
line2D14.ReportName = 2

Dim circle2D4 As Circle2D
Set circle2D4 = factory2D5.CreateClosedCircle(0#, 0#, Dc / 2)
Dim point2D8 As Point2D
```

```
Set point2D8 = axis2D5.GetItem("Origin")
circle2D4.CenterPoint = point2D8
circle2D4.ReportName = 3

Dim constraints5 As Constraints
Set constraints5 = sketch5.Constraints
Dim reference32 As Reference
Set reference32 = part3.CreateReferenceFromObject(circle2D4)
Dim constraint12 As Constraint
Set constraint12 = constraints5.AddMonoEltCst(catCstTypeRadius, reference32)
constraint12.Mode = catCstModeDrivingDimension
Dim length10 As Length
Set length10 = constraint12.Dimension
length10.Value = Dc / 2

sketch5.CloseEdition
part3.InWorkObject = sketch5
part3.UpdateObject sketch5

Dim shapeFactory3 As ShapeFactory
Set shapeFactory3 = part3.ShapeFactory
Dim pad3 As Pad
Set pad3 = shapeFactory3.AddNewPad(sketch5, B / 2)
pad3.IsSymmetric = True
Dim limit5 As Limit
Set limit5 = pad3.FirstLimit
Dim length11 As Length
Set length11 = limit5.Dimension
length11.Value = B / 2
part3.UpdateObject pad3
Dim reference33 As Reference
Set reference33 = part3.CreateReferenceFromName("Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultOUT;Z0;G3055)")

Dim sketch6 As Sketch
Set sketch6 = sketches3.Add(reference33)

Dim arrayOfVariantOfDouble6(8)
arrayOfVariantOfDouble6(0) = B / 2
arrayOfVariantOfDouble6(1) = 0#
arrayOfVariantOfDouble6(2) = 0#
arrayOfVariantOfDouble6(3) = 0#
arrayOfVariantOfDouble6(4) = 1#
arrayOfVariantOfDouble6(5) = 0#
arrayOfVariantOfDouble6(6) = 0#
arrayOfVariantOfDouble6(7) = 0#
arrayOfVariantOfDouble6(8) = 1#
Set sketch6Variant = sketch6
sketch6Variant.SetAbsoluteAxisData arrayOfVariantOfDouble6
part3.InWorkObject = sketch6

Dim factory2D6 As Factory2D
Set factory2D6 = sketch6.OpenEdition()
Dim geometricElements6 As GeometricElements
Set geometricElements6 = sketch6.GeometricElements

Dim axis2D6 As Axis2D
Set axis2D6 = geometricElements6.Item("AbsoluteAxis")
Dim line2D15 As Line2D
Set line2D15 = axis2D6.GetItem("HDirection")
line2D15.ReportName = 1
Dim line2D16 As Line2D
Set line2D16 = axis2D6.GetItem("VDirection")
line2D16.ReportName = 2

Dim circle2D5 As Circle2D
Set circle2D5 = factory2D6.CreateClosedCircle(0#, 0#, d_ / 2)
Dim point2D9 As Point2D
Set point2D9 = axis2D6.GetItem("Origin")
circle2D5.CenterPoint = point2D9
circle2D5.ReportName = 3

Dim constraints6 As Constraints
Set constraints6 = sketch6.Constraints
Dim reference34 As Reference
Set reference34 = part3.CreateReferenceFromObject(circle2D5)
Dim constraint13 As Constraint
Set constraint13 = constraints6.AddMonoEltCst(catCstTypeRadius, reference34)
constraint13.Mode = catCstModeDrivingDimension
Dim length12 As Length
Set length12 = constraint13.Dimension
length12.Value = d_ / 2

sketch6.CloseEdition
part3.InWorkObject = sketch6
part3.UpdateObject sketch6

Dim pocket3 As Pocket
Set pocket3 = shapeFactory3.AddNewPocket(sketch6, B / 2)
Dim limit6 As Limit
Set limit6 = pocket3.FirstLimit
limit6.LimitMode = catUpToLastLimit
part3.UpdateObject pocket3

Dim reference35 As Reference
Set reference35 = part3.CreateReferenceFromName("")
Dim constRadEdgeFillet2 As ConstRadEdgeFillet
Set constRadEdgeFillet2 = shapeFactory3.AddNewSolidEdgeFilletWithConstantRadius(reference35, catTangencyFilletEdgePropagation, r)
Dim reference36 As Reference
Set reference36 = part3.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;1);None:();Cf11:());Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket3)
constRadEdgeFillet2.AddObjectToFillet reference36
constRadEdgeFillet2.EdgePropagation = catTangencyFilletEdgePropagation
Dim reference37 As Reference
Set reference37 = part3.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());Face:(Brp:(Pad.1;1);None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;
```

```
WithoutBuildError;WithSelectingFeatureSupport;MF
BRepVersion_CXR15)", pocket3)
constRadEdgeFillet2.AddObjectToFillet reference37
constRadEdgeFillet2.EdgePropagation =
catTangencyFilletEdgePropagation
part3.UpdateObject constRadEdgeFillet2

part3.Update

Dim product3 As Product
Set product3 = partDocument3.GetItem("Part3")
product3.PartNumber = "arandelaeje"
partDocument3.SaveAs objPath &
"\arandelaeje.CATPart"
partDocument3.Close

'====================================
'----------------------------------Arandela de soporte------
'====================================
Dim documents4 As Documents
Set documents4 = CATIA.Documents
Dim partDocument4 As PartDocument
Set partDocument4 = documents4.Add("Part")
Dim part4 As Part
Set part4 = partDocument4.Part
Dim bodies4 As Bodies
Set bodies4 = part4.Bodies
Dim body4 As Body
Set body4 = bodies4.Item("PartBody")
Dim sketches4 As Sketches
Set sketches4 = body4.Sketches
Dim originElements4 As OriginElements
Set originElements4 = part4.OriginElements
Dim reference38 As Reference
Set reference38 = originElements4.PlaneYZ
Dim sketch7 As Sketch
Set sketch7 = sketches4.Add(reference38)

Dim arrayOfVariantOfDouble7(8)
arrayOfVariantOfDouble7(0) = 0#
arrayOfVariantOfDouble7(1) = 0#
arrayOfVariantOfDouble7(2) = 0#
arrayOfVariantOfDouble7(3) = 0#
arrayOfVariantOfDouble7(4) = 1#
arrayOfVariantOfDouble7(5) = 0#
arrayOfVariantOfDouble7(6) = 0#
arrayOfVariantOfDouble7(7) = 0#
arrayOfVariantOfDouble7(8) = 1#
Set sketch7Variant = sketch7
sketch7Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble7
part4.InWorkObject = sketch7

Dim factory2D7 As Factory2D
Set factory2D7 = sketch7.OpenEdition()
Dim geometricElements7 As GeometricElements
Set geometricElements7 =
sketch7.GeometricElements

Dim axis2D7 As Axis2D
Set axis2D7 =
geometricElements7.Item("AbsoluteAxis")
Dim line2D17 As Line2D
Set line2D17 = axis2D7.GetItem("HDirection")
line2D17.ReportName = 1
Dim line2D18 As Line2D
Set line2D18 = axis2D7.GetItem("VDirection")
line2D18.ReportName = 2

Dim circle2D6 As Circle2D
Set circle2D6 = factory2D7.CreateClosedCircle(0#,
0#, Dc / 2)
Dim point2D10 As Point2D
Set point2D10 = axis2D7.GetItem("Origin")
circle2D6.CenterPoint = point2D10
circle2D6.ReportName = 3

Dim constraints7 As Constraints
Set constraints7 = sketch7.Constraints
Dim reference39 As Reference
Set reference39 =
part4.CreateReferenceFromObject(circle2D6)
Dim constraint14 As Constraint
Set constraint14 =
constraints7.AddMonoEltCst(catCstTypeRadius,
reference39)
constraint14.Mode = catCstModeDrivingDimension
Dim length13 As Length
Set length13 = constraint14.Dimension
length13.Value = Dc / 2

sketch7.CloseEdition
part4.InWorkObject = sketch7
part4.UpdateObject sketch7

Dim shapeFactory4 As ShapeFactory
Set shapeFactory4 = part4.ShapeFactory
Dim pad4 As Pad
Set pad4 = shapeFactory4.AddNewPad(sketch7, B /
2)
pad4.IsSymmetric = True
Dim limit7 As Limit
Set limit7 = pad4.FirstLimit
Dim length14 As Length
Set length14 = limit7.Dimension
length14.Value = B / 2

part4.UpdateObject pad4

Dim reference40 As Reference
Set reference40 =
part4.CreateReferenceFromName("Selection_RSur:(
Face:(Brp:(Pad.1;2);None:();Cf11:());Pad.1_ResultO
UT;Z0;G3055)")

Dim sketch8 As Sketch
Set sketch8 = sketches4.Add(reference40)
Dim arrayOfVariantOfDouble8(8)
arrayOfVariantOfDouble8(0) = B / 2
arrayOfVariantOfDouble8(1) = 0#
arrayOfVariantOfDouble8(2) = 0#
arrayOfVariantOfDouble8(3) = 0#
arrayOfVariantOfDouble8(4) = 1#
arrayOfVariantOfDouble8(5) = 0#
arrayOfVariantOfDouble8(6) = 0#
arrayOfVariantOfDouble8(7) = 0#
arrayOfVariantOfDouble8(8) = 1#
Set sketch8Variant = sketch8
sketch8Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble8
part4.InWorkObject = sketch8

Dim factory2D8 As Factory2D
Set factory2D8 = sketch8.OpenEdition()
```

```
Dim geometricElements8 As GeometricElements
Set geometricElements8 = sketch8.GeometricElements

Dim axis2D8 As Axis2D
Set axis2D8 = geometricElements8.Item("AbsoluteAxis")
Dim line2D19 As Line2D
Set line2D19 = axis2D8.GetItem("HDirection")
line2D19.ReportName = 1
Dim line2D20 As Line2D
Set line2D20 = axis2D8.GetItem("VDirection")
line2D20.ReportName = 2

Dim circle2D7 As Circle2D
Set circle2D7 = factory2D8.CreateClosedCircle(0#, 0#, D1 / 2)
Dim point2D11 As Point2D
Set point2D11 = axis2D8.GetItem("Origin")
circle2D7.CenterPoint = point2D11
circle2D7.ReportName = 3

Dim constraints8 As Constraints
Set constraints8 = sketch8.Constraints
Dim reference41 As Reference
Set reference41 = part4.CreateReferenceFromObject(circle2D7)
Dim constraint15 As Constraint
Set constraint15 = constraints8.AddMonoEltCst(catCstTypeRadius, reference41)
constraint15.Mode = catCstModeDrivingDimension
Dim length15 As Length
Set length15 = constraint15.Dimension
length15.Value = D1 / 2

sketch8.CloseEdition
part4.InWorkObject = sketch8
part4.UpdateObject sketch8

Dim pocket4 As Pocket
Set pocket4 = shapeFactory4.AddNewPocket(sketch8, B / 2)
Dim limit8 As Limit
Set limit8 = pocket4.FirstLimit
limit8.LimitMode = catUpToLastLimit
part4.UpdateObject pocket4

Dim reference42 As Reference
Set reference42 = part4.CreateReferenceFromName("")
Dim constRadEdgeFillet3 As ConstRadEdgeFillet
Set constRadEdgeFillet3 = shapeFactory4.AddNewSolidEdgeFilletWithConstantRadius(reference42, catTangencyFilletEdgePropagation, r)
Dim reference43 As Reference
Set reference43 = part4.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;0:(Brp:(Sketch.1;3)));None:();Cf11:());Face:(Brp:(Pad.1;2);None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket4)
constRadEdgeFillet3.AddObjectToFillet reference43
constRadEdgeFillet3.EdgePropagation = catTangencyFilletEdgePropagation
Dim reference44 As Reference
Set reference44 = part4.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(Pad.1;2);None:();Cf11:());Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithTemporaryBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", pocket4)
constRadEdgeFillet3.AddObjectToFillet reference44
constRadEdgeFillet3.EdgePropagation = catTangencyFilletEdgePropagation
part4.UpdateObject constRadEdgeFillet3
part4.Update

Dim product4 As Product
Set product4 = partDocument4.GetItem("Part4")
product4.PartNumber = "arandelasoporte"
partDocument4.SaveAs objPath & "\arandelasoporte.CATPart"
partDocument4.Close
'====================================
'----------------------------------ENSAMBLAJE------
'====================================
Dim documents5 As Documents
Set documents5 = CATIA.Documents
Dim productDocument1 As ProductDocument
Set productDocument1 = documents5.Add("Product")
Dim product5 As Product
Set product5 = productDocument1.Product
Dim products1 As Products
Set products1 = product5.Products

Dim arrayOfVariantOfBSTR1(0)
arrayOfVariantOfBSTR1(0) = objPath & "\arandelasoporte.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR1, "All"

Dim constraints9 As Constraints
Set constraints9 = product5.Connections("CATIAConstraints")
'fijamos el aro1
Dim reference45 As Reference
Set reference45 = product5.CreateReferenceFromName("Product5/arandelasoporte.1/!Product5/arandelasoporte.1/")
Dim constraint16 As Constraint
Set constraint16 = constraints9.AddMonoEltCst(catCstTypeReference, reference45)

Dim arrayOfVariantOfBSTR2(0)
arrayOfVariantOfBSTR2(0) = objPath & "\arandelaeje.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR2, "All"

Dim reference46 As Reference
Set reference46 = product5.CreateReferenceFromName("Product5/arandelasoporte.1/!Axis:(Selection_RSur:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055))")
Dim reference47 As Reference
```

```
Set reference47 = product5.CreateReferenceFromName("Product5/arandelaeje.1/!Axis:(Selection_RSur:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055))")
Dim constraint17 As Constraint
Set constraint17 = constraints9.AddBiEltCst(catCstTypeOn, reference46, reference47)

Dim reference48 As Reference
Set reference48 = product5.CreateReferenceFromName("Product5/arandelasoporte.1/!Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055)")
Dim reference49 As Reference
Set reference49 = product5.CreateReferenceFromName("Product5/arandelaeje.1/!Selection_RSur:(Face:(Brp:(Pad.1;1);None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055)")
Dim constraint18 As Constraint
Set constraint18 = constraints9.AddBiEltCst(catCstTypeDistance, reference48, reference49)
Dim length16 As Length
Set length16 = constraint18.Dimension
length16.Value = -H
constraint18.Orientation = catCstOrientOpposite

Dim arrayOfVariantOfBSTR3(0)
arrayOfVariantOfBSTR3(0) = objPath & "\agujas.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR3, "All"

Dim arrayOfVariantOfBSTR4(0)
arrayOfVariantOfBSTR4(0) = objPath & "\jaula.CATPart"
Set products1Variant = products1
products1Variant.AddComponentsFromFiles arrayOfVariantOfBSTR4, "All"

Dim reference50 As Reference
Set reference50 = product5.CreateReferenceFromName("Product5/arandelasoporte.1/!Axis:(Selection_RSur:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055))")
Dim reference51 As Reference
Set reference51 = product5.CreateReferenceFromName("Product5/jaula.1/!Axis:(Selection_RSur:(Face:(Brp:(Pocket.1;0:(Brp:(Sketch.2;3)));None:();Cf11:());CircPattern.1_ResultOUT;Z0;G3055))")
Dim constraint19 As Constraint
Set constraint19 = constraints9.AddBiEltCst(catCstTypeOn, reference50, reference51)

Dim reference52 As Reference
Set reference52 = product5.CreateReferenceFromName("Product5/arandelasoporte.1/!Selection_RSur:(Face:(Brp:(Pad.1;2);None:();Cf11:());EdgeFillet.1_ResultOUT;Z0;G3055)")
Dim reference53 As Reference
Set reference53 = product5.CreateReferenceFromName("Product5/jaula.1/!yz plane")
Dim constraint20 As Constraint
Set constraint20 = constraints9.AddBiEltCst(catCstTypeDistance, reference52, reference53)
Dim length17 As Length
Set length17 = constraint20.Dimension
length17.Value = -H / 2
constraint20.Orientation = catCstOrientSame

product5.Update
productDocument1.SaveAs objPath & "\AxialAgujas.CATProduct"
```