

Capítulo 7

Programación desarrollada

En el presente capítulo se describen cada uno de los componentes que forman parte de la aplicación *Fretting Fatigue*. Se detallan cada una de las clases de las que se hace uso, sus propiedades y sus métodos. Pero no se discute acerca del código utilizado, ya que para una misma tarea, las posibilidades son múltiples y dependen en muchos casos del criterio del programador.

Junto con el fichero de instalación de la aplicación, en el CD adjunto a este Proyecto, se proporciona el proyecto de Visual Studio.NET y el código fuente del mismo. El Capítulo 8 se dedica a las modificaciones sobre este código.

El objetivo de este capítulo es, por tanto, describir las relaciones a un nivel por debajo del proporcionado por el modelo, es decir, especificando qué objetos (con sus propiedades y métodos) “entran en juego” para realizar una determinada tarea.

7.1 Las clases *Fretting Fatigue*

La aplicación está formada por varias clases, pero se pueden distinguir tres grupos principales:

- las clases proporcionadas por la .NET Framework y que se usan directamente;
- las clases propias de la .NET Framework pero modificadas (adecuadas) a *Fretting Fatigue*; y,
- las clases implementadas de forma exclusiva para esta aplicación.

Cada uno de estos grupos engloba varias clases, y en algunos casos, éstas a su vez a otras, hasta constituir un árbol de dependencia con varios niveles. Además, no es cierto totalmente que existan clases implementadas de forma externa a la plataforma .NET (como se pretende en el tercer grupo); puesto que si recuerda el Capítulo 5 (uno de los aspectos principales de la tecnología .NET es un sistema común de tipos), toda clase implementada en .NET contiene elementos .NET y por tanto no puede ser externa a la plataforma. El resultado es por tanto, el desarrollo de clases basadas en los principios de la plataforma .NET y haciendo uso de ellos, pero según unas relaciones que sí pueden ser exclusivas y específicas.

El propósito de este apartado es enumerar y describir las clases de las que se hacen uso en la aplicación, así como sus propiedades y métodos; no se pretende alcanzar un

conocimiento exhaustivo de aquéllas que no han sido implementadas por el autor y que pertenecen a la biblioteca .NET Framework.

7.1.1 La clase Tensiones

El primer elemento que se desarrolló al inicio de este Proyecto fue la clase para el cálculo de las tensiones en un contacto esférico. Su posterior desarrollo y ampliación ha estado propiciado por la demanda de servicios de otras clases que hacían uso de ésta. Con ello, la clase Tensiones que forma parte de la versión 1.1 de *Fretting Fatigue* constituye —a pesar de no ser perfecta— uno de los grandes logros de esta programación.

El objetivo es calcular las tensiones en un contacto esférico; en vista de que se necesitará la información en otras clases, para no mantener en memoria un gran volumen de datos como pueden ser las variables de tensiones (a lo largo de una línea, en una superficie o en varios puntos, por ejemplo) se propone el cálculo puntual; es decir: para unas condiciones en el contacto dadas, en un punto de coordenadas (x, y, z) las componentes de tensión son proporcionadas por la clase Tensiones. Esto permite hacer uso de la clase para el cálculo en un punto, en una línea, en una superficie o en un volumen de material.

Las tablas 7.1 y 7.2 muestran cada uno de los componentes de la clase Tensiones. Observe que los métodos son algo muy particular de la programación llevada a cabo.

<u>Propiedades</u>	<u>Descripción</u>
R	Radio de la zona de contacto
E, ν, μ	Propiedades mecánicas (E, ν, μ)
P, Q, σ_a	Cargas (P, Q, σ_a)
x_a, y_a, z_a	Coordenadas adimensionalizadas
x_c, y_c, z_c	Coordenadas respecto de la zona de adhesión
ca	Coefficiente de tensiones, p_0 .
$ca2$	Coefficiente de tensiones tangenciales, q_0
n, m	Índices para los bucles de cálculo
$precision$	Variable para recoger la precisión del cálculo (mínimo incremento en la coordenada dinámica)
exc	Excentricidad
sx, sy, sz	Componentes normales del tensor de tensiones

<u>Propiedades</u>	<u>Descripción</u>
t_{xy}, t_{yz}, t_{xz}	Componentes tangenciales del tensor de tensiones
a, c	Radio de la zona de contacto (“ a ”) y de adhesión (“ c ”)

Tabla 7.1: La clase Tensiones. Propiedades.

<u>Método</u>	<u>Descripción</u>
<i>calculaRadio</i>	Calcula el valor del radio de la zona de contacto y lo almacena en la variable “ a ”
<i>calculaTensiones1</i>	Calcula las tensiones en el contacto a lo largo de una curva: (1) debida a la carga normal (2) debida a la carga tangencial (3) debida a la carga normal y tangencial (4) debida a la zona de adhesión
<i>calculaTensiones2</i>	
<i>calculaTensiones3</i>	
<i>calculaTensiones4</i>	
<i>calculaX</i>	Incrementa la variable correspondiente para proseguir con el cálculo a lo largo de una línea.
<i>calculaY</i>	
<i>calculaZ</i>	
<i>calculaPOyQO</i>	Calcula los coeficientes de tensión normal y tangencial.

Tabla 7.2: La clase Tensiones. Métodos.

7.1.2 La clase Form1

Esta clase permite implementar el Módulo de Tensiones que a su vez depende de la clase anterior; de alguna manera gestiona el cálculo de las tensiones a lo largo de una línea o en un punto. Del mismo modo, puede utilizarse para calcular las tensiones en una superficie o en un volumen de material (en versiones posteriores a la 1.1).

El objetivo de la clase es proporcionar una interfaz gráfica que sea atractiva, y que al mismo tiempo facilite la adquisición de datos y la muestra de los resultados.

NOTA:

Las clases que hacen uso de un formulario de tipo Windows heredan las características de una clase de la biblioteca .NET (la clase *Windows::Forms*). Pero en realidad no heredan todas sus características, sino que sólo toman las que son requeridas; por ejemplo, si necesita un cuadro o una barra de título la toma, si no es necesaria, no. Visual Studio.NET gestiona automáticamente los recursos requeridos de forma visual, haciendo mucho más sencilla la tarea, y permitiendo una programación mucho más directa.

Las tablas 7.3 y 7.4 muestran las características específicamente implementadas para la clase Form1; ésta dispone de otras muchas, pero son las propias de la clase Forms y se omiten por tanto. Observe, por otro lado, que hay algunos métodos que no se corresponden con nombre alguno; se trata de métodos que se asignan a controles propios de la clase Forms de la biblioteca .NET; es el caso de los botones, los “checkBox” o cajas de validación entre otros. Estos métodos no tienen un nombre propio, sino que están asignados a “eventos” de los controles que realizan la tarea (por ejemplo, el método “button1_Click” está asociado al control “button1” y al evento “Click”), y por tanto el nombre puede sufrir variaciones en función de la programación. En estos casos, lo que aparece únicamente es la descripción de su función.

<u>Propiedades</u>	<u>Descripción</u>
<i>datos1</i>	Se trata de un listado de cadenas de texto que proviene de la lectura del fichero de Datos Iniciales.
<i>leeBdD</i>	Propiedad de tipo decisión (“bool”) que permite conocer si los datos del material provienen de la base de datos.

Tabla 7.3: La clase Form1. Propiedades.

<u>Métodos</u>	<u>Descripción</u>
<i>selectPrecision</i>	Método para la obtención de la máxima precisión posible y de los valores estándar que aparecen en el Módulo de Tensiones para que el usuario elija.
<i>representaGrafico</i>	Método para la representación gráfica de los resultados en un formulario independiente; hace uso de la clase FormGrafico
<i>escribeDatosIni</i>	Método para la escritura de un fichero de Datos Iniciales; recopila la información proporcionada por el usuario a través del interfaz gráfico.

<u>Métodos</u>	<u>Descripción</u>
<i>escribeTensiones</i>	Método para la escritura de un fichero de resultados
--	Método para mostrar el formulario de información
--	Método para el cálculo del tensor de tensiones y su escritura en una zona del formulario del Módulo de Tensiones
--	Método para el cálculo de las tensiones a lo largo de una línea.
--	Método para obtener el nombre y la ruta de un fichero a escribir.
--	Método para la conexión a una base de datos de Microsoft Access, y la obtención de su información.

Tabla 7.4: La clase Form1. Métodos.

7.1.3 La clase FormGrafico

Esta clase es una de las más costosas de toda la programación; a pesar de su escaso contenido (representación gráfica de un conjunto de valores x-y), conlleva aceptar una serie de conceptos que no se aprecian fácilmente: los valores a representar son valores reales, positivos o negativos, sin embargo en un formulario (y en general en toda aplicación informática) la información gráfica está a nivel de píxel, con lo cual la tarea realmente más compleja es la conversión de un tipo (el real) a otro (el píxel).

El objetivo de esta clase ya ha sido declarado, se trata de la representación gráfica de un conjunto de valores x-y procedentes de un cálculo anterior. Pero no sólo representa gráficamente ese conjunto de valores, sino que escala el gráfico en función de los valores extremos, muestra los ejes en su posición más adecuada e incluso muestra unas líneas de referencia que indica el tamaño de la zona de contacto.

La tabla 7.5 y 7.6 muestra las principales características de esta clase. Recuerde que en el fondo se trata de un formulario de Windows y que por tanto dispone de una serie de métodos propios alojado en los eventos de los controles (en particular, los eventos gráficos se recogen en un método llamado “Paint” que tiene ciertas peculiaridades, como el refresco de la pantalla con cada modificación del formulario).

<u>Propiedades</u>	<u>Descripción</u>
n	Entero que permite conocer el número de datos a representar.
x, y	Vectores que contienen los datos a representar en el eje correspondiente.
x_{max}, x_{min}	Valores extremos de los valores en abscisa que permiten escalar el gráfico.
y_{max}, y_{min}	Valores extremos de los valores en ordenadas que permiten escalar el gráfico.
x_{ref}	Valor de referencia para la zona de contacto.
$title$	Variable que permite colocar un título en la ventana gráfica.

Tabla 7.5: La clase FormGrafico. Propiedades.

<u>Métodos</u>	<u>Descripción</u>
Evento Paint	Representación de un conjunto de datos (x, y)
Evento MouseMove	Muestra el valor de la posición del cursor en el área del gráfico.

Tabla 7.6: La clase FormGrafico. Métodos.

7.1.4 La clase FormCriterios

Esta clase implementa el Módulo de Iniciación. Se trata de una clase bastante pobre, en el sentido de que lo único que hace es reconocer la ruta hacia un fichero de Datos Iniciales y llamar a la clase correspondiente del criterio de iniciación, proporcionándole dicho fichero para que sea ésta la que calcule los parámetros.

Constituye, por tanto, un elemento de enlace cuyo objetivo no es otro que garantizar la correcta comunicación de los datos entre distintas entidades, al mismo tiempo que (por tratarse de un formulario) proporciona un entorno gráfico sencillo y manejable.

La tabla 7.7 muestra el listado de los métodos que la componen. Se presenta una única tabla, ya que esta clase no dispone de propiedades específicas sino las propias de la biblioteca .NET.

<u>Métodos</u>	<u>Descripción</u>
<i>leeDatosIni</i>	Método que realiza la lectura de un fichero de Datos Iniciales y le pasa la información a otro método para que la utilice
<i>colocaDatosIni</i>	Método que recoge la información proporcionada por la lectura de un fichero de Datos Iniciales y hace uso de ella para realizar los cálculos oportunos.
---	La clase OpenFileDialog forma parte de la librería .NET y permite reclutar el nombre y la ruta de un fichero que se desee abrir. Puede ser manipulada para que se adapte a cada aplicación.

Tabla 7.7: La clase FormCriterios. Métodos.

7.1.5 La clase FormMcD, FormSWT y FormCross

En este apartado se agrupan un conjunto de clases que tienen un comportamiento muy similar. Se trata de las clases que implementan los criterios de iniciación de grietas a partir de un fichero de Datos Iniciales; para ello calculan una serie de parámetros auxiliares y el principal del criterio, partiendo de un fichero que reciben de entrada.

Existen otras clases que lo que implementan es el criterio correspondiente pero sobre un punto (al igual que en el caso de la clase Tensiones). Se trata de las clases McDiarmid, SmithWatsonTopper y Crossland; éstas no se describen a continuación porque lo interesante es su gestión, el modo de hacer uso de ellas, no el que se implementen en un punto determinado.

Las tabla 7.8 y 7.9 muestra las características comunes y diferenciales de cada una de las clases, especificando de qué tipo es cada una de ellas.

<u>Propiedades</u>	<u>Descripción</u>	<u>Tipo</u>
<i>Path</i>	Variable que alberga el nombre y la ruta hacia un fichero de Datos Iniciales sobre el que se está calculando.	Común
<i>s, s2</i>	Matrices para el calculo de las tensiones a lo largo de la línea definida en el fichero de Datos Iniciales.	Común
<i>cs</i>	Índice que determina la variable dinámica que aparece en el fichero de Datos Iniciales	Común
<i>E, un, mu</i>	Propiedades mecánicas del material (E, ν , μ)	Común

<u>Propiedades</u>	<u>Descripción</u>	<u>Tipo</u>
<i>sy, su, sfat</i>	Límite elástico, límite de rotura y límite de fatiga respectivamente	Común
<i>sf_tor, sf_flx</i>	Límite de fatiga a torsión y flexión	Común
<i>sf, b</i>	Coefficiente y exponente de resistencia a fatiga.	Común
<i>xa, ya, za</i>	Variables que indican el máximo del parámetro a lo largo de la línea de cálculo según el fichero de Datos Iniciales.	Común
<i>tmax, smax</i>	Vectores para alojar los valores del rango de tensión tangencial máximo y la tensión normal máxima asociada.	FormMcD
<i>alpha</i>	Vector para alojar el valor del plano crítico en cada punto	FormMcD
<i>emax, smax</i>	Vectores para los parámetros auxiliares del criterio de Smith-Watson-Topper.	FormSWT
<i>swt</i>	Vector para alojar el valor del parámetro del criterio de Smith-Watson-Topper	FormSWT
<i>swt_cross</i>	Valor máximo del criterio Smith-Watson-Topper a lo largo de la línea definida en el fichero de Datos Iniciales.	FormSWT
<i>cross</i>	Matriz para albergar los valores del parámetro en el criterio de Crossland (para la línea del fichero de Datos Iniciales)	FormCross
<i>cross_max</i>	Valor extremo del parámetro	FormCross

Tabla 7.8: Las clases FormMcD, FormSWT y FormCross. Propiedades.

<u>Métodos</u>	<u>Descripción</u>	<u>Tipo</u>
<i>leeDatosIni</i>	Método para la lectura de un fichero de Datos Iniciales proporcionado como entrada, y que genera información para que otro método la interprete.	Común
<i>colocaDatosIni</i>	Método para interpretar la información procedente del método anterior y la ubica de forma adecuada para realizar los cálculos.	Común
<i>escribeResultados</i>	Método para la escritura de los resultados proporcionados por cada uno de los criterios de iniciación.	Común
<i>calculaVida</i>	Método para obtener la predicción de vida por modelos basados en la iniciación.	Común

<u>Métodos</u>	<u>Descripción</u>	<u>Tipo</u>
<i>nfichero</i>	Método para obtener el nombre de un fichero de resultados a través de formularios del tipo SaveFileDialog.	Común
<i>calculaTmax</i>	Método para calcula el rango de variación máximo de la tensión tangencial a lo largo de una línea.	FormMcD
<i>calculaTmaxP</i>	Método para calcula el rango de variación máximo de la tensión tangencial en un punto.	FormMcD
<i>calculaTmaxPK</i>	Método para calcula el rango de variación máximo de la tensión tangencial en un punto y para una orientación dada.	FormMcD
<i>calculaSmax</i>	Método para calcular la tensión normal máxima asociada a una orientación dada.	FormMcD
<i>calculaTensionMD</i>	Método para calcular la tensión equivalente de McDiarmid.	FormMcD
<i>calculaEmax</i>	Método para calcular el parámetro SWT a lo largo de una línea.	FormSWT
<i>calculaEmaxP</i>	Método para calcular el parámetro SWT en un punto.	FormSWT
<i>calculaEmaxPK</i>	Método para calcular el parámetro SWT en un punto y para una orientación determinada.	FormSWT
<i>calculaDefMax</i>	Método para calcular la deformación normal máxima.	FormSWT
<i>calculaCross</i>	Método para calcular el parámetro del criterio de Crossland, a lo largo de una línea definida en el fichero de Datos Iniciales.	FormCross

Tabla 7.9: Las clases FormMcD, FormSWT y FormCross. Métodos.

7.1.6 La clase FormProp

Esta clase implementa el Módulo de Propagación, pero al igual que ocurría con la clase FormCriterios, hace de enlace entre las clases que implementan las distintas leyes de crecimiento y por tanto no tiene mayor interés que el de soportar la comunicación entre éstas y el usuario.

7.1.7 Las clases Paris, KitaTaka, KitaTaka2, KitaTaka3 y KitaTaka4

Estas clases implementan cada una de las leyes de crecimiento que se han estudiado en el documento Memoria. La clase Paris hace referencia a la Ley de Paris, mientras que las otras hacen lo propio con las modificaciones de dicha ley⁴.

Al igual que en el caso de los criterios de iniciación, estas clases comparten sus propiedades y métodos; es decir que tienen el mismo nombre y prácticamente la misma utilidad. Por ello, en las tablas 7.10 y 7.11 se muestran dichas características comunes.

<u>Propiedades</u>	<u>Descripción</u>
<i>tens</i>	Objeto de la clase Tensiones
<i>w</i>	Espesor de probeta del ensayo
<i>sy, su, sfat</i>	Límite elástico, límite de rotura y límite de fatiga respectivamente
<i>dkth</i>	FIT (factor de intensidad de tensiones) umbral para grieta larga.
<i>f</i>	Factor de transición de grieta corta a larga
<i>D</i>	Tamaño de grano típico
<i>alfa1, alfa2, z_paso</i>	Parámetros para el caso de cálculo con grieta no perpendicular.

Tabla 7.10: Las clases Paris, KitaTaka, KitaTaka2, KitaTaka3 y KitaTaka4. Propiedades.

<u>Métodos</u>	<u>Descripción</u>
<i>func_peso</i>	Método para obtener el valor de la función de peso de Bueckner
<i>FIT</i>	Método para calcular el FIT en cada ley.
<i>FConv, FConv2</i>	Método para convertir una profundidad en una longitud de grieta, en el caso de grieta no perpendicular.
<i>set_grieta</i>	Método para parametrizar el camino de grieta
<i>integral</i>	Permite implementar un método de integración numérico.

⁴ El nombre de la clase (KitaTaka) surge como consecuencia de acortar el de Kitagawa-Takahashi.

<u>Métodos</u>	<u>Descripción</u>
<i>tensionGrieta</i>	Método para implementar el cálculo de la tensión en el borde de grieta.
<i>leeDatosIni</i>	Método para la lectura del fichero de Datos Iniciales que se comunica como otro método para interpretar la información de lectura.
<i>colocaDatosIni</i>	Método que interpreta la información proporcionada en la lectura de un fichero de Datos Iniciales.
<i>calculaVida</i>	Método que proporciona una predicción de vida según la ley de crecimiento.
<i>correcciónKi</i>	Método que proporciona el coeficiente que permite extrapolar el FIT de una grieta 3D a una grieta plana.
<i>escribeResultados</i>	Método para la escritura de ficheros de resultados.
<i>vectorLongitud</i>	Método que determina los puntos de cálculo a lo largo del camino de grieta.

Tabla 7.11: Las clases Paris, KitaTaka, KitaTaka2, KitaTaka3 y KitaTaka4. Métodos.

El método FIT puede ser el más representativo de cada clase, ya que varía de una a otra. En el caso de la clase Paris, proporciona el valor del FIT directamente, mientras que en los casos de las leyes KitaTakaX (X= , 2, 3 ó 4) proporciona el valor del término con exponente de la ley de crecimiento correspondiente. En cualquier caso, la variación afecta a unas pocas sentencias.

7.1.8 La clase FormMixto

Esta clase es la más compleja de todas las que se presentan debido a su importancia, a su envergadura y a su ramificación. Permite implementar el Módulo Mixto de la aplicación, que puede trabajar bien con un fichero de Datos Iniciales, bien con un fichero de Datos de Ensayos.

El objetivo de esta clase es servir de soporte para los métodos desarrollados anteriormente, así como permitir la entrada de información procedente del usuario.

Las tablas 7.12 y 7.13 muestran las propiedades y métodos de la clase, pero al igual que en anteriores ocasiones, el inicio de las operaciones de cálculo tiene lugar desde un método asociado al evento de un control (hacer clic sobre un botón por ejemplo) y es por ello que algunos métodos aparecen sin nombre.

<u>Métodos</u>	<u>Descripción</u>
<i>md, swt, cross</i>	Variables de decisión que permiten disponer (o no) del cálculo de cada uno de los criterios de iniciación de grieta.
<i>paris, um1, um2, fm1, fm2</i>	Variables de decisión que hacen lo propio con las leyes de crecimientos.
<i>crit, ley</i>	Variables de decisión que permiten comprobar si la selección de criterios y leyes para el cálculo ha sido realizada por el usuario o queda pendiente.
<i>cond_met</i>	Variable de decisión
<i>alfa1, alfa2, z_paso</i>	Propiedades que permiten parametrizar el camino de grieta en dos tramos no perpendiculares.
<i>cond_grieta</i>	Variable de decisión que permite determinar el cálculo de grietas perpendiculares o no.
<i>ten1, tens2</i>	Objetos de la clase Tensiones para calcular las tensiones en el Estado A y el Estado B (ver Memoria).
<i>ley1, ley2, ley3, ley4, ley5</i>	Objetos de las clases Paris y KitaTakaX (X= ,2,3,4) para calcular la propagación de grietas según cada ley.
<i>mcd, swt1, cross1</i>	Objetos de la clase McDiarmid, SmithWatsonTopper y Crossland para el cálculo de la iniciación de grietas.
<i>ni, np</i>	Vectores para albergar la predicción de vida a iniciación y a propagación respectivamente.
<i>a</i>	Vector que alberga las distintas las profundidades de cálculo (en el caso de grieta perpendicular, coincide con la longitud de grieta)
<i>fit, fit2</i>	Vectores que almacenan los valores del FIT máximo y mínimo (respectivamente) para cada componente del vector anterior “a”.
<i>path</i>	Variable para albergar el nombre y la ruta de un fichero de Datos Iniciales o de Datos de Ensayo.
<i>pathwork</i>	Variable para almacenar la ruta hacia un directorio sobre el que volcar los resultados.
<i>listaEnsayos</i>	Variable que almacena las configuraciones de cada uno de los ensayos para proporcionarlas en el momento del cálculo.
<i>af</i>	Profundidad final de grieta (en el caso de una grieta perpendicular, coincide con la longitud final).
<i>w</i>	Espesor de probeta para el ensayo.
<i>c, n</i>	Coeficiente y exponente para la ley de crecimiento.
<i>sy, su, sfat</i>	Límite elástico, de rotura y de fatiga para el material de cálculo.

<u>Métodos</u>	<u>Descripción</u>
sf, b	Coeficiente y exponente de la relación S-N del material.
f	Factor de transición de grieta corta a larga (toma el valor 2.5 por defecto)
D	Tamaño típico de grano del material.

Tabla 7.12: La clase FormMixto. Propiedades.

<u>Métodos</u>	<u>Descripción</u>
<i>directorioTrabajo</i>	Método que proporciona la ruta hacia un directorio en el que poder volcar los resultados.
<i>nombreResultados</i>	Sobrecargado. Método que proporciona el nombre de cada fichero de resultados en función de la combinación criterio-ley con la que se haya calculado.
<i>modificaCabecera</i>	Método para preparar el fichero de resumen de ensayos.
<i>calculaResultados</i>	Sobrecargado. Método para calcular un ensayo.
<i>calculaFIT</i>	Método para calcula el FIT para cada ensayo.
<i>nSaveFile, nFichero</i>	Métodos que proporciona el nombre y la ruta de un fichero para guardar y para leer (repectivamente).
<i>leeDatosIni</i>	Método para la lectura de un fichero de Datos Iniciales.
<i>colocaDatosIni</i>	Método que interpreta la información de la lectura de un fichero de Datos Iniciales.
<i>leeDatosEnsayos</i>	Método para la lectura de un fichero de Datos de Ensayos.
<i>escribeDatosIni</i>	Método que genera fichero de Datos Iniciales para cada uno de los ensayos.
<i>crearDirectorio</i>	Método que crea directorios y subdirectorios en una ubicación proporcionada.
<i>ajusteCabecera</i>	Método que prepara el fichero de resultados
<i>calculaEnsayos</i>	Método para proceder al cálculo de los ensayos.
<i>restauraPathwork</i>	Método para modificar algunos parámetros.
<i>longGrieta</i>	Método que proporciona la profundidad de cálculo.
<i>calculaVida</i>	Método que calcula cada una de las combinaciones criterio-ley que puedan darse.

<u>Métodos</u>	<u>Descripción</u>
<i>criterio_mcd,</i> <i>criterio_swt,</i> <i>criterio_cross</i>	Métodos que calculan la predicción de vida de cada uno de los criterios de iniciación.
<i>ley_paris, ley_um1,</i> <i>ley_um2, ley_fm1,</i> <i>ley_fm2</i>	Métodos para calcular la predicción de vida para cada una de las leyes de crecimiento.
<i>escribeResultados</i>	Método para la escritura, en un fichero, de los resultados de cada combinación criterio-ley.
<i>escribeCabecera</i>	Método que complementa la escritura de un fichero de resultados.
<i>escribeMínimo</i>	Método que complementa la escritura de un fichero de resultados.
<i>escribeResumen</i>	Método que permite escribir el fichero resumen de resultados para todos los ensayos en un único archivo.
--	Métodos de comienzo el cálculo por petición del usuario.

Tabla 7.13: La clase FormMixto. Métodos.

7.1.9 Otras clases

Existe adicionalmente otras clases pero que no tienen interés más allá de su uso, ya que se trata de formularios para recolectar información sobre algunos parámetros. Es el caso, por ejemplo, de las clases FormDatos, cuyo objetivo es que el usuario completa cierta información. Otras, en cambio, proporcionan una información de poco valor (como la clase FormAbout que proporciona información sobre la versión de la aplicación) que no proviene de ningún cálculo.

Por otro lado, la clase FormInicial permite acceder a cada uno de módulos implementados; hace de ventana principal de la aplicación, pero sus funciones están restringidas a este uso.

7.2 Los procesos de *Fretting Fatigue*

En este apartado se describen algunos procedimientos por medio de diagramas de flujo; estos diagramas muestran las relaciones entre las propiedades y los métodos de cada clase con los de otras para obtener un resultado final.

No se muestran todos los procesos de cada uno de los cálculos que tienen lugar en cada módulo, sino que se muestran únicamente los procesos globales de la aplicación y del Módulo Mixto que reúne a todos los implementados (Módulo de Tensiones, de Iniciación y de Propagación).

7.2.1 El proceso global de *Fretting Fatigue*

La figura 7.1 muestra el diagrama de flujo correspondiente al este proceso. Observe que dicho diagrama comienza en la ventana inicial de la aplicación, desde la que se puede acceder a los distintos módulos.

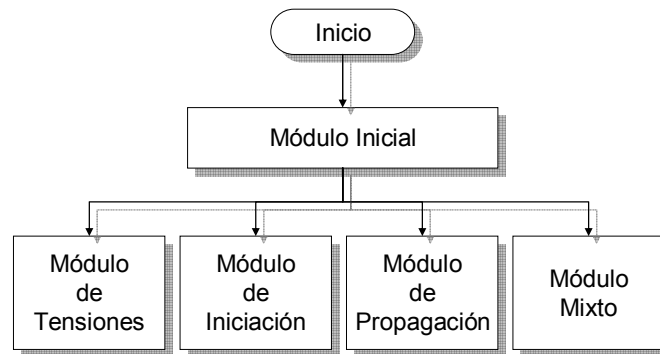


Figura 7.1: Proceso global de la aplicación.

7.2.2 El proceso de cálculo en el Módulo Mixto

La figura 7.2 muestra el diagrama de flujo de este módulo. Observe que se distingue entre dos líneas de procedimientos: una correspondiente a trabajar con un fichero de Datos Iniciales y otra con un fichero de Datos de Ensayos.

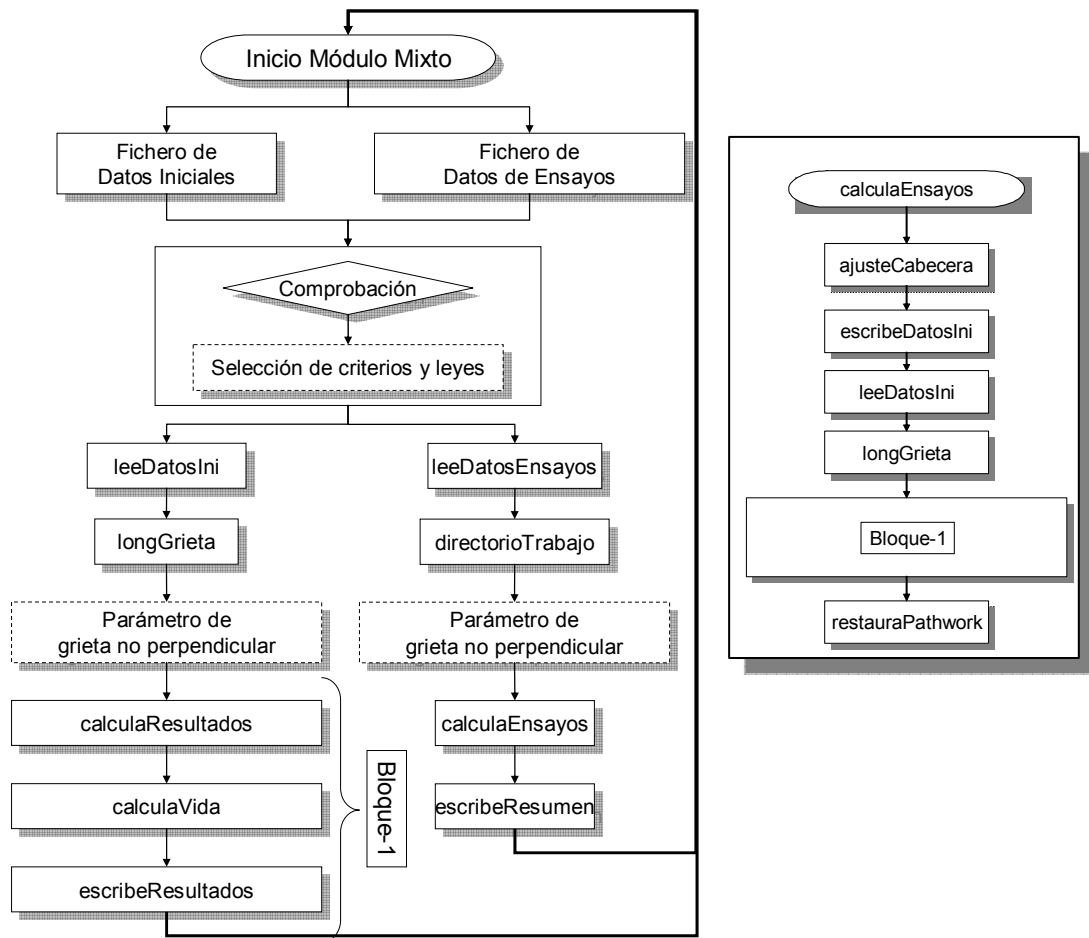


Figura 7.2: Diagrama de flujo del Módulo Mixto.

El proceso se inicia con la entrada de un tipo de fichero, continúa con la comprobación de que dicho fichero es válido y con la selección de los métodos de cálculo. En este punto observe que el proceso de selección se encuentra recuadrado de forma diferente al resto; esto es debido a que la selección puede realizarse de forma anterior al cálculo y si es así no tiene porque repetirse.

A continuación las líneas de cálculo se separan en virtud del tipo de fichero que haya introducido.

Para el caso de un fichero de Datos Iniciales, se procede a su lectura para la adquisición de datos; seguidamente, haciendo uso del método “longGrieta”, se procede a la determinación de las longitudes de grieta de cálculo (se trata en realidad de un proceso de cálculo discreto, es decir, en determinados puntos cercanos, y no continuo). Aparece a continuación el proceso de adquisición de datos para parametrizar el camino de grieta, observe que (de igual forma que en la selección de los métodos) aparece nuevamente con el contorno discontinuo, esto hace ver que no tiene porque llevarse a cabo este paso y depende de las opciones que se haya establecido de antemano.

El “Bloque-1” contiene una serie de procesos que serán iguales para varios pasos. En particular, contiene el cálculo de los resultados; para ello en el método “calculaVida” se lleva a cabo el cálculo para una combinación de criterio-ley determinada, de forma que el método “calculaResultado” le proporcione dicha combinación en función de las opciones que el usuario haya seleccionado. Por último, el método “escribeResultados” permite obtener un fichero con los valores de las variables calculadas: la predicción de vida.

En el caso de un fichero de Datos de Ensayos, el procedimiento es básicamente el mismo, con la diferencia de que debe disponerse de un fichero de Datos Iniciales para cada uno de los ensayos. Así pues, el método “leeDatosEnsayos” lee el fichero de Datos de Ensayos y almacena la información en memoria, para que, una vez seleccionado un directorio de trabajo (método “directorioTrabajo”) y los parámetros del camino de grieta (caso de ser necesarios por la petición del usuario), otro método (“calculaEnsayos”) realice las modificaciones necesarias para la creación del fichero de Datos Iniciales de cada ensayo. Ahora el control, recorre todos los ensayos realizando el proceso anteriormente descrito para un único ensayo, de forma que el resultado final se muestra en las subcarpetas contenidas en el directorio de trabajo; adicionalmente, el método “escribeResumen” realiza la tarea de crear un fichero con los resultados más representativos de cada ensayo para los distintos métodos de cálculo.