Capítulo 8

IMPLEMENTACIÓN

8.1 Introducción

A continuación vamos a explicar los cambios que han sido necesario realizar en la Base de Datos de partida para contemplar la gestión de los identificadores. En dicha base de datos no se gestionaba por ejemplo el estado de los identificadores, ni se almacenaban correctamente de acuerdo al estándar del CEN.

Tras esto mostraremos el funcionamiento detallado de los diferentes métodos implementados, tanto de la interfaz IdMgr como de la interfaz que se ha definido.

8.2 Cambios en la Base de Datos

En la Base de datos de partida, como hemos dicho antes, no se gestionaba ningún tipo de identificador, por lo que se han definido las siguientes tablas:

Tabla PersonIdentifiers:

Tabla que almacena los PersonId de acuerdo al tipo de datos InstanceIdentifier del CEN

campo	tipo SQL	clave externa
extension	VARCHAR(32)	Person(PersonId)
low	TIMESTAMP	
high	TIMESTAMP	
low_closed	BIT	
high_closed	BIT	
width	VARCHAR(32)	
state	VARCHAR(32)	
preferredId	VARCHAR(32)	

Tabla AdministrativeIdentifiers:

Tabla que almacena los identificadores administrativos de según el tipo de datos InstanceIdentifier del estándar CEN

campo	tipo SQL	clave externa
extension	VARCHAR(32)	
root	VARCHAR(32)	assigningAuthorityName(root)
low	TIMESTAMP	
high	TIMESTAMP	
low_closed	BIT	
high_closed	BIT	
width	VARCHAR(32)	
preferredId	VARCHAR(32)	Person (PersonId)
PersonRole	VARCHAR(8)	RoleCodes(codeValue)

Tabla AssigningAuthorityName:

Almacena los OID y el nombre de las entidades asignadoras.

campo	tipo SQL	clave externa
root	VARCHAR(32)	
assigningAuthorityName	VARCHAR(32)	

Tabla InfoPID:

Almacena información necesaria para el funcionamiento de la gestión de identificadores. Está diseñada como una tabla que almacena el nombre de una variable y su valor. Entre otras cosas debe almacenar:

campo	tipo SQL	clave externa
nombre	VARCHAR(32)	
valor	VARCHAR(32)	

- LocalDomainOID: almacena el OID del Dominio Local

- Confidence: contiene el valor del parámetro confidence, utilizado en el método find_or_register_ids. Es el umbral de decisión para ver si la búsqueda devuelve un perfil considerado igual al que se intenta registrar.

- DefaultIVL: contiene, expresado como número de meses, del periodo de validez que se asigna a los identificadores generados en el dominio.

- LastPersonId: almacena el último identificador que ha sido generado.

- MandatoryTraits: está variable guarda los nombres de los Traits que han de estar registrados para una persona para que su PersonId pueda pasar a estado PERMANENT.

8.3 Interfaz IdMgr

Mostramos una breve descripción de los métodos implementados, de los parámetros que utilizan y las excepciones que pueden lanzar. Así como diagramas UML para facilitar la comprensión de su funcionamiento.

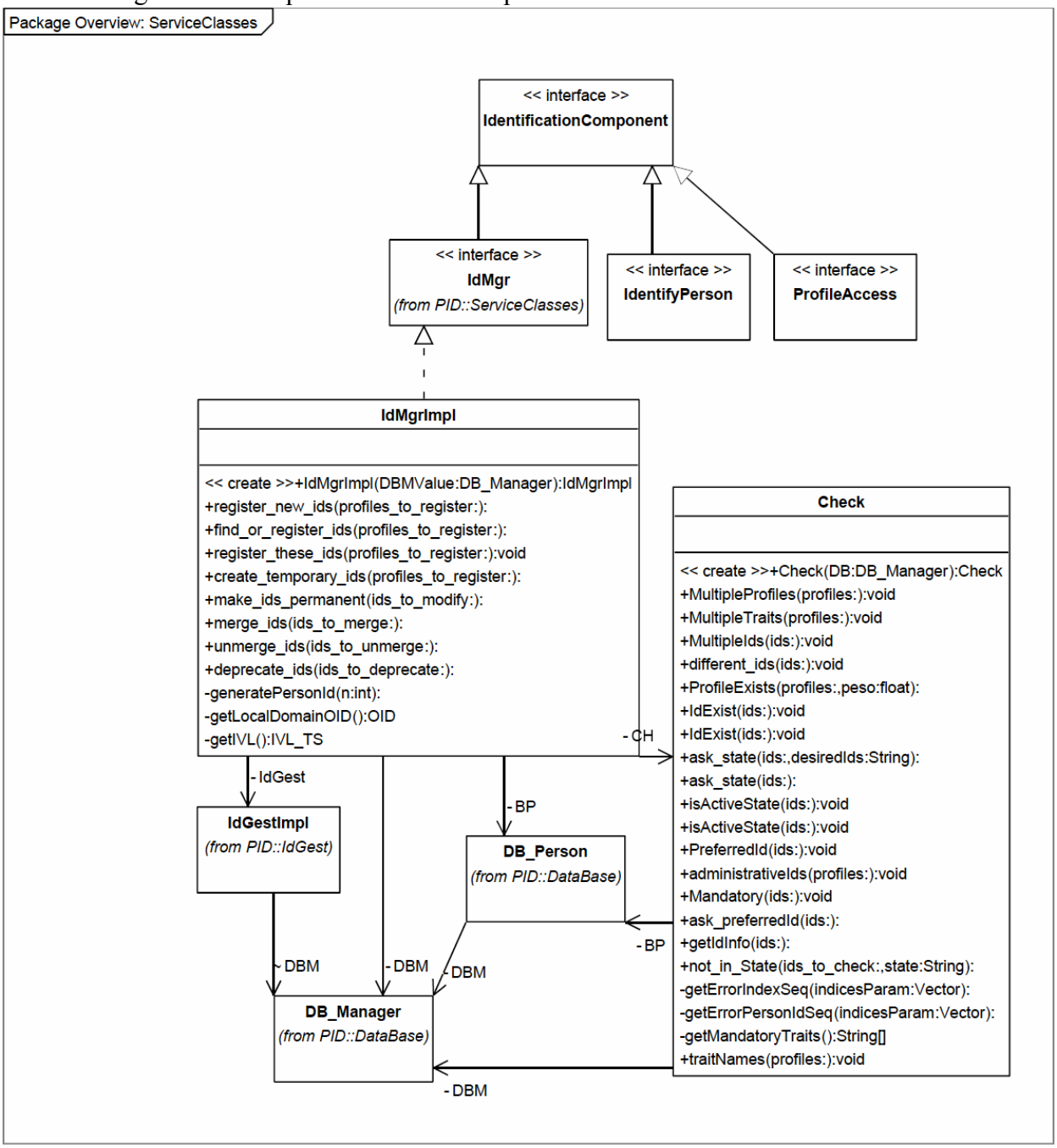


Figura 8.1 Diagrama de clases de la interfaz IdMgr

register_new_ids(profileSeq profiles)

Genera nuevos identificadores y les asocia el perfil que se pasa.
Comprueba si en los perfiles que se pasan hay Traits repetidos o perfiles duplicados.
Parámetros: secuencia de perfiles a registrar.
Devuelve: La secuencia de PersonIds generada
Excepciones: ProfilesExist, DuplicateProfiles, MultipleTraits, InvalidAdministrativeIds.

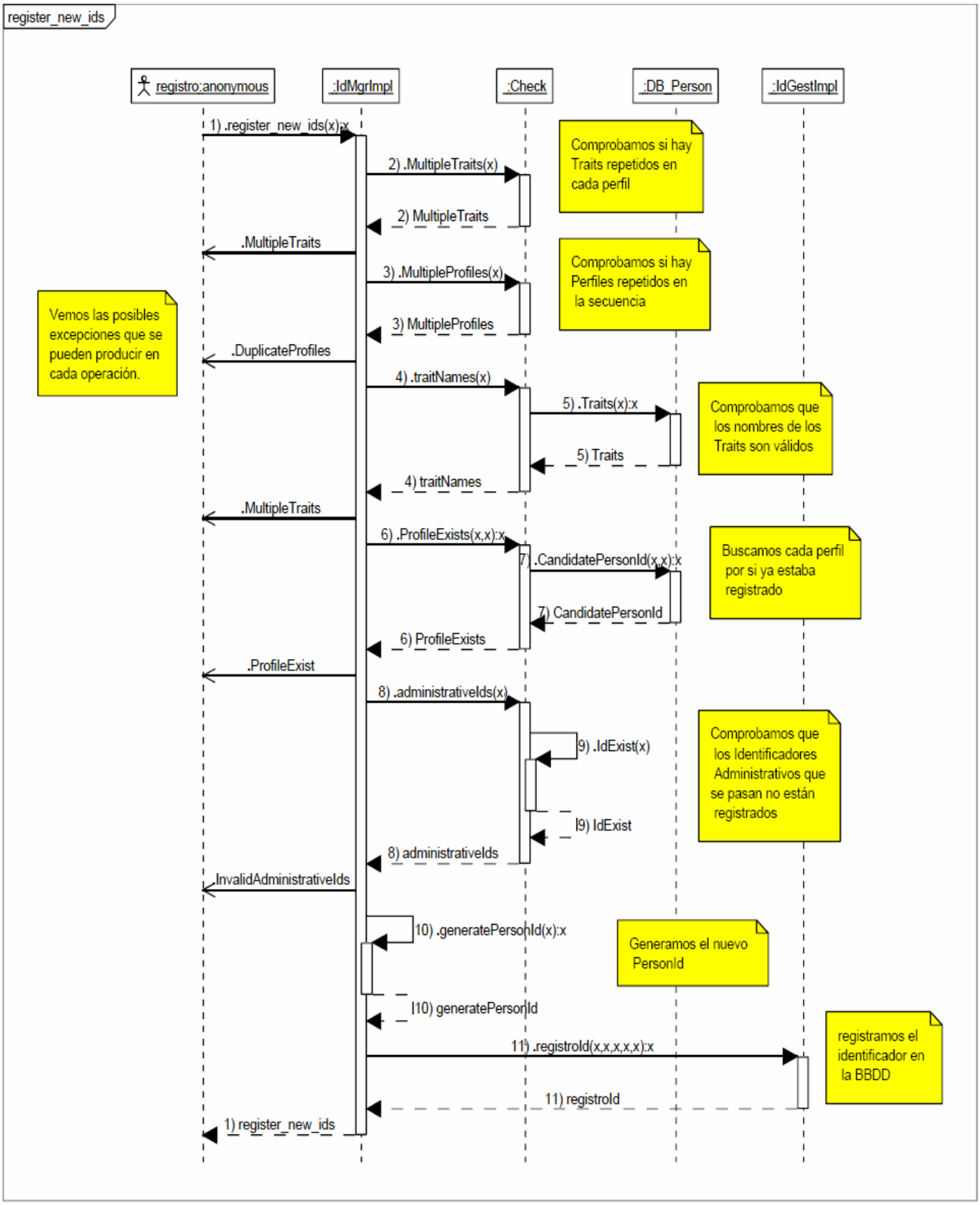


Figura 8.2 Diagrama de secuencia de register_new_ids

find_or_register_ids(ProfileSeq profiles)
Genera identificadores de forma automática. La variable interna confidence de la tabla InfoPID controla el grado de confianza para decidir si el perfil ya existe en el dominio o si es necesario generar un nuevo identificador.
Parámetros: secuencia de perfiles a registrar.
Devuelve: La secuencia de PersonIds generada
Excepciones: DuplicateProfiles,MultipleTraits, InvalidAdministrativeIds.

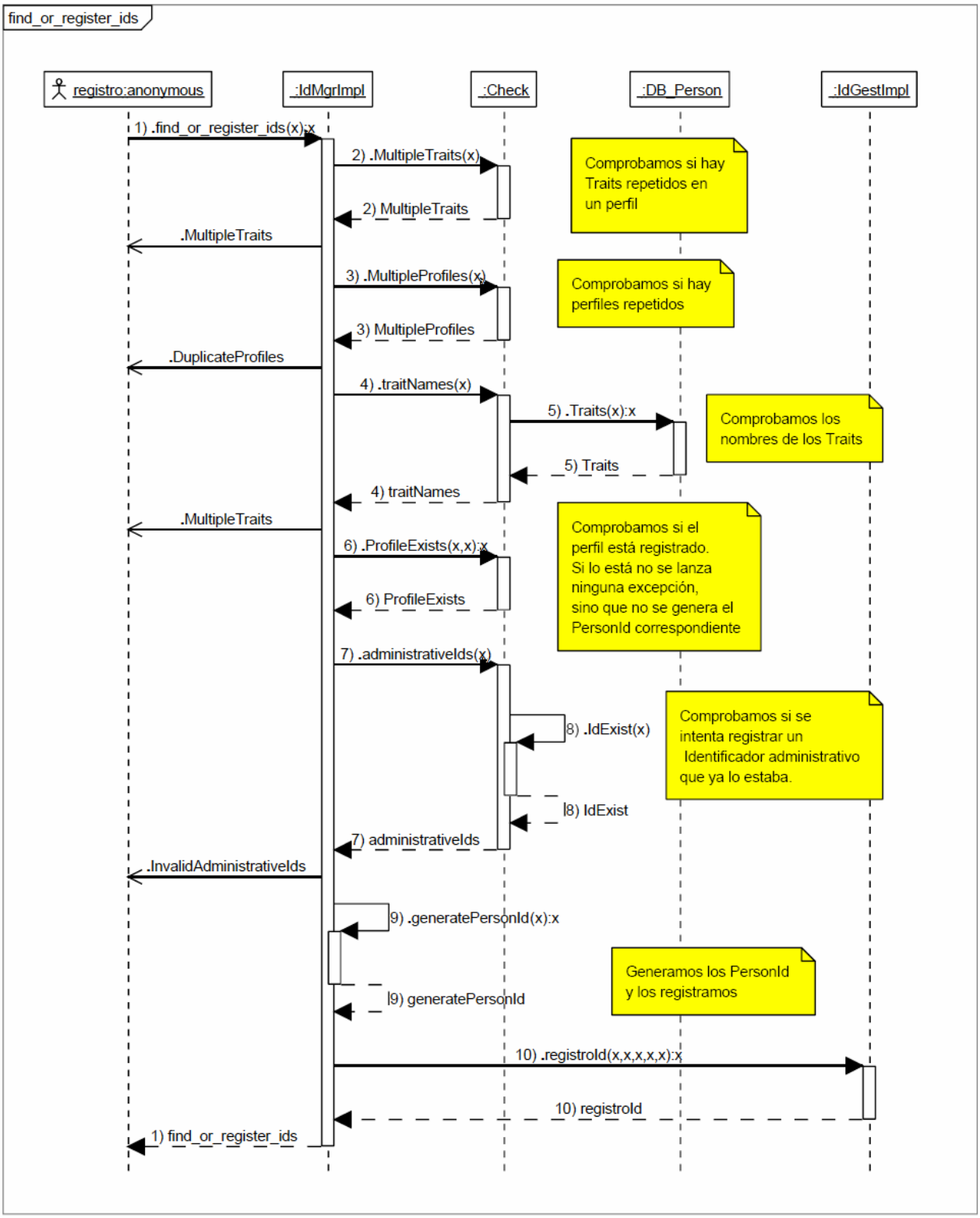


Figura 8.4 Diagrama de secuencia de find_or_register_ids

create_temporary_ids()
Genera nuevos identificadores en estado Temporary. Se generan incluso si el perfil que se pasa se corresponde a uno existente.
Parámetros: secuencia de perfiles a registrar.
Devuelve: La secuencia de PersonIds generada
Excepciones: MultipleTraits.

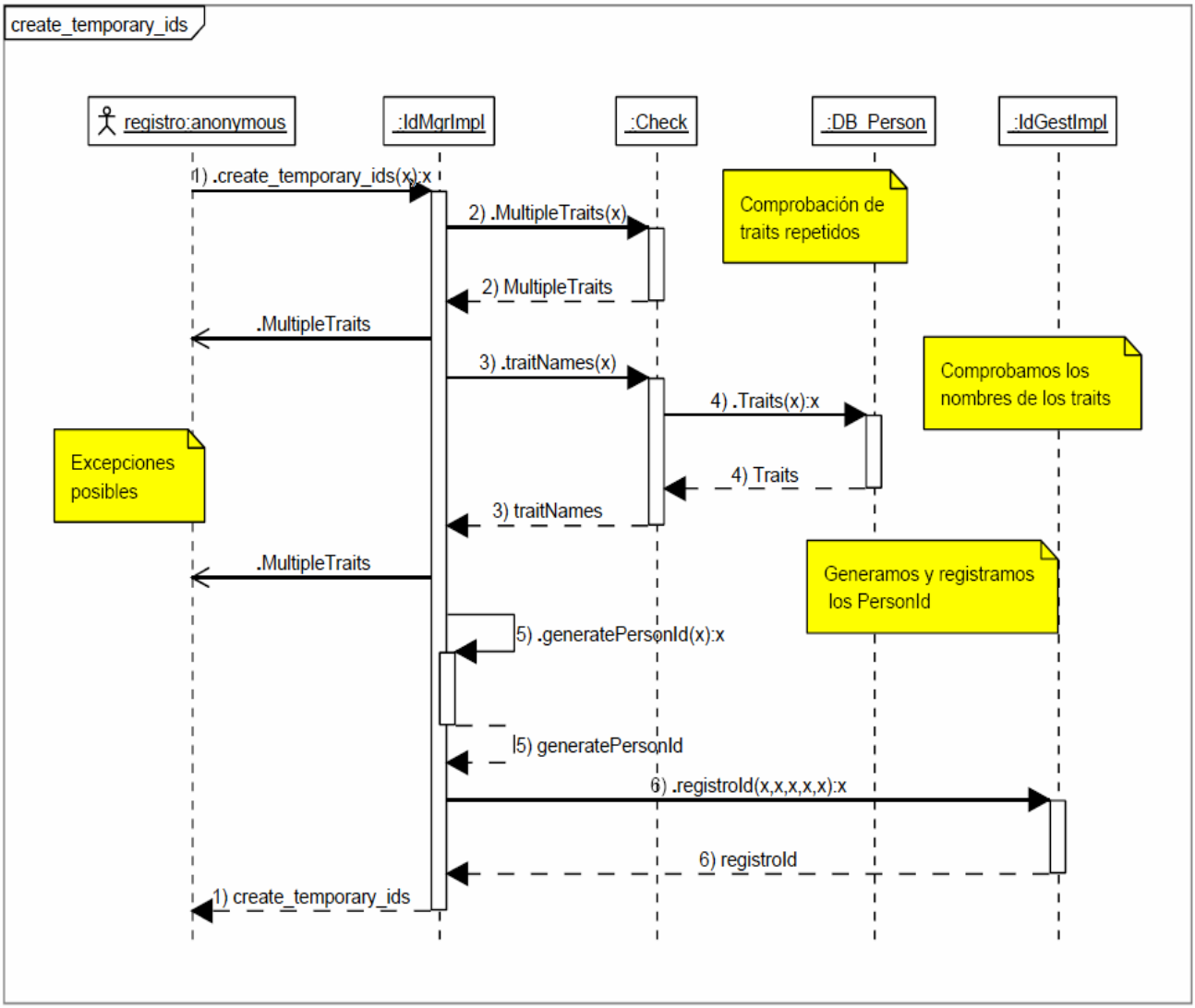


Figura 8.5 Diagrama de secuencia de create_temporary_ids

make_ids_permanent()
Cambia el estado de un identificador a permanente. Verificando que están registrados los traits obligatorios.
Parámetros: secuencia de perfiles a modificar.
Devuelve: La secuencia de PersonIds que se han cambiado.
Excepciones: InvalidIds, DuplicateIds, RequiredTraits, MultipleTraits.

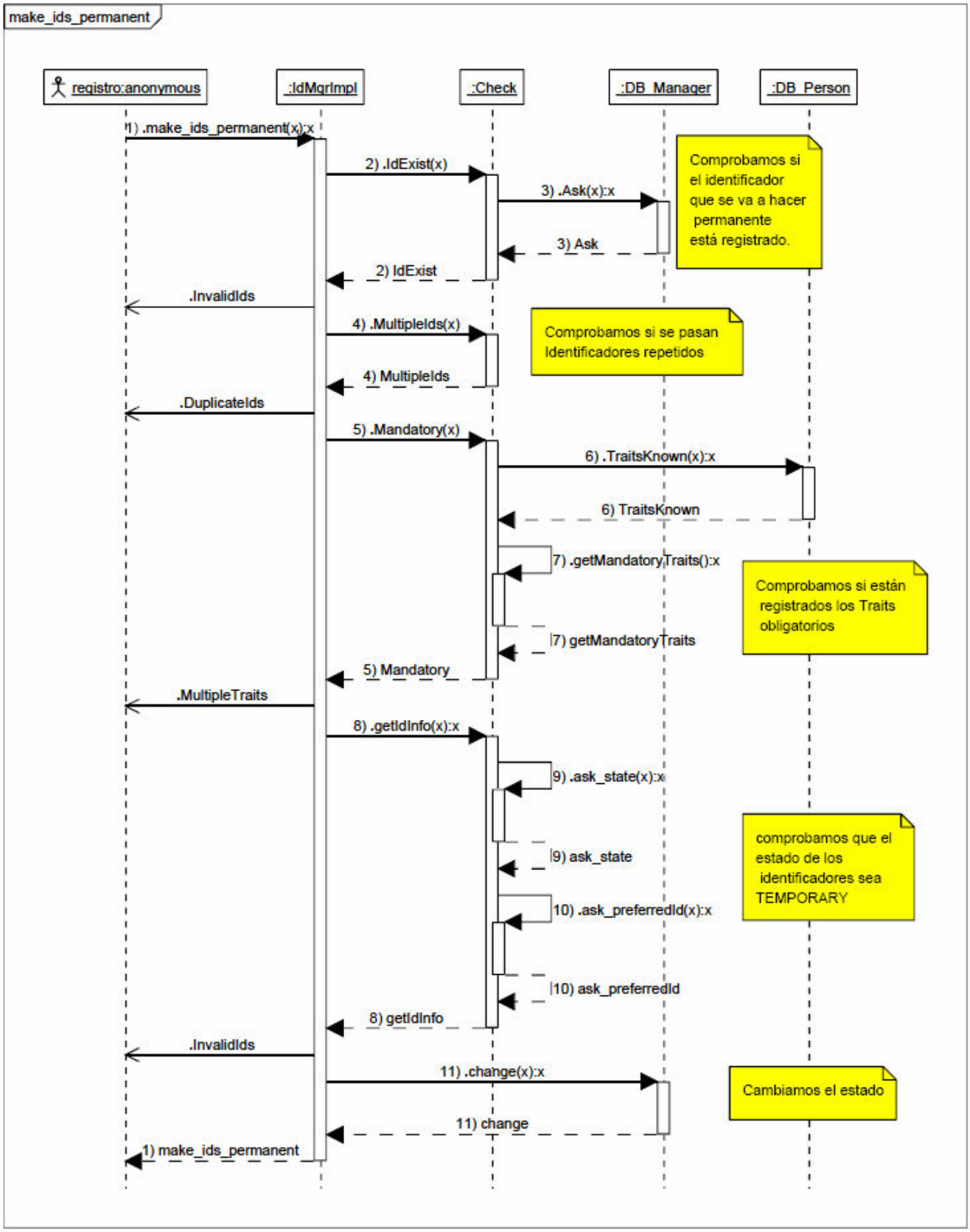


Figura 8.6 Diagrama de secuencia de make_ids_permanent

merge_ids(MergeStructSeq ids_to_merge)
Si una persona tiene mas de un Id en el mismo dominio se fusionan en uno sólo.
Parámetros: secuencia de Mergestruct que contienen los perfiles a fusionar.
Devuelve: Secuencia de IdInfo de los PersonIds que se han cambiado.
Excepciones: InvalidIds, DuplicateIds.

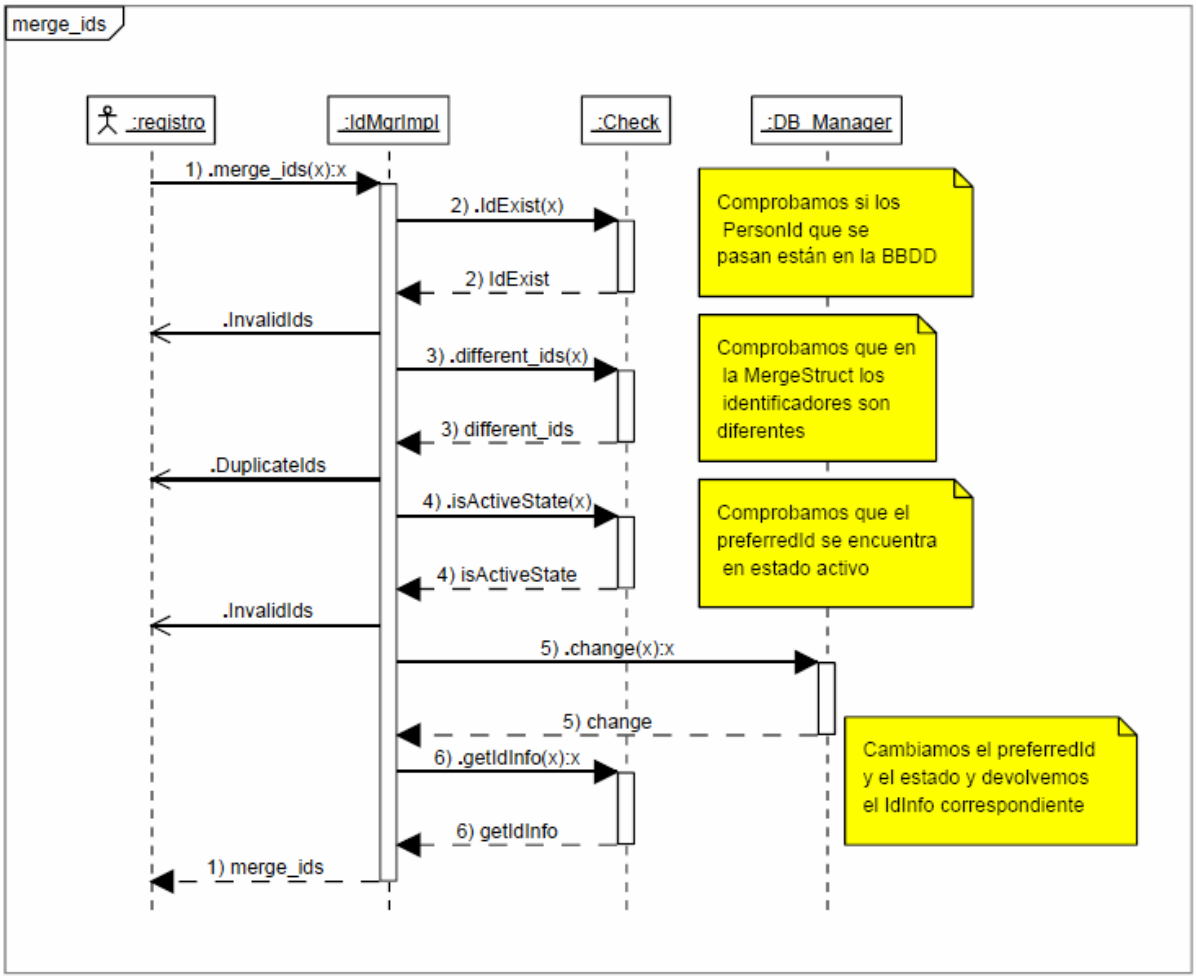


Figura 8.7 Diagrama de secuencia de merge_ids

unmerge_ids(PersonIdSeq ids_to_unmerge)
Deshace la fusión de 2 identificadores debido a que representan a personas distintas.
Parámetros: secuencia de PersonIds.
Devuelve: Secuencia de IdInfo de los PersonIds que se han cambiado.
Excepciones: InvalidIds, DuplicateIds.

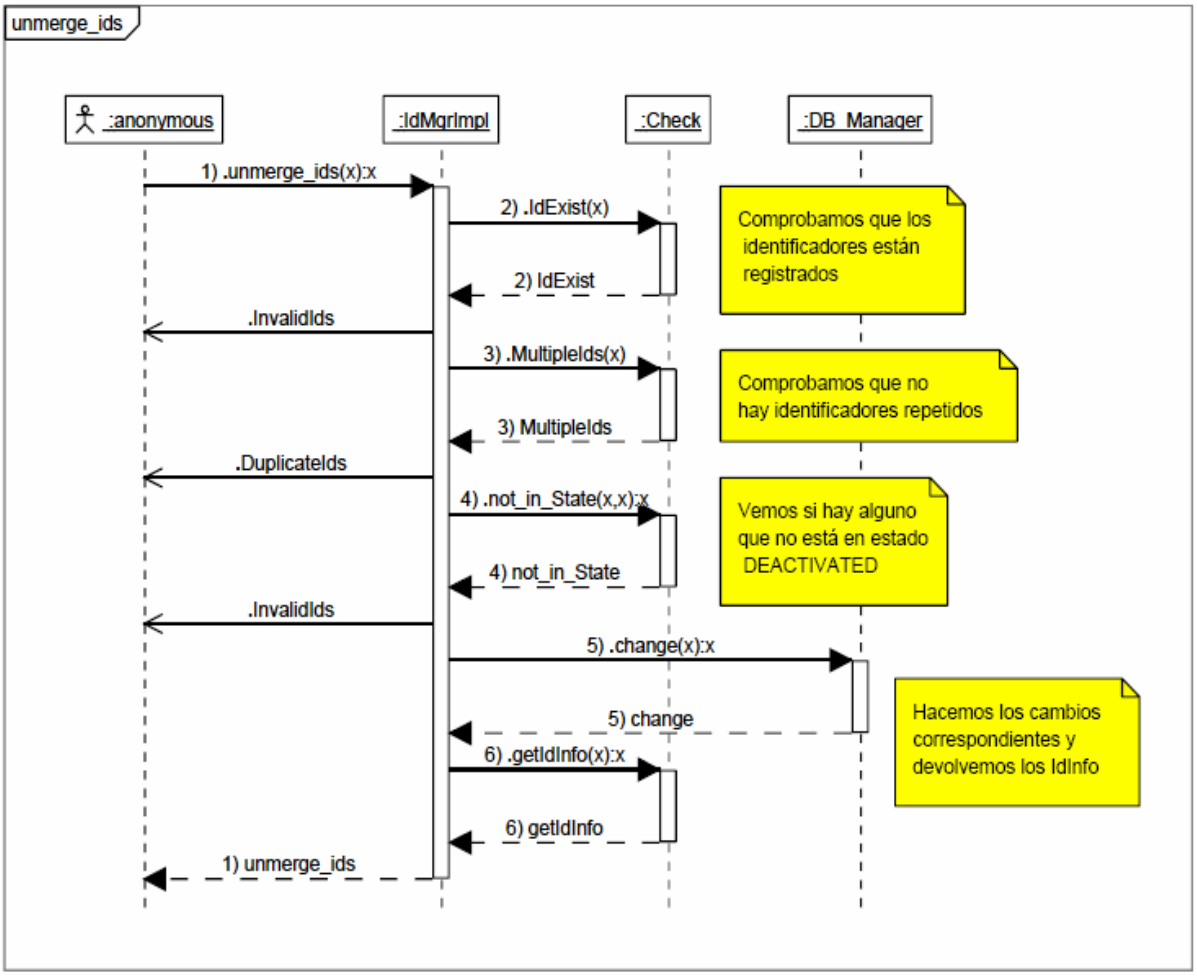


Figura 8.8 Diagrama de secuencia de unmerge_ids

deprecate_ids(PersonIdSeq ids_to_deprecate)
Desactiva un Identificador (estado Deactivated), cuando no se va a usar más.
Parámetros: secuencia de PersonIds a desactivar.
Devuelve: Secuencia de IdInfo de los PersonIds que se han desactivado.
Excepciones: InvalidIds, DuplicateIds.

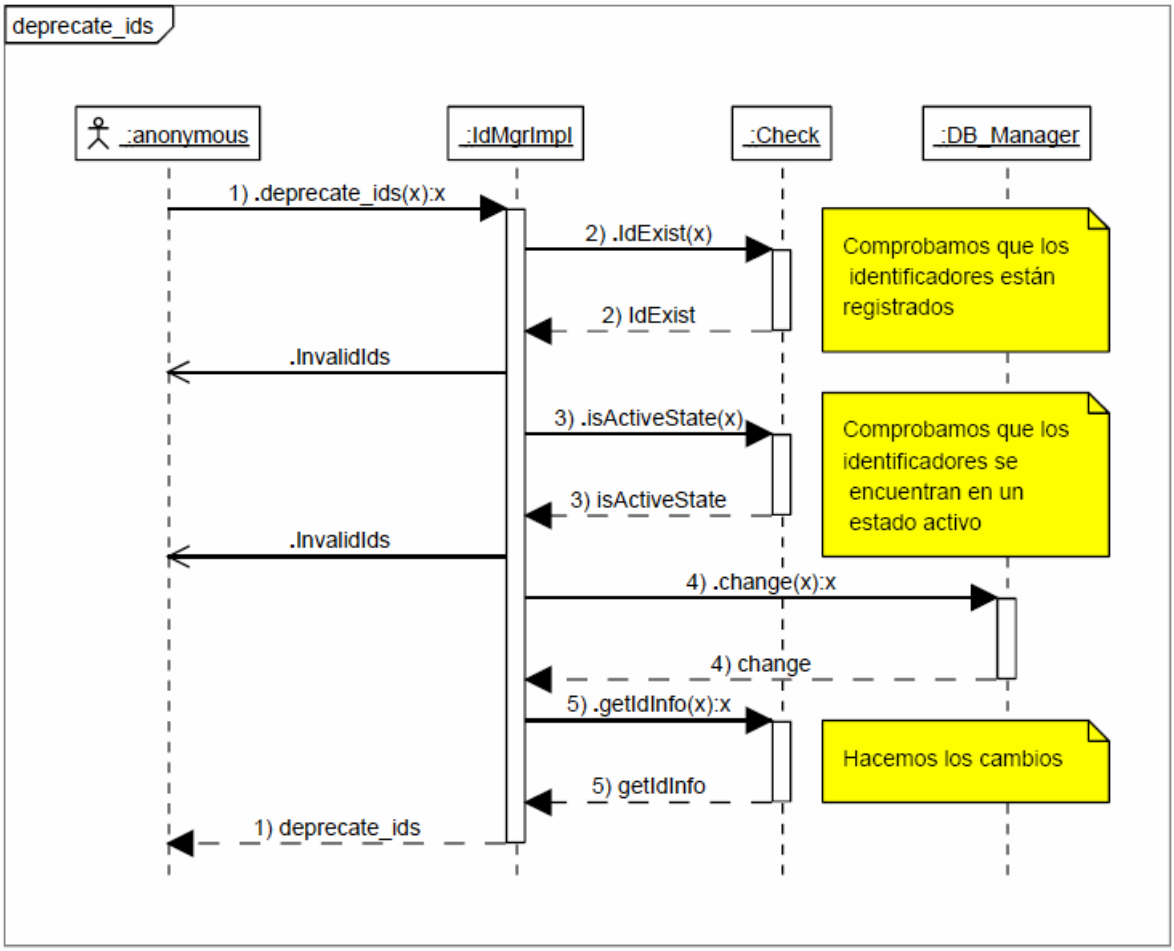


Figura 8.9 Diagrama de secuencia de deprecate_ids

Describimos ahora las funciones auxiliares y de comprobación utilizadas.

boolean MultipleProfiles(ProfileSeq profiles)

Comprueba que en la secuencia de perfiles que se pasa no hay ninguno repetido.
Parámetros: secuencia de Perfiles a comprobar.
Devuelve: true si no hay ninguno repetido.
Excepciones: DuplicateProfiles.

boolean MultipleTraits(ProfileSeq profiles)

Comprueba que en la secuencia de perfiles que se pasa no hay ninguno que contenga Trais repetidos.
Parámetros: secuencia de Perfiles a comprobar.
Devuelve: true si no hay traits repetidos.
Excepciones: MultipleTraits.

boolean MultipleIds (PersonIdSeq ids)

Comprueba que en la secuencia de identificadores que se pasa no hay ninguno repetido.
Parámetros: secuencia de identificadores a comprobar.
Devuelve: true si no se repiten identificadores.
Excepciones: DuplicateIds.

boolean different_ids (MergeStructSeq ids)

Comprueba que cada MergeStruct de la secuencia no contiene 2 identificadores iguales y que no hay MergeStructs repetidas en la secuencia.
Parámetros: secuencia de MergeStruct a comprobar.
Devuelve: true si los identificadores son diferentes.
Excepciones: DuplicateIds.

IndexSeq ProfileExists(ProfileSeq profiles,float peso)

Comprueba que los perfiles que se pasan no existen ya en la Base de Datos y si contienen un identificador administrativo que ya ha sido registrado.
Parámetros: secuencia de Perfiles a comprobar, y el factor de confianza de la búsqueda.
Devuelve: Los índices de los perfiles que están registrados.

boolean IdExist(MergeStructSeq ids)

Comprueba que los identificadores que forman parte de cada MergeStruct que se pasa existe en la BBDD.
Parámetros: secuencia de MergeStruct a comprobar.
Devuelve: true si existe en la Base de Datos.
Excepción: InvalidIds.

boolean IdExist(PersonIdSeq ids)

Comprueba que los identificadores que forman parte de cada PersonId que se pasa existe en la BBDD.
Parámetros: secuencia de identificadores a comprobar.
Devuelve: true si existe en la Base de Datos.
Excepciones: InvalidIds.

IdStateSeq ask_state(MergeStructSeq ids,String desiredIds)

Devuelve el estado en que se encuentran los PersonId,o los preferredIds que forman parte de cada MergeStruct que se pasa.
Parámetros: secuencia de MergeStruct de la que se quiere comprobar el estado y los identificadores que se quieren comprobar. (PersonId o preferredId)
Devuelve: Secuencia con el estado de los identificadores.

IdStateSeq ask_state(PersonIdSeq ids)

Devuelve el estado en que se encuentran los PersonId que se pasan.
Parámetros: secuencia de identificadores a comprobar.
Devuelve: Secuencia con el estado de los identificadores.

boolean isActiveState(PersonIdSeq ids)

Comprueba si los identificadores que se pasan estén un estado Activo (Temporary o Permanent) si no lo están se lanza la excepción correspondiente.
Parámetros: secuencia de identificadores a comprobar.
Devuelve: true si es un estado activo.
Excepciones: InvalidIds.

boolean isActiveState(MergeStructSeq ids)

Comprueba si los identificadores que forman el MergeStructSeq que se pasan están en un estado Activo (Temporary o Permanent).
Parámetros: secuencia de MergeStruct a comprobar.
Devuelve: true si es un estado activo.
Excepciones: InvalidIds.

boolean PreferredId(PersonIdSeq ids)

Comprueba si los identificadores de la secuencia tienen el campo preferred_id distinto a "" sino lanza InvalidId. Esto es así porque el PIDS especifica este valor para el preferredId de los PersonId que no están asociados a ningún otro.
Parámetros: secuencia de identificadores a comprobar.
Devuelve: true si es distinto.
Excepciones: InvalidIds.

boolean administrativeIds(ProfileSeq profiles)

Comprueba si en los perfiles que se pasan hay identificadores administrativos que ya están registrados. Y que tiene rellenos los campos extension y root.
Parámetros: secuencia de Perfiles a comprobar.
Devuelve: true si no están registrados.
Excepciones: InvalidAdministrativeIds.

boolean Mandatory(PersonIdSeq ids)

Comprueba que para cada identificador que se pasa, la información indicada como obligatoria esta presente en la BBDD.
Parámetros: secuencia de identificadores a comprobar.
Devuelve: true si está registrada toda la necesaria
Excepciones: MultipleTraits.

PersonIdSeq ask_preferredId(PersonIdSeq ids)

Genera un PersonIdSeq con los PreferredIds de los PersonId que se pasan como parámetro.
Parámetros: secuencia de identificadores.
Devuelve: secuencia con los preferredIds correspondientes a los pasados como parámetro.

IdInfoSeq getIdInfo(PersonIdSeq ids)

Genera la IdInfo de la secuencia de identificadores que se le pasa.
Parámetros: secuencia de identificadores.
Devuelve: secuencia de IdInfo.

PersonIdSeq not_in_State(PersonIdSeq ids_to_check,String state)

Comprueba que la secuencia de identificadores que se pasa no se encuentra en el estado indicado.
Parámetros: secuencia de identificadores a comprobar y el estado por el que se pregunta.
Devuelve: secuencia de identificadores que no se encuentran en el estado que se pasa como parámetro.

IndexSeq getErrorIndexSeq(Vector indicesParam)

A partir de una tabla de Index, genera un IndexSeq con los índices correspondientes a cada error error detectado.
Parámetros: Vector con que contiene los índices.
Devuelve: secuencia de índices correspondiente.

PersonIdSeq getErrorPersonIdSeq(Vector indicesParam)

A partir de una tabla de PersonIds, genera un PersonIdSeq con los PersonId correspondientes a cada error encontrado.
Parámetros: Vector con que contiene los índices.
Devuelve: secuencia de identificadores.

String[] getMandatoryTraits()

Devuelve nombres de los Traits que son obligatorios leyéndolos de la tabla auxiliar InfoPID.
Devuelve: tabla con los nombres correspondientes.

boolean traitNames(ProfileSeq profiles)

Comprueba que los traits que se pasan tienen un nombre válido.
Parámetros: secuencia de Perfiles a comprobar.
Devuelve: true si los nombres son correctos.
Excepciones: MultipleTraits.

8.4 Interfaz IdGest

A continuación se muestran las funciones que han sido definidas para completar la interfaz de gestión de identificadores.

II registroId (String extension,OID root, String preferredId, IVL_TS IVL)

Registra un identificador en la Base de Datos dados su extension y el OID de la entidad asignadora.
Parámetros: extensión del identificador, Oid de la entidad asignadora, identificador al que está asociado, estado y periodo de validez.
Devuelve: Instance Identifier con los datos del identificador.
Excepciones: InvalidId, InvalidAAN, InvalidIVL.

II registroId (String extension,String AAN, String preferredId,IdState estado, IVL_TS IVL)

Registra un identificador dadas su extensión y el nombre de la entidad asignadora.
Parámetros: extensión del identificador, nombre de la entidad asignadora, identificador al que está asociado, estado y periodo de validez.
Devuelve: Instance Identifier con los datos del identificador.
Excepciones: InvalidId, InvalidAAN, InvalidIVL, InvalidAAName, DuplicateAAN.

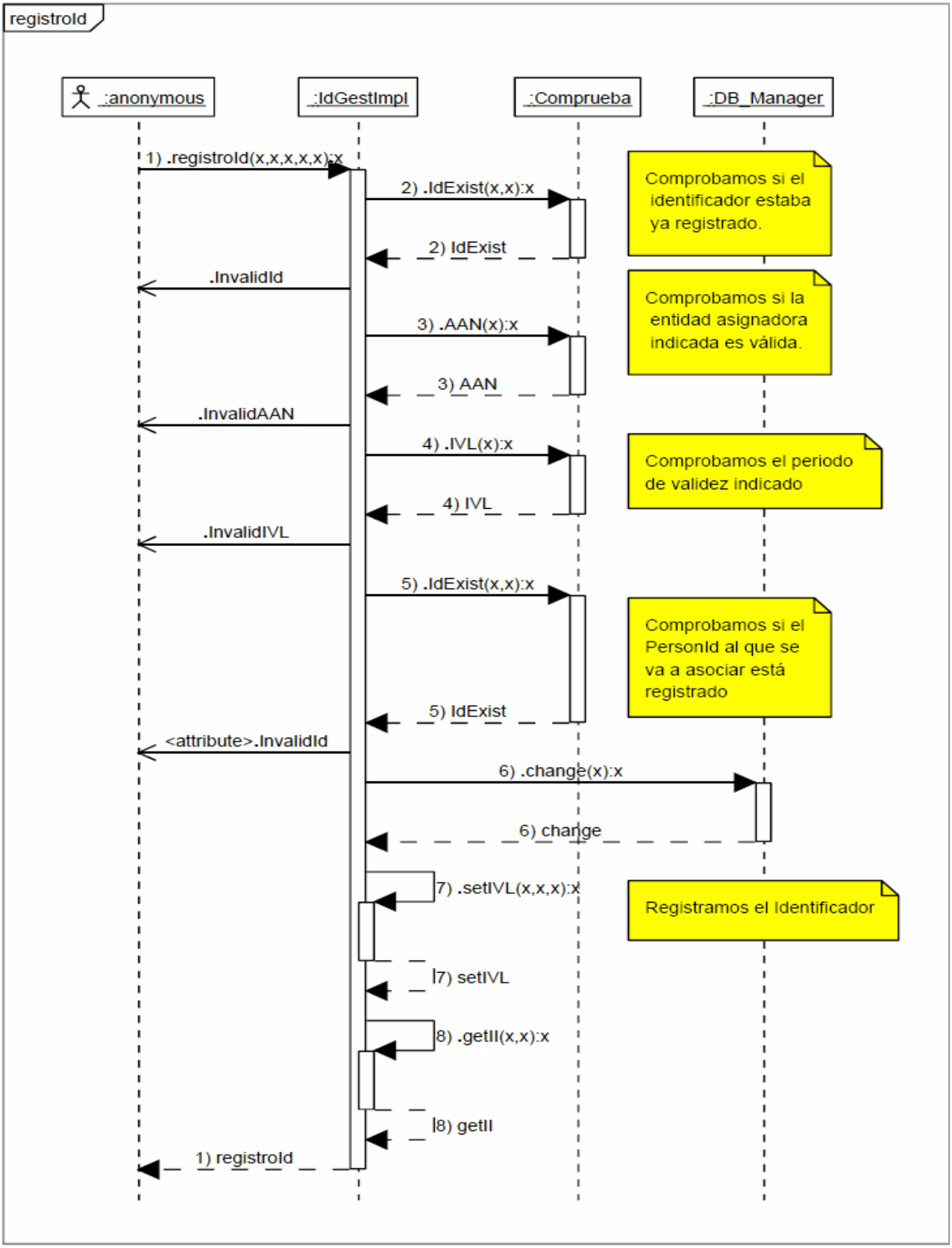


Figura 8.10 Diagrama de secuencia de registroId

IISeq getIdsbyAAN (OID root)

Devuelve una secuencia de Instance Identifiers que pertenecen a la entidad asignadora indicada con su OID.
Parámetros: OID de la entidad asignadora.
Devuelve: secuencia de InstanceIdentifier con los identificadores.
Excepciones: InvalidAAN.

IISeq getIdsbyAAN (String AAN)

Devuelve una secuencia de Instance Identifiers que pertenecen a la entidad asignadora indicada con su nombre como parámetro.
Parámetros: nombre de la entidad asignadora.
Devuelve: secuencia de InstanceIdentifier con los identificadores.
Excepciones: DuplicateAAN, InvalidAAName,InvalidAAN.

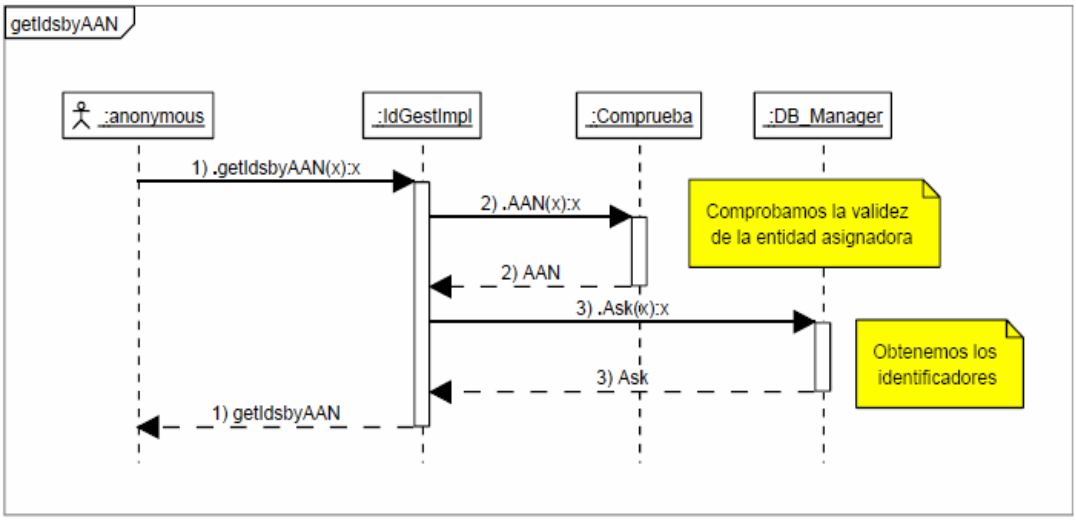


Figura 8.11 Diagrama de secuencia de getIdsbyAAN

IISeq getIdsbyIVL (IVL_TS IVL)

Devuelva una secuencia de identificadores cuyo periodo de validez se encuentra dentro de los límites del que se pasa como parámetro.
Parámetros: periodo de validez.
Devuelve: secuencia de II con los identificadores.
Excepciones: InvalidIVL.

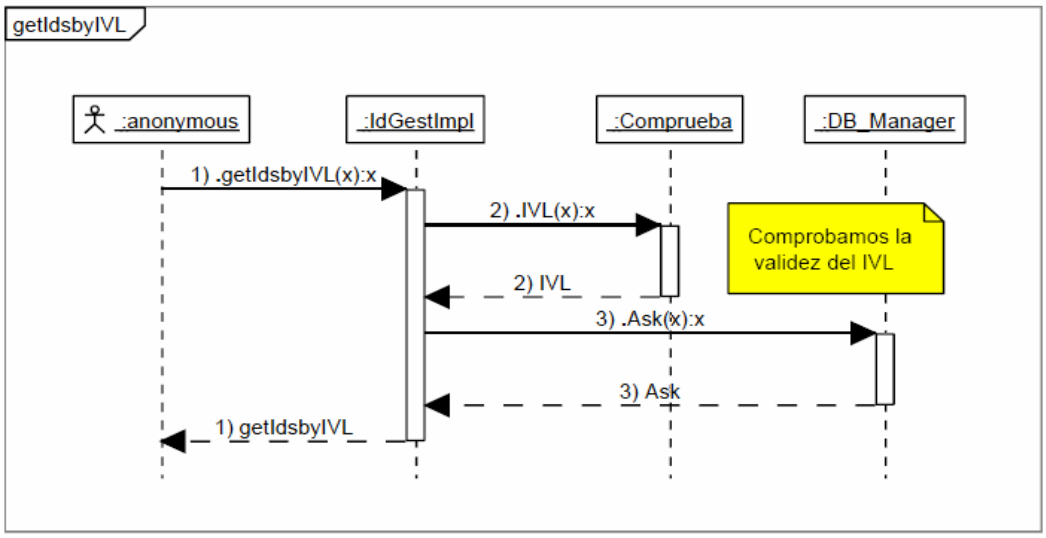


Figura 8.12 Diagrama de secuencia de getIdsbyIVL

IISeq getMergedIds (String extension, OID root)

Devuelve los identificadores asociados al que se pasa como parámetro.
Parámetros: extension y Oid del identificador.
Devuelve: InstanceIdentifier con los identificadores asociados.
Excepciones: InvalidId.

IISeq getMergedIds (String extension, String AAN)

Devuelve los identificadores asociados al que se pasa como parámetro.
Parámetros: extension y OID del identificador.
Devuelve: InstanceIdentifier con los identificadores asociados.
Excepciones: InvalidId, InvalidAAName.

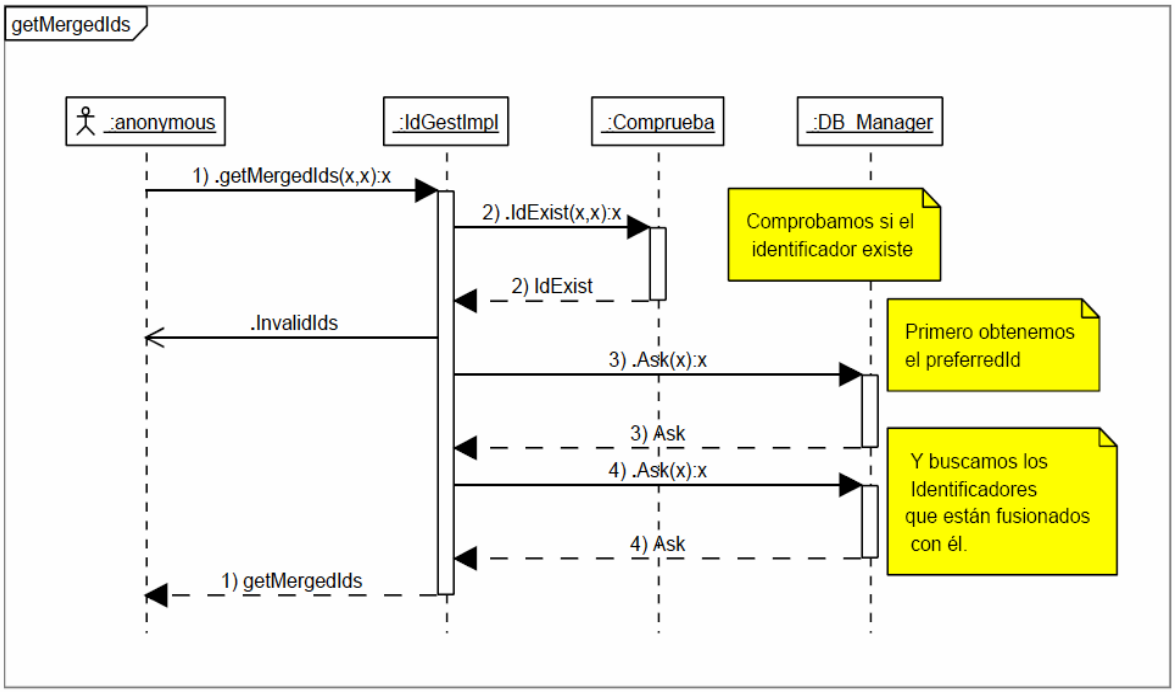


Figura 8.13 Diagrama de secuencia de getMergedIds

IISeq getMergedIdsbyAAN (String extension, OID root, OID AANroot)

Devuelve los identificadores asociados al que se pasa como parámetro y que además pertenecen a la entidad asignadora que se indica con su OID.
Parámetros: extensión y OID del identificador y OID de la entidad asignadora.
Devuelve: InstanceIdentifier con los identificadores asociados.
Excepciones: InvalidAAN, InvalidId

IISeq getMergedIdsbyAAN (String extension, OID root,String AAN)

Devuelve los identificadores asociados al que se pasa como parámetro y que además pertenecen a la entidad asignadora que se indica con su nombre.
Parámetros: extensión y OID del identificador y nombre se la entidad asignadora.
Devuelve: InstanceIdentifier con los identificadores asociados.
Excepciones: InvalidAAN,DuplicateAAN,InvalidAAName, Invalidad.

IISeq getMergedIdsbyAAN (String extension, String AAN, OID AANroot)

Devuelve los identificadores asociados al que se pasa como parámetro y que además pertenecen a la entidad asignadora que se indica.
Parámetros: extensión y nombre de la entidad asignadora del identificador y OID de la entidad asignadora por la que se pregunta.
Devuelve: InstanceIdentifier con los identificadores asociados.
Excepciones: InvalidAAN,DuplicateAAN,InvalidAAName,InvalidId.

IISeq getMergedIdsbyAAN (String extension, String AAN, String AANDes)

Devuelve los identificadores asociados al que se pasa como parámetro y que además pertenecen a la entidad asignadora que se indica.
Parámetros: extensión y nombre de la entidad asignadora del identificador y nombre de la entidad asignadora porla que se pregunta.
Devuelve: InstanceIdentifier con los identificadores asociados.
Excepciones: throws InvalidAAN, DuplicateAAN, InvalidAAName, InvalidId.

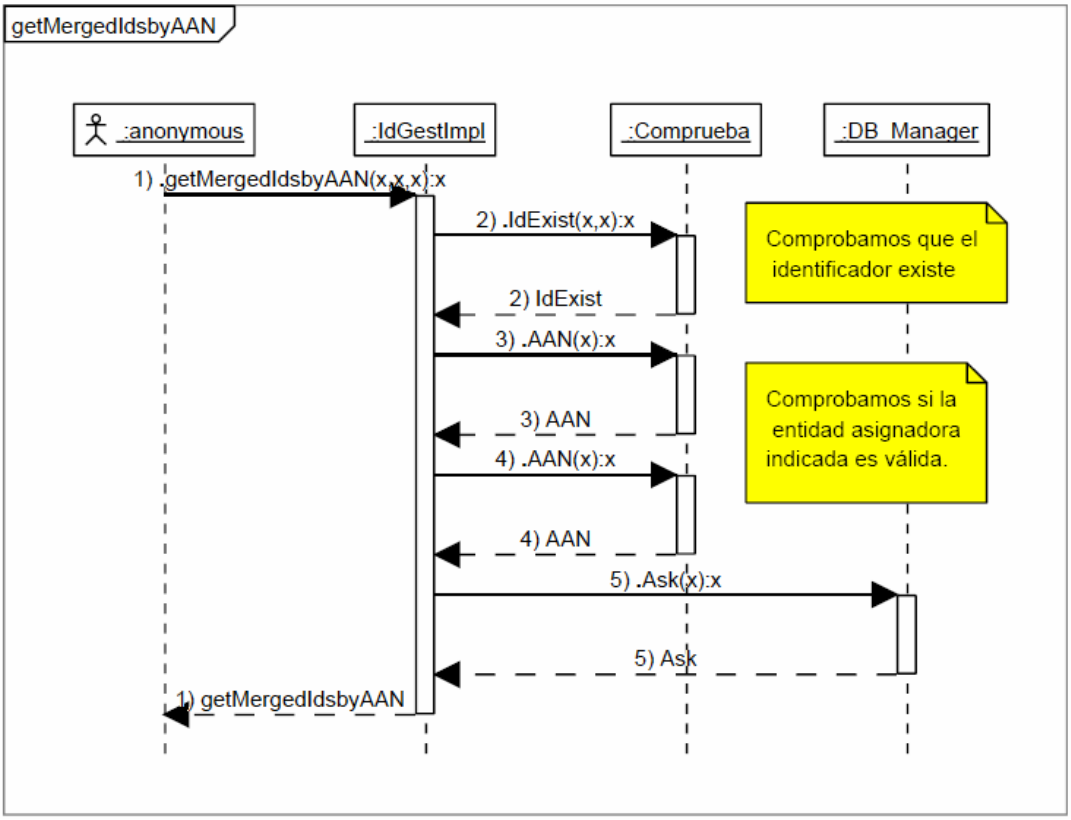


Figura 8.14 Diagrama de secuencia de getMergedIdsbyAAN

II mergeIds(String extension, OID root,String preferredId)

Fusiona el identificador que se pasa con su extensión y root con el que se pasa con su extensión y que es un identificador local.
Parámetros: extensión y OID del identificador a fusionar y extensión del identificador principal.
Devuelve: II que se ha fusionado.
Excepciones: InvalidAAN,InvalidId

II mergeIds(String extension,String AAN,String preferredId)

Fusiona el identificador que se pasa con su extensión y nombre de entidad asignadora con el que se pasa con su extensión y que es un identificador local.
Parámetros: extensión y nombre de la entidad asignadora del identificador a fusionar y extensión del identificador principal.
Devuelve: II que se ha fusionado.
Excepciones: InvalidAAN,DuplicateAAN,InvalidAAName,InvalidId.

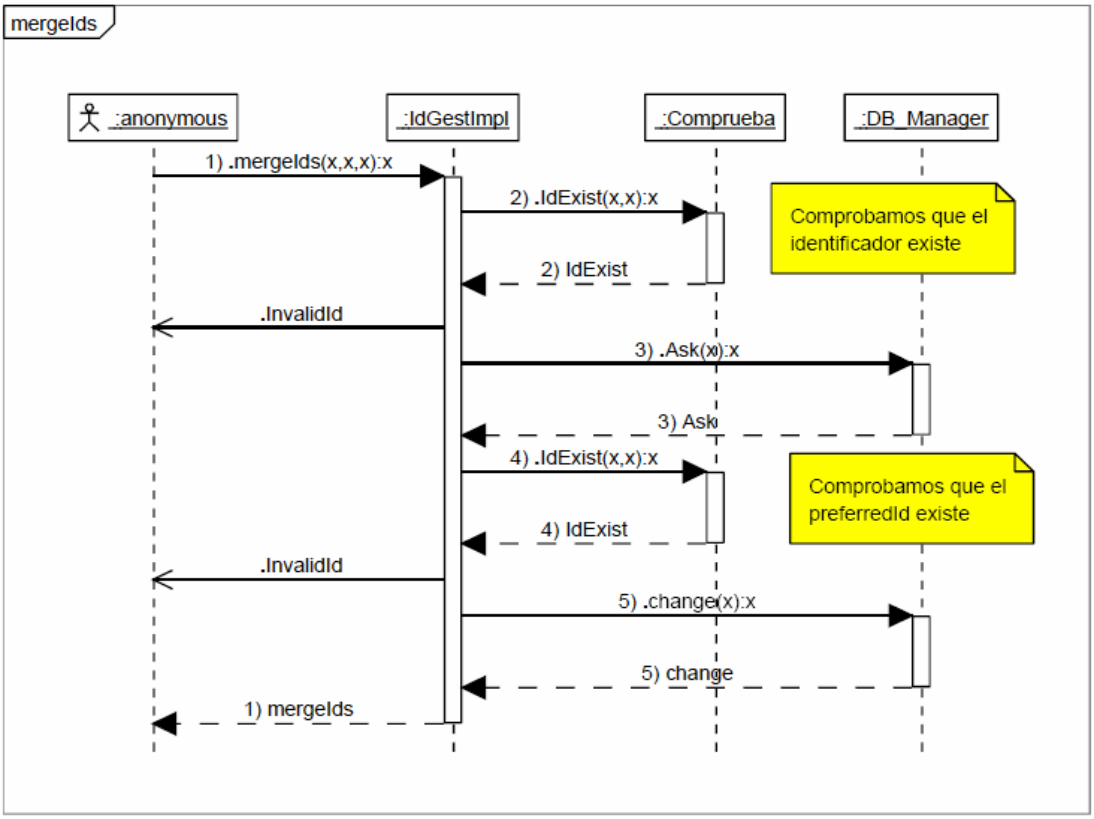


Figura 8.15 Diagrama de secuencia de mergeIds

II setIVL (IVL_TS IVL,String extension, OID root)

Establece el periodo de validez del identificador que se pasa.
Parámetros: extensión y OID del identificador a modificar y periodo de validez.
Devuelve: II que se ha modificado.
Excepciones: InvalidAAN,InvalidId.

II setIVL (IVL_TS IVL,String extension, String AAN)

Establece el periodo de validez del identificador que se pasa ahora con su extension y nombre de la entidad asignadora.
Parámetros: extensión y nombre de la entidad asignadora del identificador a modificar y periodo de validez.
Devuelve: II que se ha modificado.
Excepciones: InvalidAAN,DuplicateAAN,InvalidAAName,InvalidId.

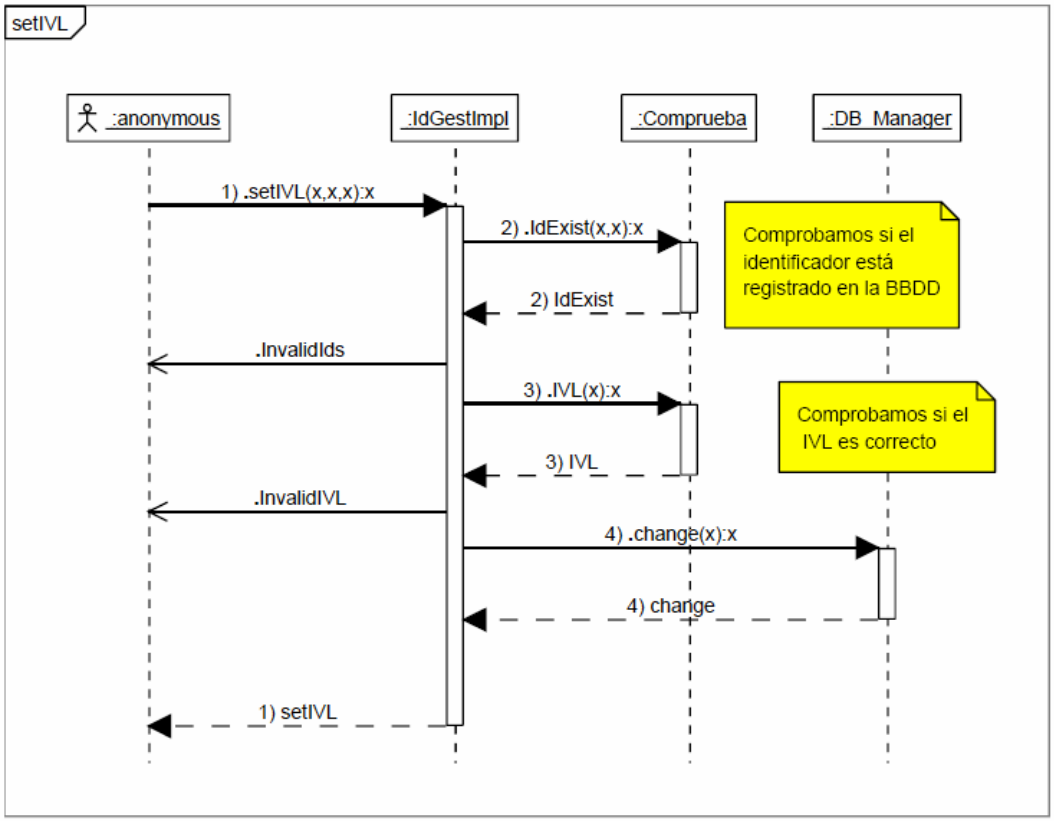


Figura 8.16 Diagrama de secuencia de setIVL

II getII(String extension,OID root)

Devuelve el Instance Identifier correspondiente al identificador que se pasa como parámetro.
Parámetros: extensión y OID del identificador.
Devuelve: II correspondiente.
Excepciones: InvalidId.

DV_IDENTIFIER getOEhrId(String extension,OID root)

Devuelve el DV_IDENTIFIER correspondiente al identificador que se pasa como parámetro.
Parámetros: extensión y OID del identificador.
Devuelve: DV_IDENTIFIER equivalente.
Excepciones: IdExist.

QualifiedPersonId getPIDId(String extension,OID root)

Devuelve el QualifiedPersonId correspondiente al identificador que se pasa como parámetro.
Parámetros: extensión y OID del identificador.
Devuelve: QualifiedPersonId equivalente.
Excepciones: IdExist.

Funciones de comprobación de la interfaz IdGest.

boolean IdExist(String extension,OID root)
Comprueba si el identificador que se pasa como parámetro está registrado en la base de datos.
Parámetros: extension y OID del identificador.
Devuelve: true si está en la BBDD y false si no.
Excepción: InvalidAAN.

boolean IdExist(String extension, String AAN)

Comprueba si el identificador que se pasa como parámetro, como extensión y nombre de la entidad asignadora, está registrado en la base de datos.
Parámetros: extensión y entidad asignadora del identificador.
Devuelve: true si está en la BBDD y false si no.
Excepciones: InvalidAAName,DuplicateAAN,InvalidAAN.

boolean AAN (OID root)

Comprueba si la entidad asignadora está registrada con su OID.
Parámetros: OID de la entidad asignadora.
Devuelve: true si está en la BBDD y false si no.
Excepción: InvalidAAN.

OID AAN (String AAN)

Devuelve el OID correspondiente a la entidad asignadora indicada con su nombre.
Parámetros: nombre de la entidad asignadora.
Devuelve: OID correspondiente.
Excepciones: DuplicateAAN,InvalidAAName.

boolean IVL(IVL_TS IVL)

Comprueba si el campo low del IVL que se pasa tiene fecha anterior al campo high.
Parámetros: IVL a comprobar.
Devuelve: false si no es cierto.
Excepciones: InvalidIVL.

Por último mostramos el diagrama de clases de la interfaz.

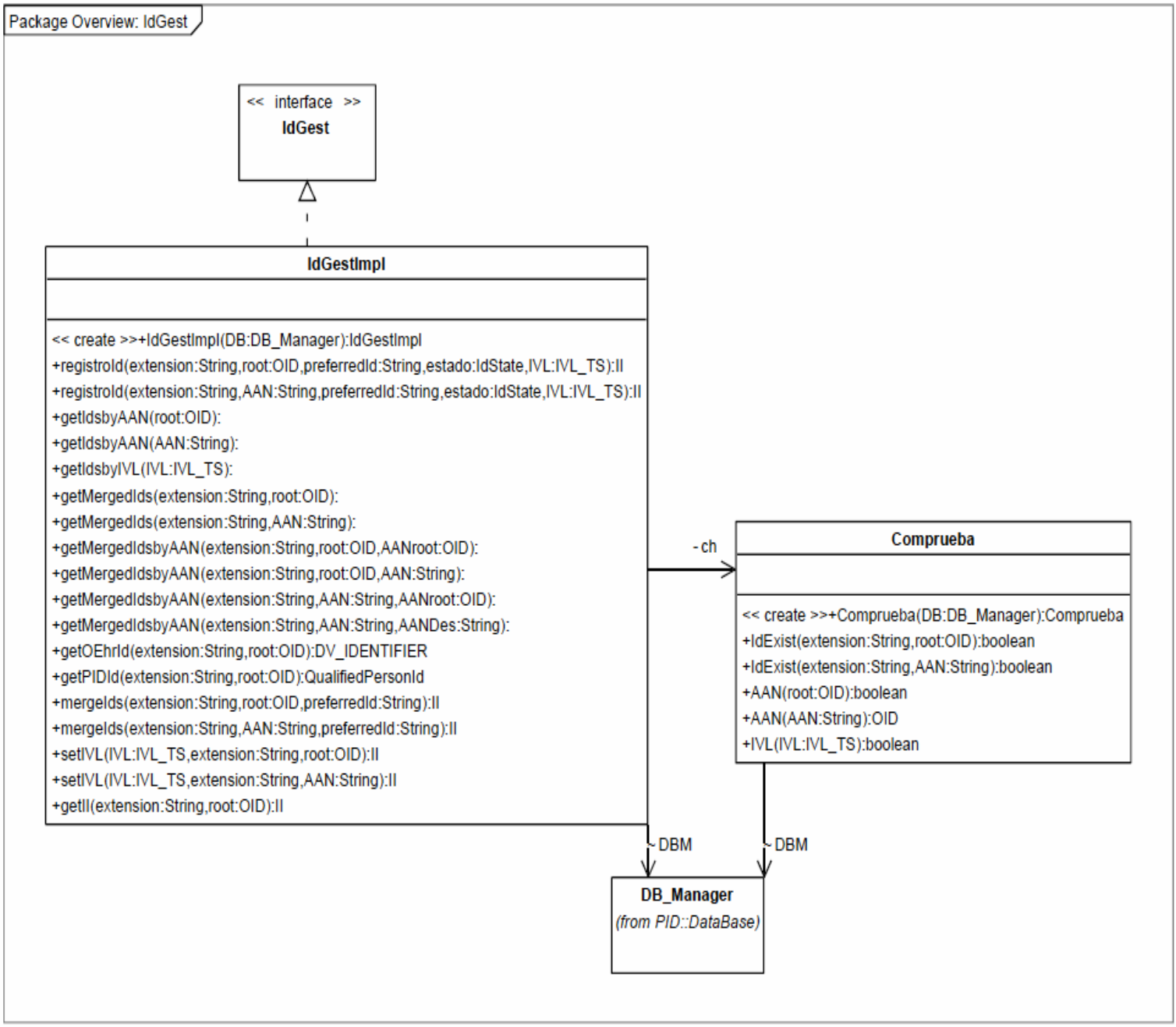


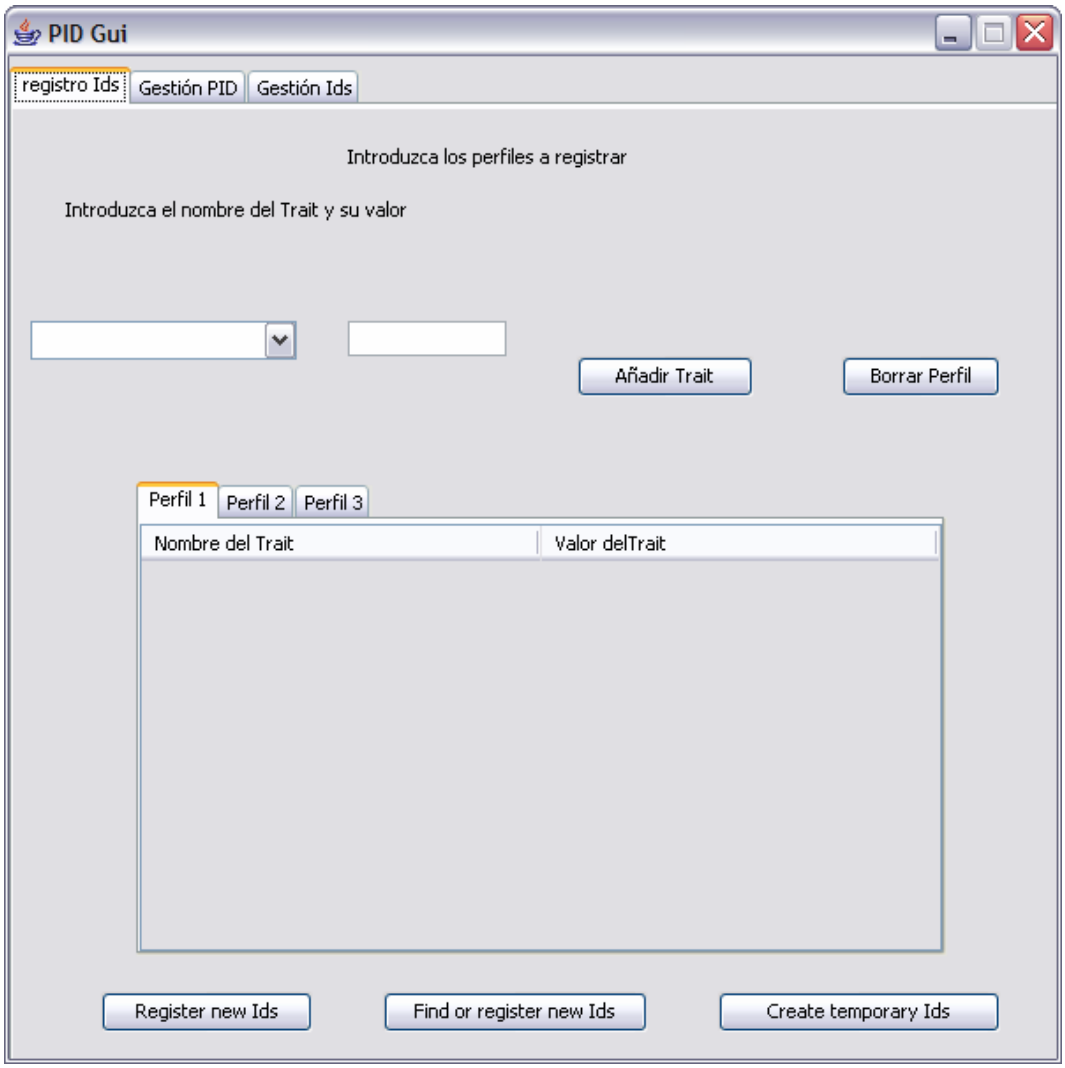
Figura 8.17 Diagrama de clases de IdGest.

8.5 Interfaz gráfica

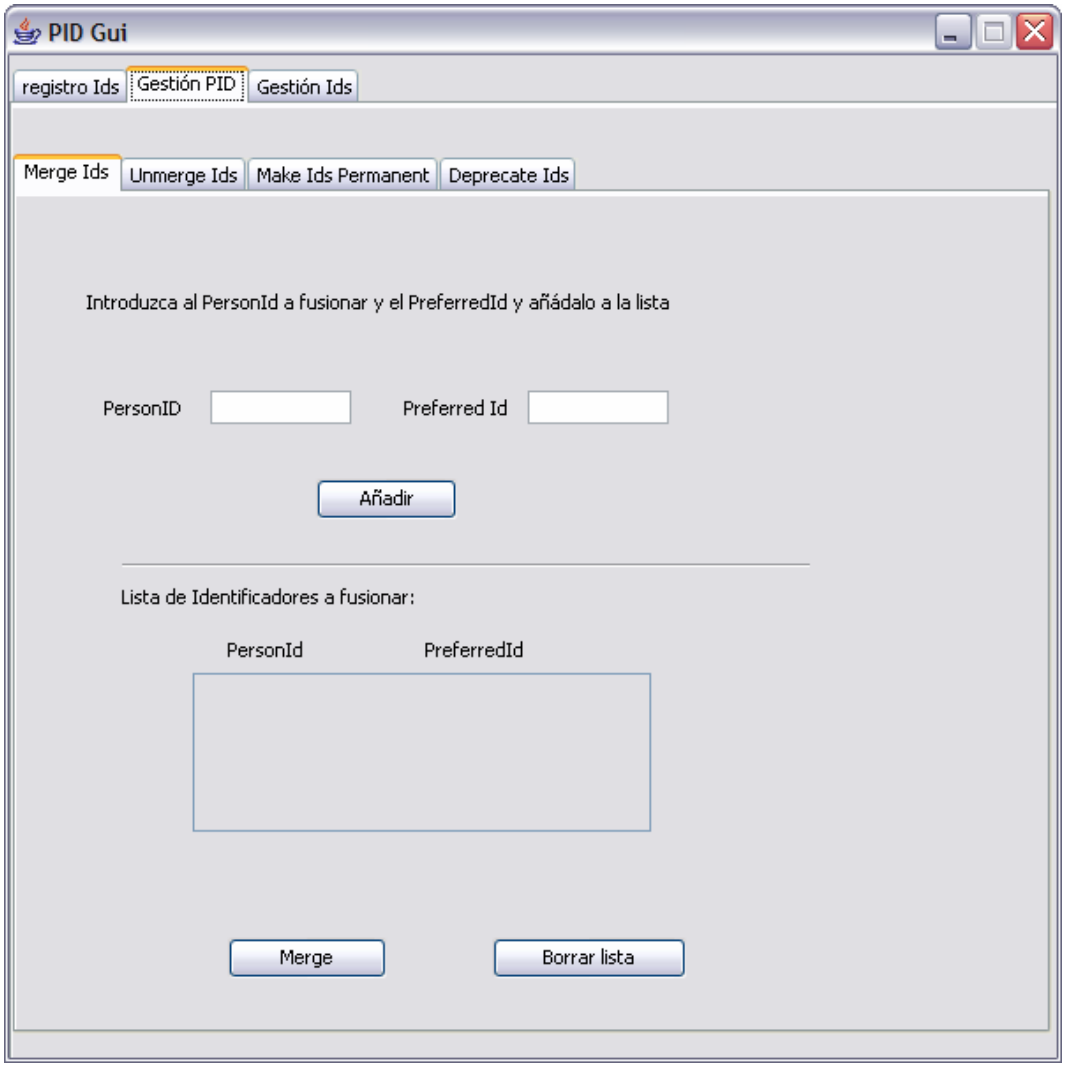
Para facilitar la comprobación del correcto funcionamiento de las funciones implementadas se ha diseñado una interfaz gráfica.

De esta forma el acceso a las funciones es muy simple, ya que la interfaz se encuentra dividida en tres partes principales, que a su vez se subdividen en otras.

Por una parte tenemos el registro de identificadores según el PIDS, esto da acceso a las funciones register_new_ids, find_or_register_new_ids y register_temporary_ids. El funcionamiento es bastante sencillo, se presentan tres posibles perfiles que se completan con los traits correspondientes, tras lo cuál se decide que función se quiere utilizar.

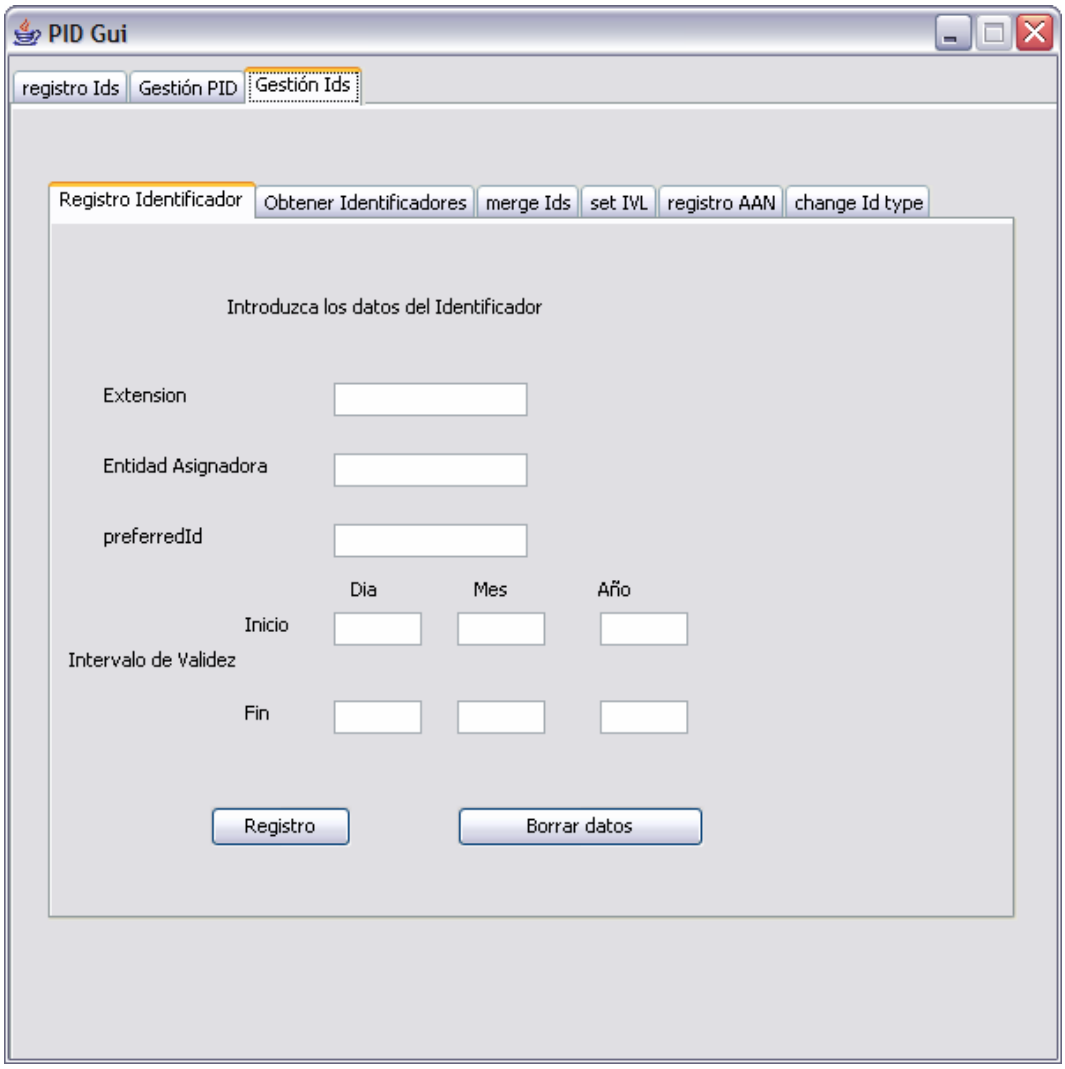


Por otra parte tenemos las funciones de gestión del PIDS, desde aquí se pueden introducir secuencias de identificadores sobre los que aplicar la operaciones correspondientes.



Y por último tenemos la parte destinada a los identificadores administrativos.

En la siguiente figura vemos como se divide en una parte para el registro de los identificadores, otra para acceder a ellos, para funciones de gestión, como fusionar identificadores o cambiar su periodo de validez y finalmente para registrar nuevas entidades asignadoras o cambiar el formato de los identificadores según el estándar deseado.


PID Gui
registro Ids
Gestión PID
Gestión Ids
Registro Identificador
Obtener Identificadores
merge Ids
set IVL
registro AAN
change Id type
Introduzca los datos del Identificador
Extension
Entidad Asignadora
preferredId
Dia
Mes
Año
Inicio
Intervalo de Validez
Fin
Registro
Borrar datos