Capítulo 5

SERVICIO DE IDENTIFICACIÓN DE PERSONAS (PIDS)

5.1 Necesidad del servicio de identificación de personas

A lo largo de su vida una persona recibe servicios médicos por parte de diferentes centros sanitarios, siéndole asignando en cada uno de estos un identificador de forma totalmente independiente.

Esta forma de asignar los identificadores es totalmente funcional para la organización que los asigna, pues permite tener unívocamente identificados a todos los pacientes, y por tanto el acceso a la información local sobre dicho paciente se puede realizar sin ningún problema.

Los inconvenientes surgen cuando se desea recopilar información acerca de un paciente que está almacenada en diferentes entidades, pues en cada una de ellas el mismo paciente tendrá un identificador distinto, válido en el centro donde se generó pero sin sentido en los demás.

Los sistemas de información sanitaria permiten la búsqueda de los datos de un paciente a partir de algunas características, por lo que cada vez que se necesiten dichos datos que estén en un sistema externo habría que realizar una búsqueda.

Este problema adquiere cada vez más importancia, ya que la creciente especialización de los servicios médicos conlleva la atención puntual en centros diferentes por lo que en cada uno de ellos se almacena parte de la historia clínica del paciente teniendo que acceder a todos ellos para obtener la historia completa.

Para realizar estas búsquedas, la información útil para identificar a una persona, se compone de aquellos atributos cuyo valor permanece constante o cambia muy lentamente a lo largo del tiempo, ya que puede almacenarse y usarse posteriormente con fines identificativos.

Por tanto la información que es utiliza para identificar a una persona, puede ser de distinta naturaleza, como:

- Datos demográficos (dirección, lugar de nacimiento, etc).
- Datos administrativos (numero de la seguridad social, numero de la licencia de conducción, numero del DNI, etc).

5.2 Objetivos del servicio

El servicio de identificación de personas de CORBAmed (PIDS) [1], maneja identificadores que cumplen las necesidades exigidas en el ámbito sanitario. El servicio ha sido diseñado para:

- Soportar de manera simultánea la asignación de IDs dentro de un dominio particular y la correlación de IDs entre múltiples dominios.
- Soportar la búsqueda y localización de personas, tanto en modo interactivo como en no atendido, con independencia del algoritmo de la máquina.
- Permitir que la implementación del PIDS proteja la confidencialidad de las personas, bajo la amplia variedad de políticas de privacidad y mecanismos de seguridad.
- Definir los niveles apropiados de conformidad para varios grados de sofisticación, partiendo de pequeñas búsquedas en un sólo dominio de ID hasta entre muchos dominios federados.

5.3 Modelo de referencia del PIDS

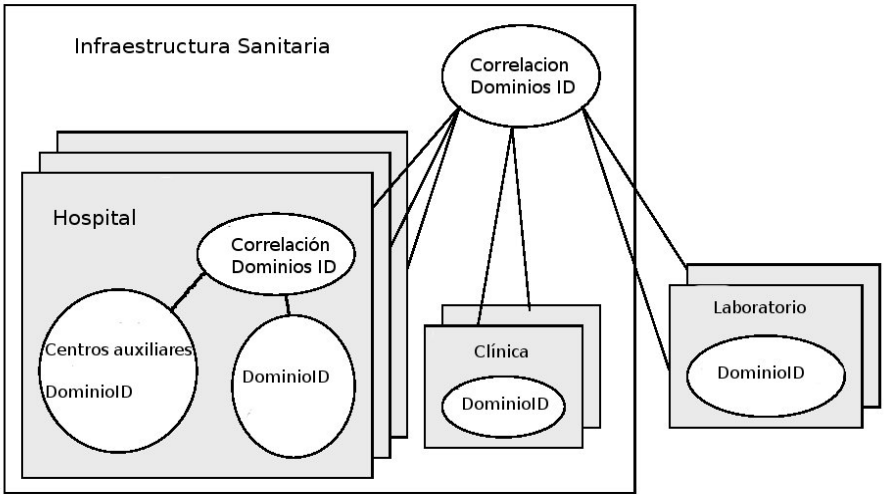

Figura 5.1 : Modelo de referencia del PIDS

Este es el modelo de referencia que proporciona el PIDS, en él se pueden apreciar los dominios de identificación que normalmente existen en una infraestructura sanitaria.

Un hospital utiliza identificadores pertenecientes a su propio dominio de identificación, trabajando a su vez con otros centros auxiliares pertenecientes al mismo hospital, como pueden ser laboratorios, los cuales tienen también su propio dominio.

El hospital es el encargado de establecer correspondencias entre ambos dominios de manera que se conviertan en uno solo, esto se hace correlando los identificadores para conocer cuál es el que corresponde a un mismo paciente en diferentes dominios.

De hecho, una buena infraestructura sanitaria contará con múltiples hospitales, laboratorios, clínicas... Cada uno con su propio dominio de identificación, por lo que una correcta gestión de dominios de IDs es fundamental.

5.4 Modelo de identificación del PIDS

Elementos básicos estructurales del modelo de identificación.

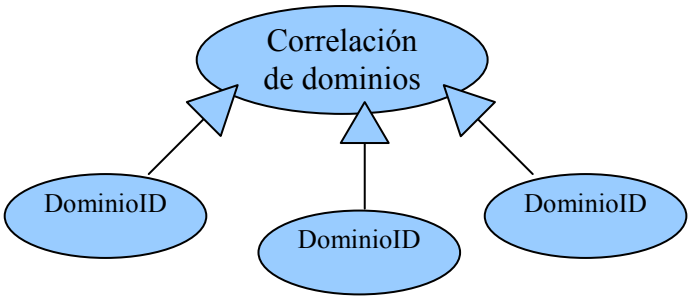

Figura 5.2 : Modelo de identificación del PIDS

En el PID el Dominio de identificación es el bloque básico, en cada dominio se mantiene un identificador único para cada persona, por tanto conociendo el dominio y el identificador podemos tener completamente identificada a una persona

Idealmente sólo debería haber un identificador por persona en cada dominio, pero puede ocurrir que esto no sea así por diferentes motivos, por lo que habrá que tener mecanismos que controlen estas situaciones.

En la especificación del PIDS se definen varias interfaces, destacando entre ellas las destinadas a identificar a una persona a partir de algunos de sus rasgo, la interfaz IdentifyPerson, y la que permite obtener los datos registrados de una persona a partir de su identificador, la ProfileAccess.

Además para mantener la integridad de cada dominio existen interfaces de gestión como la IdMgr.

5.5 Diagrama de herencia

El PIDS está formado por varias interfaces tal y como se muestra en la siguiente figura. Cada interfaz se encarga de una función dentro del PIDS y su implementación es opcional, por lo que se pueden implementar solo las interfaces necesarias.

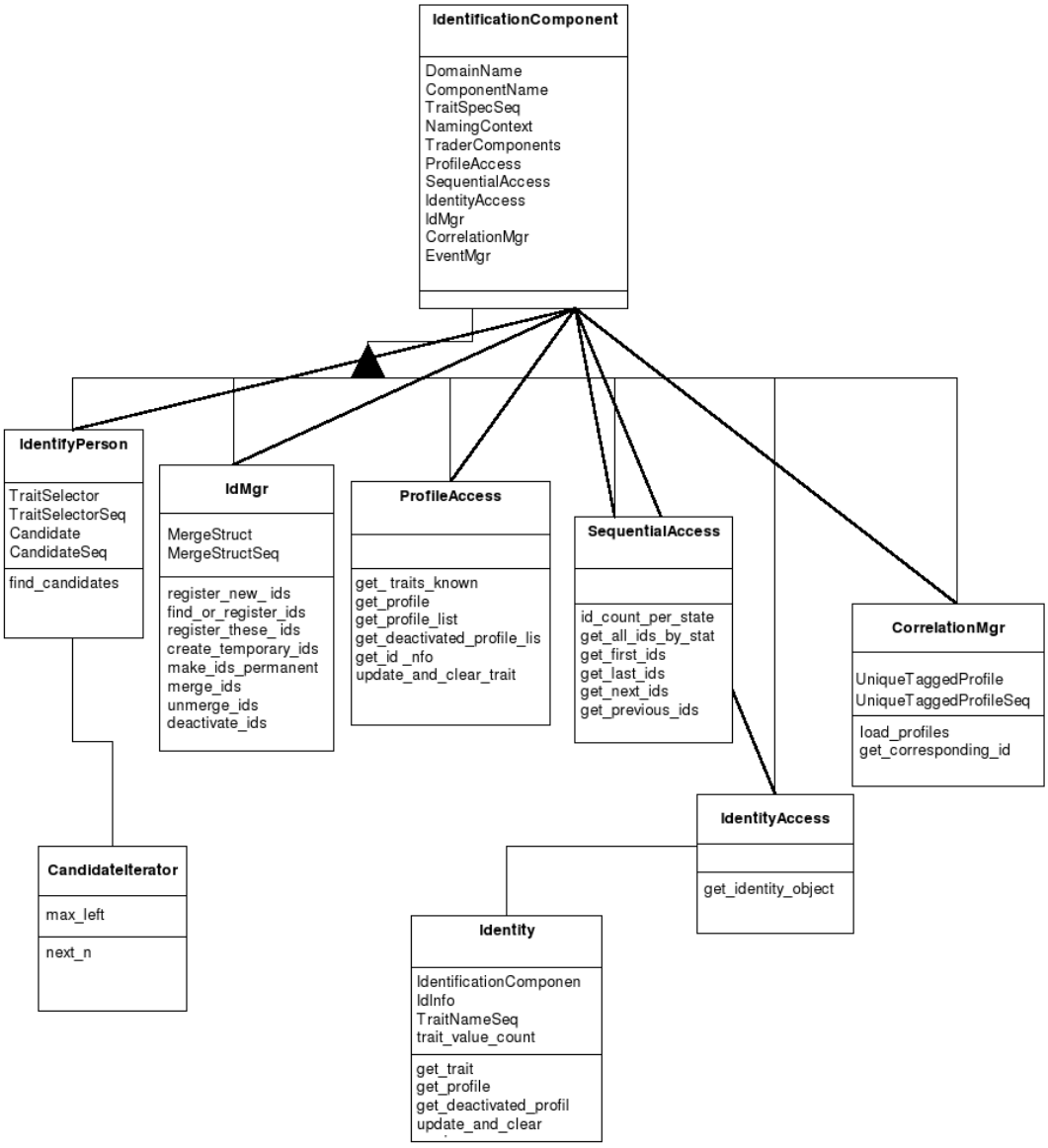

Figura 5.3: Diagrama de Herencia del PIDS

En este proyecto se parte de la implementación de las interfaces IdentifyPerson y ProfileAccess (métodos get_traits_known, get_profile y get_profile_list), y se va a implementar la interfaz IdMgr, encargada de gestionar los identificadores en cada dominio.

5.6 Interfaz IdMgr

La interfaz IdMgr proporciona funciones para la gestión de identificadores en un Dominio de Identificación. Todas las operaciones que realiza esta interfaz son de escritura, por lo que se debe tener un control de acceso para evitar cambios no deseados.

Este control de acceso es gestionado por el servicio de seguridad de CORBA y la implementación del servicio IdMgr.

5.6.1 Estado de los identificadores

Una característica muy importante de los identificadores es su estado, representado por la propiedad IdState, cada estado tiene un significado diferente y afecta a las operaciones que se puedan llevar a cabo sobre el identificador en cuestión.

Vemos los estados posibles:

- *UNKNOWN* : Este estado indica que el servicio no conoce si el identificador existe o no y, por tanto, tampoco cual puede ser el estado actual. Se usa por componentes que no administran un dominio ID pero que sí residen en un dominio donde sólo pueden conocer un pequeño subgrupo de los identificadores que existen.

- *INVALID*: Un identificador en este estado puede ser empleado únicamente en operaciones de creación de un identificador. Si un componente de PIDS conoce todos los identificadores en su dominio, cualquier otro identificador tiene el estado Invalid.

- *TEMPORARY*: Un identificador puede ser creado sin registrar ninguna característica de tipo mandatory. Un uso común es crear un identificador de forma que los datos puedan estar ligados a un paciente antes que el paciente se identifique de forma apropiada. Un identificador de este tipo puede hacerse permanente (permanent), fusionado (merged) o desactivado (deprecated). Un identificador temporal se transforma a permanente sólo si se hace de forma explícita, la sola actualización del perfil de la persona para que contenga los rasgos obligatorios no es suficiente para cambiar de estado.

- *PERMANENT*: Cuando un identificador se crea como Permanent, todas las características de tipo obligatorio deben estar reflejadas. Un identificador en el estado Permanent, puede pasar a ser fusionado (merged) o desactivado (deprecated), pero nunca pasar a temporal (temporary).

- *DEACTIVATED*: Una vez que un identificador no se piensa que vuelva a ser usado más, puede ser desactivado (merged o deprecated), manteniéndose únicamente para propósitos de auditoria. Este es el estado final de un identificador, y no puede pasar a otro estado salvo con la operación de unmerge.

Una vez visto los estados posibles, mostramos un diagrama con las transiciones entre estados y las operaciones que intervienen en cada una de ellas:

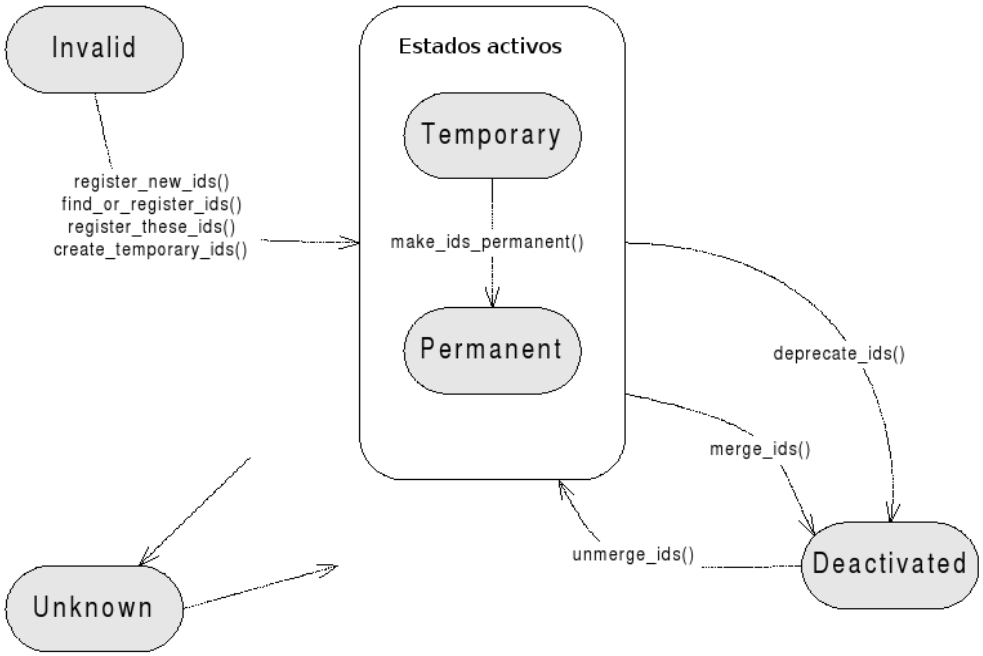

Figura 5.4: Transiciones entre estados

5.6.2 Componentes de la interfaz

- register_new_ids :

```
PersonIdSeq register_new_ids(
      in ProfileSeq profiles_to_register )
      raises (
            ProfilesExist,
            DuplicateProfiles,
            MultipleTraits );
```

Genera nuevos identificadores en el dominio y los asigna a los perfiles que se pasan. Es decir, establece una asociación entre identificadores y perfiles, haciendo a estos últimos accesibles para el resto de las operaciones.

La generación de identificadores se hace de acuerdo a las reglas y convenciones del dominio de identificación.

Con el uso de esta función el cliente está indicando que necesita el identificador de forma inmediata, y que se espera que lleguen a ser permanentes, aunque depende de la implementación del servicio si el identificador devuelto está en estado permanente o temporal a la espera de una posterior revisión.

- find_or_register_ids:

```
PersonIdSeq find_or_register_ids(
      in ProfileSeq profiles_to_register )
      raises (
               DuplicateProfiles,
               MultipleTraits );
```

Esta función es utilizada para generar identificadores de forma automática, de decir, sin atención humana. Una variable interna, indica el umbral de confianza para considerar que una persona ya está registrada en el dominio de identificación. Si la búsqueda de esa persona devuelve un valor de confianza por debajo de dicho umbral se genera un nuevo identificador y se le asigna el perfil.

Esta función solo cambia el estado de un identificador si crea uno nuevo (el cual pasará de estado Invalid a Permanent o Temporary), que podrá ser al igual que antes permanente o temporal dependiendo de las características de la implementación.

- *register_these_ids:*

```
void register_these_ids(
    in TaggedProfileSeq profiles_to_register )
    raises (
            NotImplemented,
            IdsExist,
            DuplicateIds,
            ProfilesExist,
            DuplicateProfiles,
            MultipleTraits );
```

Esta operación es similar a *register_new_ids* salvo que aquí se indica el valor de los identificadores que se van a generar. Puede ocurrir que la política de generación de identificadores de la implementación no permita esto, por lo que se lanzaría la excepción correspondiente. (NotImplemented)

- *create_temporary_ids:*

```
PersonIdSeq create_temporary_ids(
      in ProfileSeq profiles_to_register )
      raises (
            MultipleTraits );
```

Genera un nuevo identificador preferiblemente en estado temporal, aunque esto depende de la implementación. Esta operación crea un nuevo identificador aunque se corresponda a uno existente.

- *make_ids_permanent:*

```
PersonIdSeq make_ids_permanent(
        in PersonIdSeq ids_to_modify )
        raises (
          InvalidIds,
          DuplicateIds,
          RequiredTraits );
```

Los identificadores en estado temporal pueden pasar a estado permanente usando esta función. El identificador que se devuelve puede ser el mismo que se pasó pero con el estado actualizado, o se puede generar otro distinto, todo esto dependiendo de la política de la implementación.

- merge_ids:

```
IdInfoSeq merge_ids(
     in MergeStructSeq ids_to_merge )
     raises (
         InvalidIds,
         DuplicateIds );
```

Cuando una persona tiene mas de un identificador en el mismo dominio, todos excepto uno se pueden fusionar usando esta operación.
Los identificadores fusionados pasarán a estado desactivado, y el campo preferred_id de su IdInfo tendrá el valor del identificador con el que se ha fusionado.

- unmerge_ids:

```
IdInfoSeq unmerge_ids(
    in PersonIdSeq ids_to_unmerge )
      raises (
            InvalidIds,
            DuplicateIds );
```

Si un identificador que ha sido fusionado, resulta que no identificaba a una persona repetida, se puede deshacer la fusión con esta operación.

- deprecate_ids:

```
IdInfoSeq deprecate_ids(
    in PersonIdSeq ids_to_deprecate )
      raises (
            InvalidIds,
            DuplicateIds );
```

Cuando no se espera usar más un identificador, éste puede ser retirado del servicio cambiando su estado a desactivado. El perfil al que se correspondía normalmente se mantiene por motivos de auditoria.

5.6.3 *Estructura de datos utilizadas*

Vemos a continuación algunos tipos de datos definidos en el PIDS y que son usadas por las funciones antes descritas.

- *IdState*

```
enum IdState { UNKNOWN, INVALID,
TEMPORARY, PERMANENT, DEACTIVATED };
```

Representa el estado en que se encuentra un identificador.

- *MergeStruct:*

```
struct MergeStruct {
      PersonId id;
      PersonId preferred_id;
};
```

Estructura usada en la operación de merge_ids, contiene el identificador que se va a fusionar, y el identificador con el que va a ser fusionado.

- *IdInfo:*

```
struct IdInfo {
    PersonId id;
    IdState state;
    PersonId preferred_id;
};
```

Estructura que contiene información sobre un identificador, su estado y el preferred_id si está fusionado con algún otro.