

# SGME - It007 - Análisis y diseño del mini-sistema

Andrés Moschini

## Historial de revisiones

Revisión Entrega012.01 2007-12-01

Creación del documento.

Revisión Entrega012.01 2007-12-03

Última revisión, este documento es solo un borrador interno.

Revisión Entrega013.01 2007-12-08

Se comienza a rehacer el documento mas profesionalmente.

Revisión Entrega013.02 2008-01-07

Se documentan los cambios a realizar en la librería de acceso a la base de datos y como debe reaccionar el sistema

## 1. Introducción

Este documento se basará en los casos de uso definidos para esta iteración en *It007-Casos\_Uso* [PDF]. Por lo tanto deberá definir el comportamiento del sistema ante las siguientes solicitudes del usuario:

- Aplicar informe completo
- Aplicar solo un suceso específico de un informe
- Aplicar solo una operación específica de un informe
- Deshacer un informe completo
- Deshacer solo un suceso específico de un informe
- Deshacer solo una operación específica de un informe

Para ello se adaptarán y extenderán las clases que representan los informes, sucesos y operaciones utilizadas para cargar los informes XML, también se crearán nuevos comandos en la API de la base de datos y comandos de la librería de acceso a los datos.

## 2. Comportamiento de las posibles interfaces gráficas

El mini-sistema de esta iteración deberá dar soporte a una interfase que permita lo siguiente:

- Crear un nuevo informe con sus atributos.
- Modificar los atributos del informe actual.
- Agregar, editar o quitar sucesos al informe
- Agregar, editar o quitar operaciones a los sucesos del informe.
- Permitir realizar todas estas operaciones en forma local, es decir sin grabar los datos físicamente.
- Permitir aplicar los cambios parcialmente y continuar editando el informe.
- Permitir cancelar o desaplicar total o parcialmente los datos aplicados del informe actual.

Para facilitar la implementación y dar consistencia y previsibilidad a los cambios realizados, se permitirá limitar a la interfase de la siguientes formas:

- Si una operación está aplicada, que no se permita editarla o eliminarla (debería "des-aplicarse" antes).
- Si otro usuario realizó cambios que dependen de alguna operación que se desee "desaplicar", no se permitirá "desaplicar" dicha operación.
- Si una operación define un marcador utilizado por otra, no podrá eliminarse, ni eliminarse el marcador.
- Cualquier consulta que se realice no tendrá en cuenta las operaciones no aplicadas.

- Si falla una operación de aplicar no se aplicará nada.
- Si falla una operación de "desaplicar" se desaplicará todo lo posible y se permitirá dejar sin "desaplicar" las operaciones relacionadas con los fallos.

En *It007-Simulacion\_Creacion\_Informe.pdf* puede observarse una simple prueba de escritorio de la posible interfase en funcionamiento.

### 3. Operaciones básicas

Para realizar las tareas indicadas anteriormente se dispondrá de las siguientes operaciones básicas (implementadas como clases heredadas de `PF.DbMm.ComandoExecute` o `PF.DbMm.ComandoDb`):

*CargarInforme* (`CargadorInforme`)

Crea el encabezado del informe en la base de datos (identificador, fecha de recibido y comentario)

*ModificarInforme* (`ModificadorInforme`)

Modifica en la base de datos el encabezado de un informe (fecha de recibido y comentario)

*EliminarInforme* (`EliminadorInforme`)

Elimina físicamente la entrada de un informe de la base de datos, solo funciona si el informe no tiene sucesos asociados.

*CargarSuceso* (`CargadorSuceso`)

Crea el encabezado del suceso en la base de datos (identificador, fecha de ocurrido y comentario)

*ModificarSuceso* (`ModificadorSuceso`)

Modifica en la base de datos el encabezado de un suceso (fecha de ocurrido y comentario)

*EliminarSuceso* (`EliminadorSuceso`)

Elimina físicamente la entrada de un suceso de la base de datos, solo funciona si el suceso no tiene operaciones asociadas.

*CargarOperacion* (`CargadorOperacion` y sus herederos: `CargadorOpAgregarComponentes`, `CargadorOpCambiarAtributoArticulos`, `CargadorOpIngresarArticulos` y `CargadorOpIngresarArticulos`)

Crean la operación correspondiente en la base de datos, realizando los cambios correspondientes sobre los objetos del dominio del sistema afectados.

La base de datos internamente (mediante triggers) verifica la validez de los cambios con respecto a los cambios introducidos anteriormente.

*EliminarOperacion* (`EliminadorOp`)

Elimina físicamente la operación de la base de datos deshaciendo los cambios correspondientes.

La base de datos internamente (mediante triggers) verifica la posibilidad de eliminar los cambios con respecto a los cambios introducidos posteriormente.

#### 3.1. Adaptación de la librería de acceso a la base de datos

##### 3.1.1. Creación de un informe

La creación de un informe ya está implementada por la clase `PF.DbMm.Usuario.Informes.CargadorInforme`, la cual cumple los requisitos.

### 3.1.2. Modificación del encabezado de un informe

Para realizar esta tarea, se deberá desarrollar una clase *PF.DbMm.Usuario.Informes.ModificadorInforme* similar a *PF.DbMm.Usuario.Informes.CargadorInforme* que se encargue de modificar sus datos.

### 3.1.3. Eliminación de un informe

Solo debería permitirse la eliminación de un informe si aún no ha sido terminada su creación, en caso contrario debería anularse de manera que quede un registro de su previa existencia.

En cualquier caso, solo se puede eliminar o anular un informe si antes fueron eliminados o anulados todos sus sucesos y las operaciones de estos.

La restricción acerca de permitir la eliminación será controlada a nivel interfase. En cuanto a la restricción de que antes deben eliminarse todos sus componentes, será verificada a nivel triggers en la base de datos.

Para realizar esta función se desarrollará la clase *PF.DbMm.Usuario.Informes.EliminadorInforme*.

### 3.1.4. Creación de un suceso

La creación de un suceso ya está implementada por la clase *PF.DbMm.Usuario.Informes.CargadorSuceso*. Al cargar un suceso de forma independiente del informe, se deberá verificar que el informe esté cargado

### 3.1.5. Modificación de un suceso

Para realizar esta tarea, se deberá desarrollar una clase *PF.DbMm.Usuario.Informes.ModificadorSuceso* similar a *PF.DbMm.Usuario.Informes.CargadorSuceso* que se encargue de modificar sus datos.

### 3.1.6. Eliminación de un suceso

Solo debería permitirse la eliminación de un suceso si aún no ha sido terminada la creación del informe que lo contiene, en caso contrario debería anularse de manera que quede un registro de su previa existencia.

En cualquier caso, solo se puede eliminar o anular un suceso si antes fueron eliminadas o anuladas todas sus operaciones.

La restricción acerca de permitir la eliminación será controlada a nivel interfase. En cuanto a la restricción de que antes deben eliminarse todos sus componentes, será verificada a nivel triggers en la base de datos.

Para realizar esta función se desarrollará la clase *PF.DbMm.Usuario.Informes.EliminadorSuceso*.

### 3.1.7. Objetos Aplicados

Dado que al aplicar, deben aplicarse todos los objetos correspondientes y en caso de fallo, debe deshacerse la transacción y no debe quedar ninguno aplicado, se intentará aplicar todas los objetos y recién al final, si no hubo ningún fallo, se informará a los objetos correspondientes que fueron aplicados.

### 3.1.8. Eliminar una operación

Debe crearse una clase *PF.DbMm.Usuario.Informes.EliminadorOp* que se encargue de eliminarla operación de la base de datos.

Se deberá verificar que los estados provocados por los cambios generados por la operación a eliminar no sean base de operaciones ejecutadas luego, en caso contrario no podrá realizarse la eliminación.

Dicha verificación se realizará a nivel base de datos mediante triggers.

## 4. Realización de las solicitudes del usuario

A continuación se definirá como debe responder el sistema a las posibles solicitudes del usuario.

### Aplicar informe completo

1. Si el informe aún no se ha creado en la base de datos se realizará *CargarInforme*, en caso contrario, si se modifico el encabezado con respecto a la última vez que se grabo se realizará *ModificarInforme*.
2. Por cada suceso aplicado del informe, si se modificó su encabezado, se realizará *ModificarSuceso*.
3. Por cada operación no aplicada de los sucesos aplicados se realizará *CargarOperacion*.
4. Por cada suceso no aplicado del informe, en orden según fecha de ocurrencia, se realizará *CargarSuceso*.
5. Por cada operación de los sucesos no aplicados se realizará *CargarOperacion*.
6. Se marcará el informe, sus sucesos y sus operaciones como aplicados.
7. Si falla alguna de las operaciones, se deshacen los cambios realizados en la base de datos y no se marca ningún elemento como aplicado.

### Deshacer informe completo

1. En orden contrario a la fecha de ocurrencia de los sucesos y primero las últimas en aplicar, se desharán las operaciones de los sucesos.  
Cada operación deshecha se quitará de la lista de operaciones aplicadas y se agregará al inicio de la lista de operaciones no aplicadas del suceso correspondiente.  
Si una operación no se puede deshacer, continuará en la lista de operaciones aplicadas y se continuará deshaciendo.
2. Cada suceso al que se le puedan deshacer todas las operaciones, se deshará también.
3. Si se logró deshacer todos los sucesos del informe, este también se deshará, en caso contrario continuará aplicado.

### Aplicar un suceso determinado

1. Si el suceso no está aplicado se realizará *CargarSuceso*, en caso contrario se realizará *ModificarSuceso*.
2. Por cada operación no aplicada se realizará *CargarOperacion*.
3. Se marcarán los objetos correspondientes (el suceso y sus operaciones) como aplicados.
4. Si falla alguna operación se desharán los cambios en la base de datos y no se marcará ningún objeto como aplicado.

### Deshacer un suceso determinado

1. Primero las últimas en aplicar, se desharán las operaciones del suceso.  
Cada operación deshecha se quitará de la lista de operaciones aplicadas y se agregará al inicio de la lista de operaciones no aplicadas del suceso correspondiente.  
Si una operación no se puede deshacer, continuará en la lista de operaciones aplicadas y se continuará deshaciendo.
2. Si fue posible deshacer todas las operaciones del suceso, este se deshará también, en caso contrario no

### Aplicar una operación determinada

1. Se realizará *CargarOperacion* para la operación seleccionada.
2. Si falla la carga de la operación se informa.
3. Si la carga es exitosa la operación se quita de la lista de operaciones no aplicadas del suceso y se agrega al inicio de la lista de operaciones aplicadas.

### Deshacer una operación determinada

1. Se realizará *DeshacerOperacion* para la operación.

- 
2. Si se puede deshacer con éxito la operación, esta se retira de la lista de operaciones aplicadas y se inserta al inicio de las no aplicadas del suceso.
3. Si no es posible deshacer la operación, se informa.