

## **3.2 Descripción general**

### **3.2.1 Perspectiva del proyecto**

Los clientes desean un producto que sea sencillo de parametrizar y para ello se dispone con este proyecto de una herramienta que la pueden utilizar personas sin ningún conocimiento de programación.

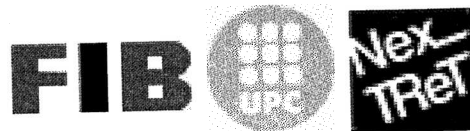
Existen otro tipo de herramientas, que incorporan un lenguaje de parametrización complicado que implica una formación específica. En el lado opuesto está ISM, donde si una persona sabe navegar y comprende los conceptos básicos de Internet puede ser capaz de grabar una transacción con ISM.

La perspectiva inicial, era la de solventar la monitorización de los sistemas de La Caixa, pero visto el potencial, la herramienta a desarrollar, debía poder testear diferentes aplicaciones web basadas en HTML.

### **3.2.2 Funciones del producto**

Los objetivos principales que espera resolver La Caixa y los diferentes clientes donde está funcionando ISM son:

- Eliminar los testeos manuales, que implican un coste elevado en personal.
- Eliminar los errores humanos en el testeo de aplicaciones web.
- Disponer de una herramienta capaz de avanzarse a la llamada de los clientes en caso de que una aplicación web o una operativa no funcionen correctamente.
- Un sistema capaz de reproducir con exactitud lo que un usuario humano realizaría
- Poder realizar operaciones desde diferentes puntos de la red o desde diferentes operadores de Internet.
- Sistema de alarmas, en caso de error, con aviso a los operadores de guardia.
- Disponer de una herramienta de Reporting, donde se puedan visualizar los datos obtenidos
- Disponer de tiempos de respuesta de cada una de las páginas cargadas
- Disponer de la disponibilidad de lo testado.

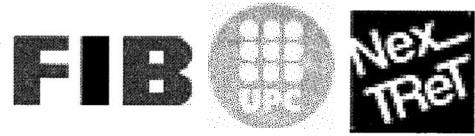


### **3.2.3 Características de los usuarios**

La aplicación dispone de un usuario administrador que se encarga de administrar los agentes y de grabar los diferentes tests.

Otro tipo de usuarios son los que consultan los datos obtenidos por la aplicación en las consolas de operación, estos usuarios no entran dentro del sistema pero son informados con datos que obtiene ISM.

Por último, existen una serie de usuarios que consultan los datos de tiempos de respuesta y disponibilidad en la aplicación de Reporting. Estos usuarios pueden ser desde operadores de guardia hasta directivos que desean saber como funciona la aplicación por la cual están invirtiendo dinero.



### **3.3 Requerimientos específicos**

#### **3.3.1 Requerimientos del interfaces**

Los requerimientos de interface que debe cumplir son los siguientes dependiendo del módulo a analizar:

##### **Consola y Agente**

- Deben utilizarse desde el sistema operativo Windows por lo tanto las ventanas de usuario deben coincidir con la apariencia normal de este sistema operativo
- La interficie para grabar y reproducir debe ser un navegador web, y este navegador debe ser el más utilizado por los usuarios. El navegador más utilizado es Internet Explorer.
- Debe ser una aplicación cliente servidor,
- La consola y el agente deben poder situarse en cualquier parte del mundo con acceso a Internet y con visibilidad al servidor que se encargará de recoger datos y sincronizar árboles de ejecución.
- Actualización en tiempo casi real, debe haber una demora en las actualizaciones de los árboles de ejecución de los agentes de unos minutos.

##### **Reporting**

- Sistema multiplataforma
- Acceso web

#### **3.3.1.1 Requerimientos del usuario**

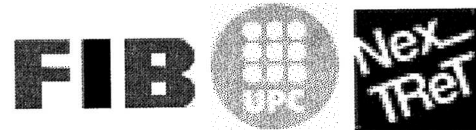
No existen unos requerimientos específicos de usuario para el entorno tratado en este PFC. Existen administradores que tienen acceso a la Consola de ISM y a los Agentes.

#### **3.3.1.2 Requerimientos de hardware**

Los requerimientos de hardware se pueden dividir dependiendo del elemento a analizar:

##### **Consola y Agente:**

- Pentium IV a 2,5, con al menos 512Mb de memoria RAM



- Conexión directa a Internet a por medio de una LAN
- Modem para conexiones de acceso telefónico a red.

### **Reporting**

- Servidor Pentium IV 2,5 GHz con 512 Mb de RAM, pero se aconseja un servidor puro para hospedar el Reporting.

### **Servidor de Base de Datos**

- Servidor Pentium IV 2,5 GHz con 2 Gb de RAM.

#### **3.3.1.3 Requerimientos de Software**

##### **Consola y Agente:**

- Windows 2000 Profesional con SP2, actualmente Windows XP Profesional en castellano.
- Internet Explorer 6.0 con SP2 en castellano

### **Reporting**

- Servidor de aplicaciones para J2EE, la aplicación está certificada para Tomcat 5.0 y Websphere.

### **Servidor de Base de Datos**

- SQL Server 2000 con SP2, aunque soporta Oracle y Informix.

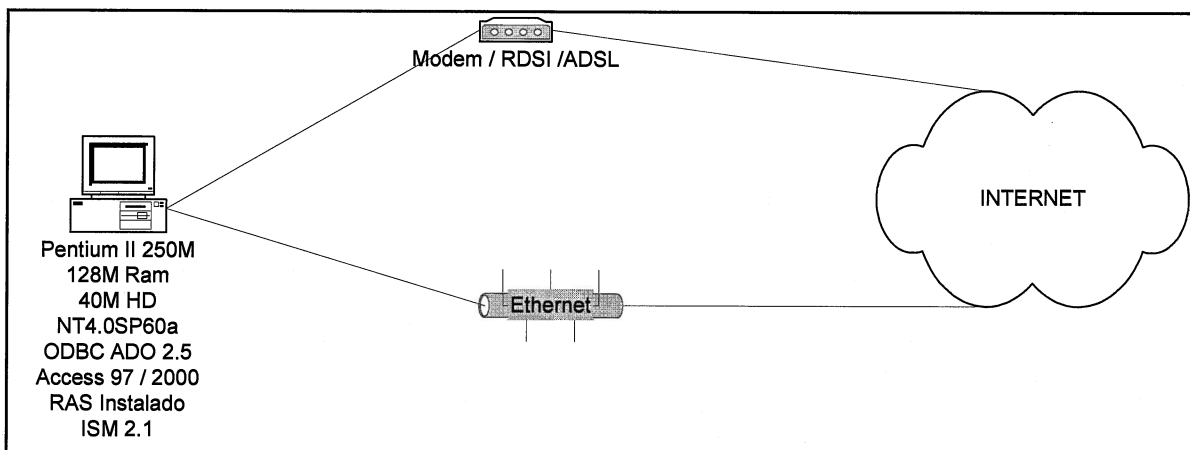
#### **3.3.1.4 Requerimientos de comunicaciones**

A continuación, se exponen las tres posibles topologías, o configuraciones del ISM. Las tres topologías son:

1. Stand alone.
2. Múltiples ISM En Red.

- 3. Con o sin parte Web
- 4. ISM en Red y con conexión módem.

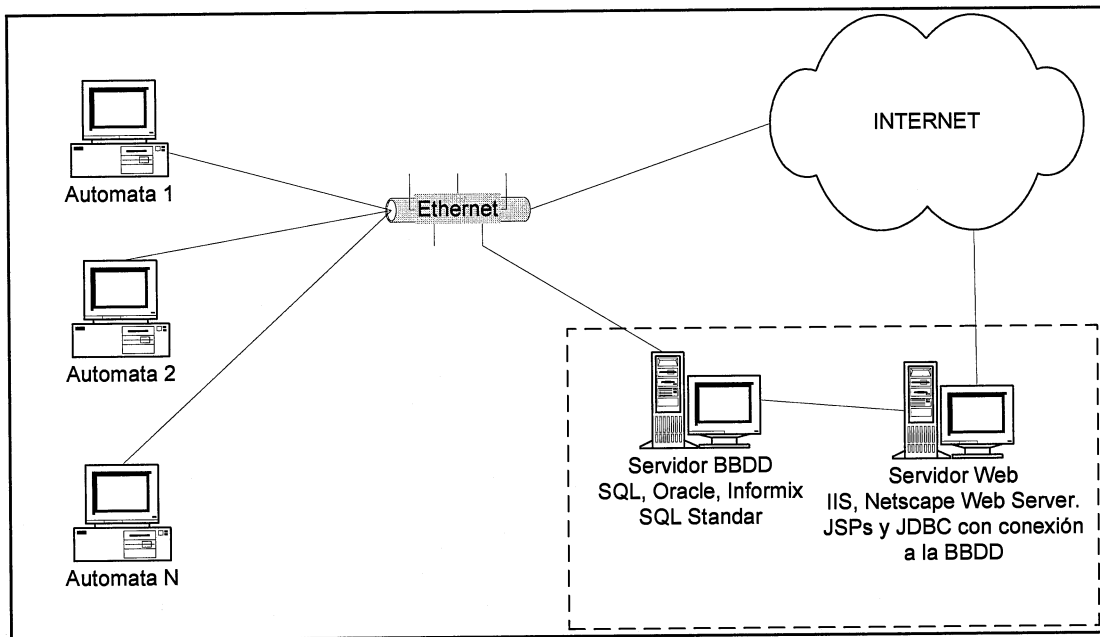
La primera topología consiste en instalar un ISM en modo Stand alone, o sea sin ningún tipo de conexión a base de datos externa. Esta topología tiene el inconveniente que la extracción de datos se realiza en local, y tan solo permite evaluar el producto, delante de la maquina.



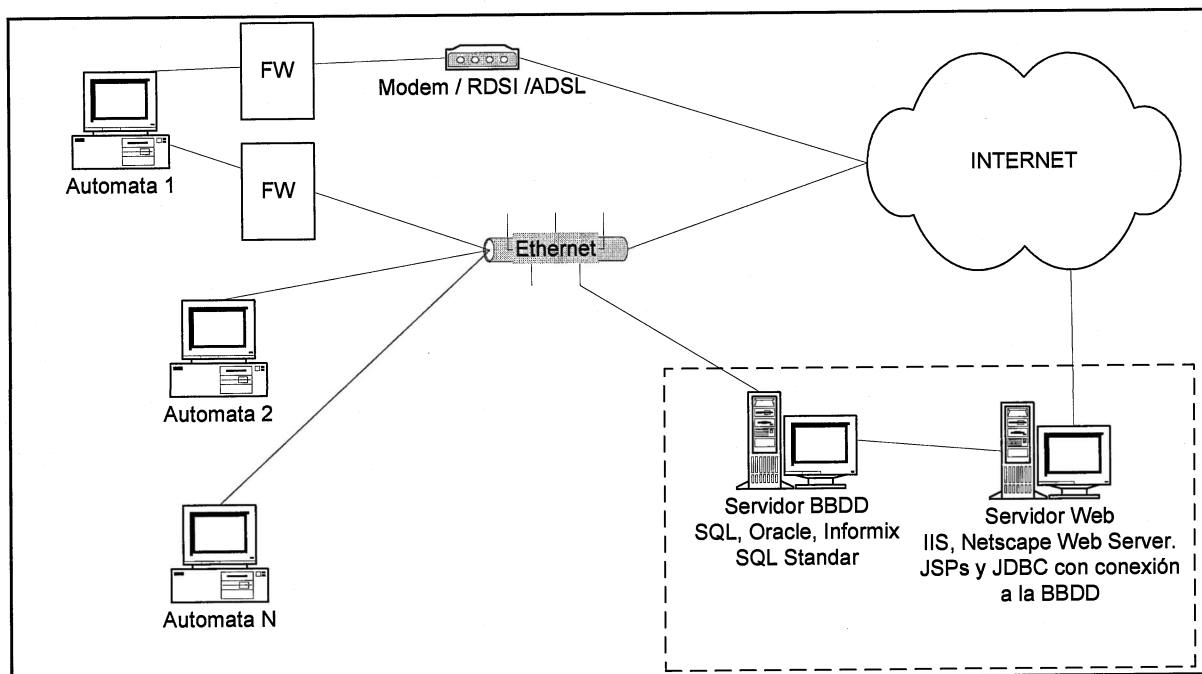
La segunda opción es montar una red, con diversos autómatas y una base de datos remota. Para la versión 2.5 del ISM, es necesario tener también un servidor Web, que sea capaz de ejecutar páginas JSP y tener acceso JDBC a la base de datos.

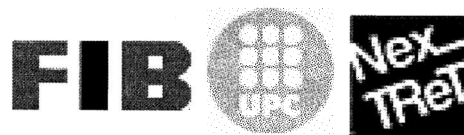
En este caso, dado que la conexión a Internet es interna no se producen problemas de seguridad por culpa del ISM.

La configuración quedaría tal como se muestra en el siguiente gráfico:



El último caso, es aquel en el cual tenemos una máquina con conexión a Internet a través de un módem. Este caso plantea serios problemas de seguridad. La mejor solución es aislar esta máquina con un FireWall, cosa que se puede hacer o mediante módem con FireWall incorporado (RDSI / ADSL) o poniendo un FireWall entre la maquina y el servidor que solo permita el acceso a la base de datos.





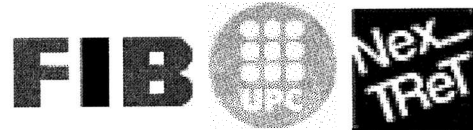
### **3.3.2 Requerimientos de rendimiento**

Las diferentes alarmas deben poder notificarse, como se puede ver anteriormente, a través de los siguientes métodos:

- OpenView.
- Ventana emergente en la consola.
- Reporting.
- SMS
- Correo electrónico

#### **Base de datos.**

Tan importante como el hecho de que se notifiquen las alarmas, es el hecho de que éstas se detecten con la mayor brevedad posible, por lo que el rendimiento debe ser acorde con este requerimiento, y acercarse, en la medida de lo posible, a una aplicación en tiempo real.



### **3.3.3 Requerimientos de diseño**

El diseño debe tener en cuenta que el sistema tiene las siguientes características:

- Sistema distribuible, donde la consola y los agentes pueden situarse en cualquier PC con conexión a Internet.
- El diseño debe favorecer la usabilidad del sistema, facilitando, en el mayor grado posible, el trabajo del usuario.

### **3.3.4 Otros Requerimientos**

Se debe tener en cuenta que el ancho de red es limitado, por lo tanto, cuando se está realizando un testeo de una web no se debe utilizar la conexión a Internet para comunicarse con el servidor.

Las comunicaciones entre agente y consola deben ser lo más ligero posible para minimizar los tiempos.



## **4 Especificación**

### **4.1 Modelo de casos de uso**

El modelo de casos de uso tiene como objetivo describir que funciones aportan el sistema para cada uno de los actores de éste.

Debido al gran número de funcionalidades, se ha dividido el diagrama de casos de uso en 4 subdiagramas, 3 para los casos de uso de la parte cliente de la aplicación, y uno para el núcleo del sistema.

Los diagramas de la parte cliente son:

- Mantenimiento.
- Grabación.
- Testeo-Monitorización.

Por su complejidad, los casos de uso del sistema que se muestra del núcleo del sistema, se ven reflejados por flechas entre procesos automáticos (que podríamos considerar en este caso especial, actores) Es importante remarcar que en el núcleo del sistema, los casos de uso no tienen ningún actor asignable, se suceden las operaciones y la comunicación entre agente y servidor, entre agente y las navegaciones, respondiendo a la tipología de operativa a realizar y a la lógica de operación.

#### **4.1.1 Mantenimiento**

Los casos de uso que hacen referencia al mantenimiento, tienen lugar en la consola, el actor principal es el administrador del ISM.

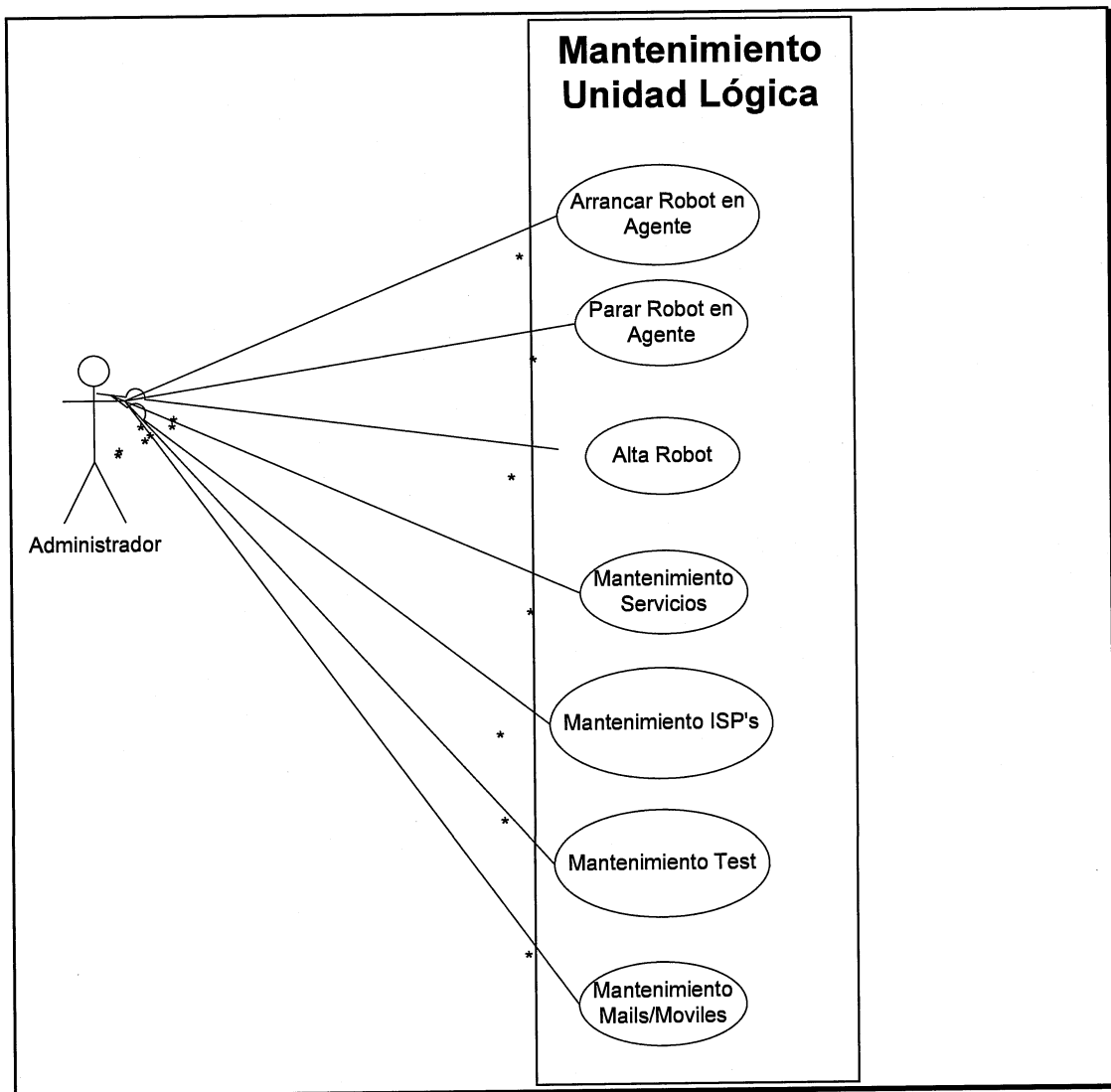
Estos casos de uso actúan dentro de la consola de ISM y se dividen en dos tipos de casos de uso, los cuales se dividirán en otros tantos dependiendo del elemento sobre el cual se actúa:

- Diagramas de Mantenimiento de Unidad Lógica, corresponden a los mantenimientos genéricos del ISM, y se actúa sobre la barra principal de la consola de administración de ISM.
- Diagramas del Árbol de Ejecución, estos se dividen por los elementos que intervienen en el árbol de ejecución:
  - Robot
  - Perfil

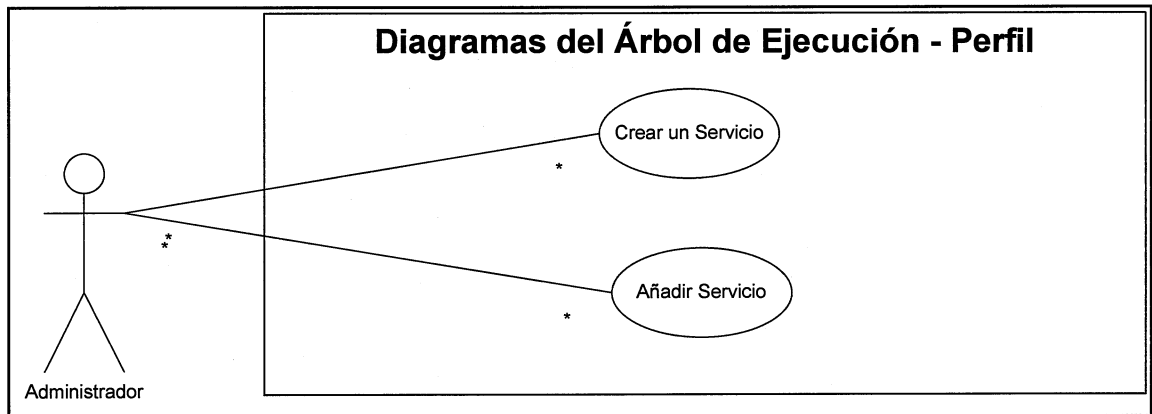
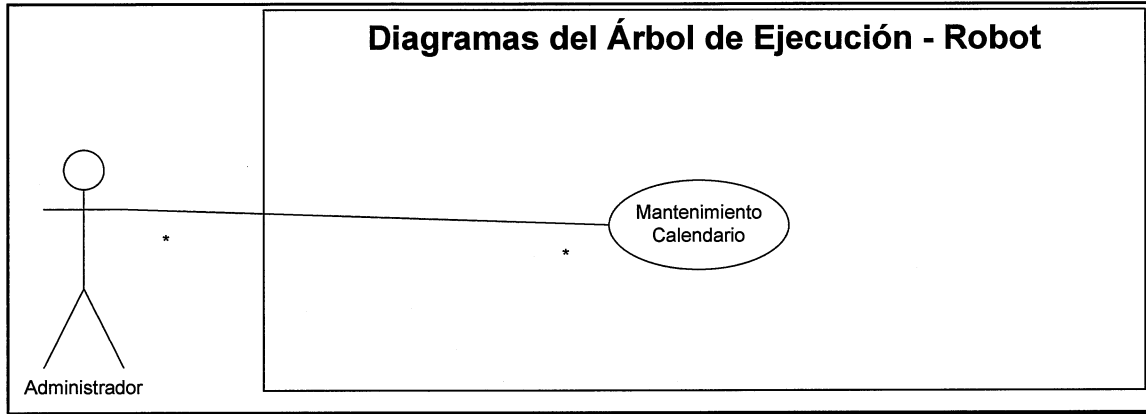
- Servicio
- ISP
- Test
- Script

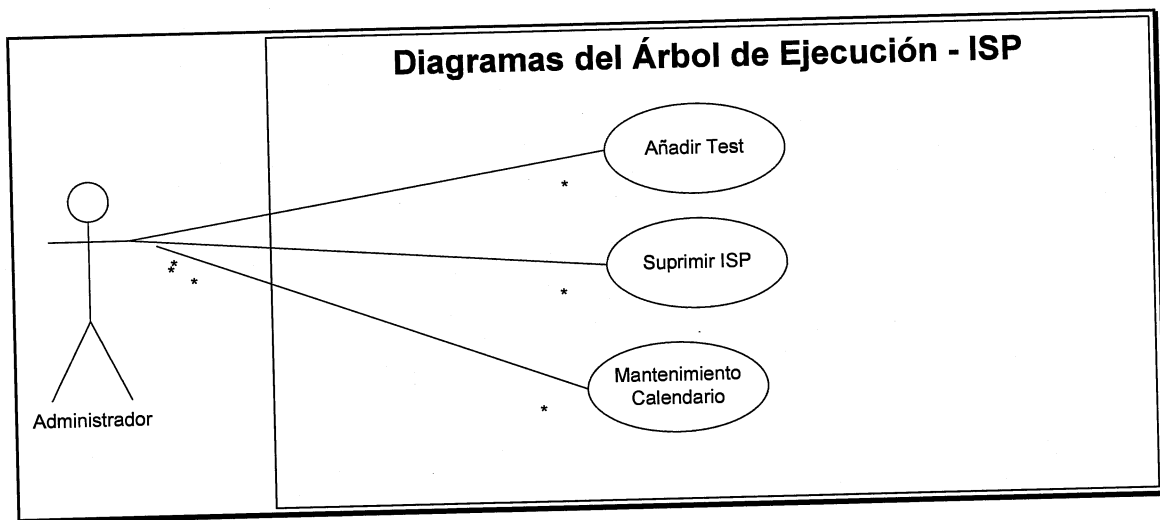
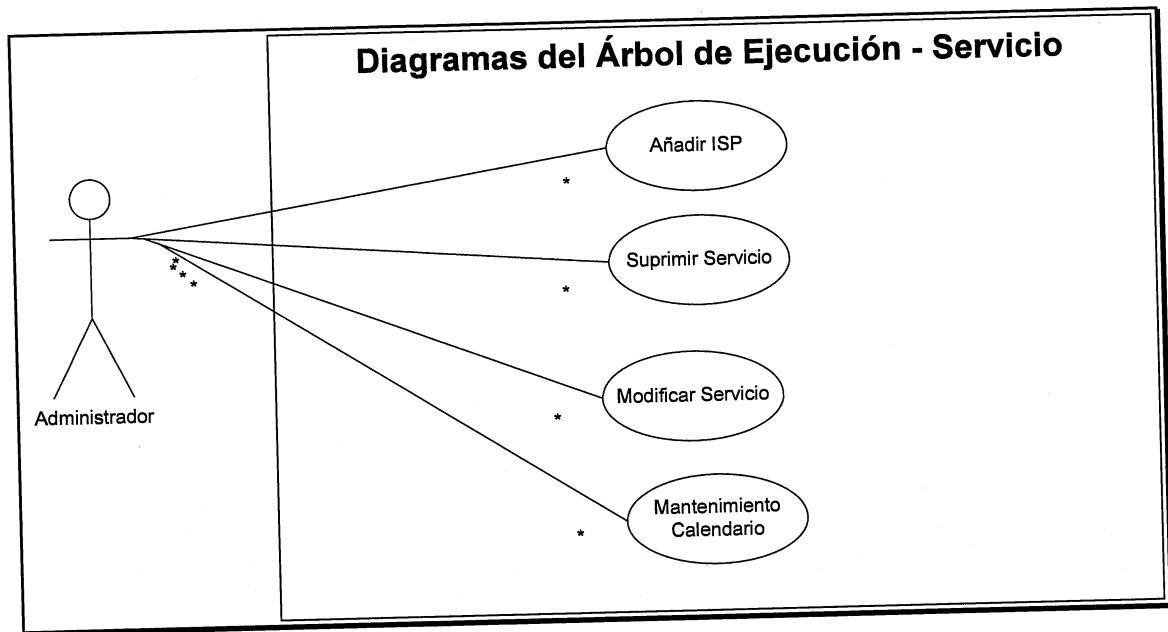
4.1.1.1 Diagrama de Casos de Uso

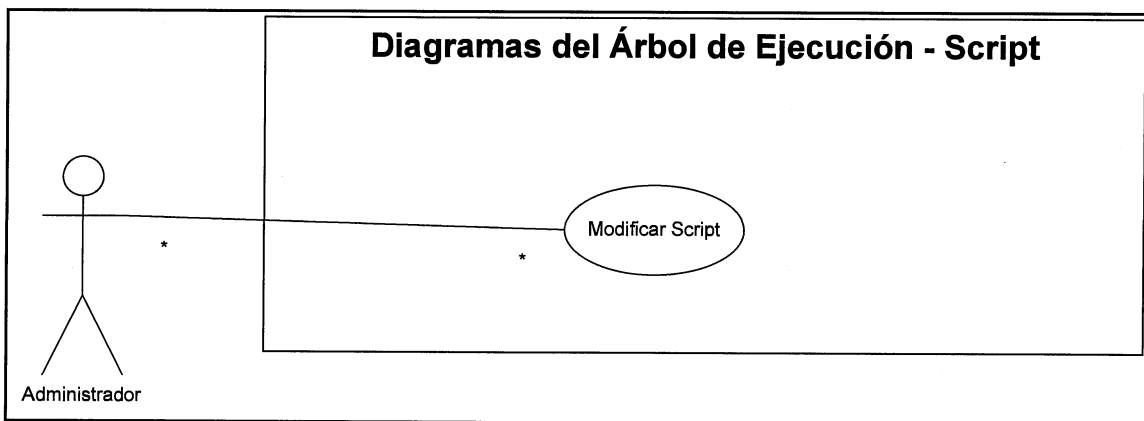
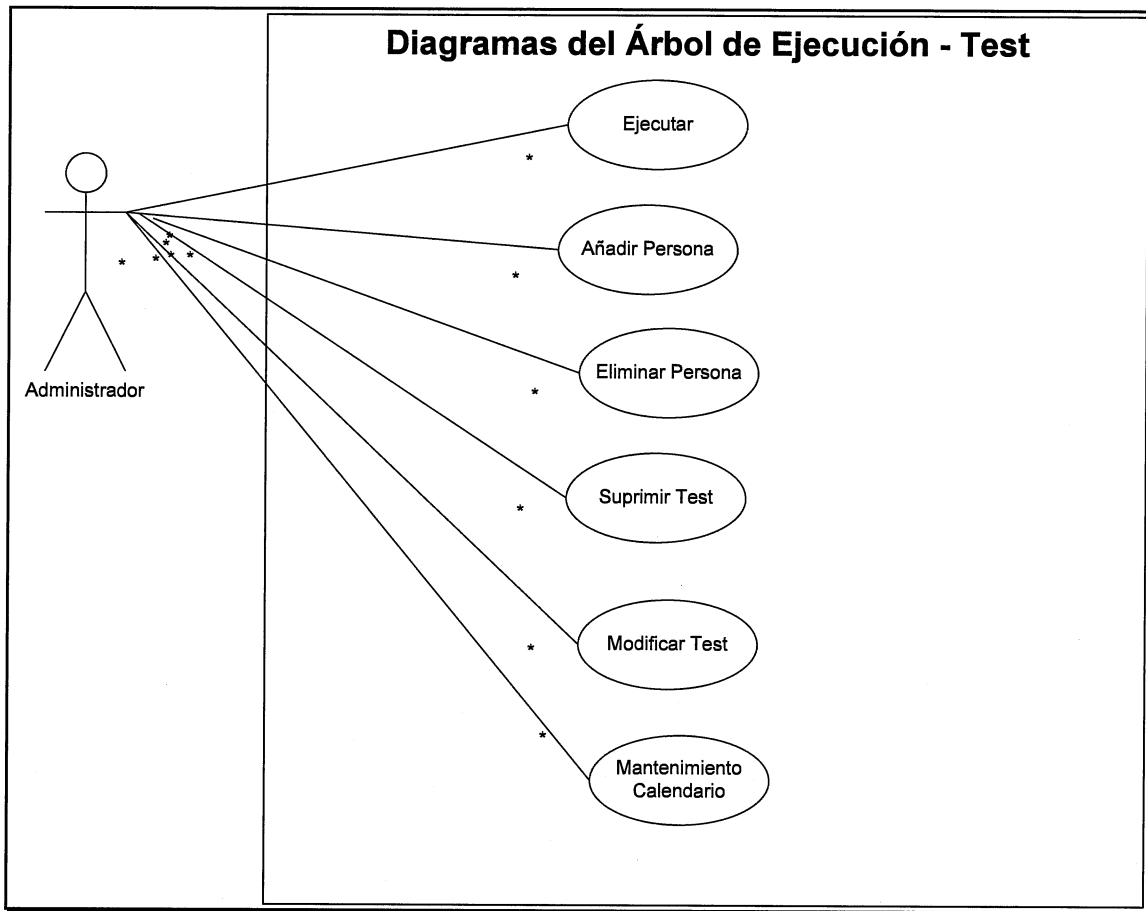
- Diagramas de Mantenimiento de Unidad Lógica.



- Diagramas del Árbol de Ejecución.





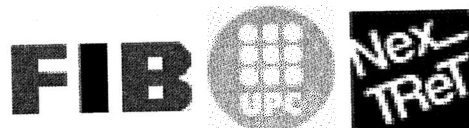


#### 4.1.1.2 Especificación de casos de uso

##### Arrancar Robot en Agente

**Actores:** Administrador

**Propósito:** Iniciar la monitorización de los tests programados en el robot que ejecuta un determinado agente.



**Resumen:** El usuario que está conectado a la consola, con un robot seleccionado, desea que se inicie la monitorización de un agente.

**Sucesión típica de acontecimientos:**

**Acciones del actor:**

1. El usuario solicita que se arranque la monitorización del agente.
2. El usuario visualiza que el resultado en la botonera teniendo el botón de stop accesible

**Respuesta del sistema:**

1. El sistema cambia el estado del Robot en la base de datos mediante una petición al Web Service.
2. El Agente asociado periódicamente consulta, por medio del Web Service, su estado y arranca la monitorización en este caso.

**Parar Robot en Agente**

**Actores:** Administrador

**Propósito:** Parar la monitorización de los tests programados en el robot que ejecuta un determinado agente.

**Resumen:** El usuario que está conectado a la consola, con un robot seleccionado, desea que se para la monitorización de una gente.

**Sucesión típica de acontecimientos:**

**Acciones del actor:**

1. El usuario solicita que se para la monitorización del agente.
2. El usuario visualiza que el resultado en la botonera teniendo el botón de play accesible

**Respuesta del sistema:**

1. El sistema cambia el estado del Robot en la base de datos mediante una petición al Web Service.
2. El Agente asociado periódicamente consulta, por medio del Web Service, su estado y para la monitorización cuando finalice el último test que está ejecutando.

**Alta de Robot**

**Actores:** Administrador

**Propósito:** Creación de un nuevo robot que aglutinara las sesiones.

**Resumen:** El usuario que está conectado a la consola desea crear un nuevo robot.

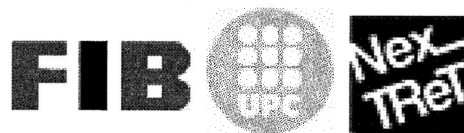
**Sucesión típica de acontecimientos:**

**Acciones del actor:**

1. El usuario solicita el alta de un nuevo robot.
2. El usuario rellena los datos necesarios.
3. El usuario ve el resultado del alta en el listado de robots

**Respuesta del sistema:**

1. El sistema crea un nuevo robot mediante una petición al Web Service.



## **Mantenimiento de Servicio**

**Actores:** Administrador

**Propósito:** Creación, modificación o eliminado de un nuevo servicio que aglutinará las sesiones.

**Resumen:** El usuario que está conectado a la consola desea crear, modificar o eliminar una nueva unidad lógica de agrupación de sesiones.

**Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita el alta, modificación o baja de un nuevo servicio.
2. El usuario rellena los datos necesarios.
3. El usuario ve el resultado del alta en el listado de servicios

Respuesta del sistema:

1. El sistema crea, modifica o elimina un nuevo servicio mediante una petición al Web Service.

## **Mantenimiento de ISP**

**Actores:** Administrador

**Propósito:** Creación, modificación o eliminado de un nuevo ISP.

**Resumen:** El usuario que está conectado a la consola desea crear, modificar o eliminar un nuevo ISP desde donde se monitorizarán los tests.

**Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita el alta, modificación o baja de un nuevo ISP.
2. El usuario rellena los datos necesarios.
3. El usuario ve el resultado del alta en el listado de ISP's

Respuesta del sistema:

1. El sistema crear, modifica o elimina un nuevo ISP mediante una petición al Web Service.

## **Mantenimiento de Test**

**Actores:** Administrador

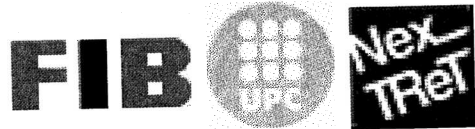
**Propósito:** Creación, modificación o eliminado de un nuevo Test.

**Resumen:** El usuario que está conectado a la consola desea crear, modificar o eliminar un nuevo Test. Este caso de uso conlleva una gran cantidad de acciones, con lo cual, el autor, ha creado un apartado específico para la grabación de tests.

**Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita el alta, modificación o baja de un nuevo Test.



2. El usuario rellena los datos necesarios (Este caso de uso se amplía en el apartado de grabación).
3. El usuario ve el resultado del alta en el listado de ISP's

Respuesta del sistema:

1. El sistema crear, modifica o elimina un nuevo Test mediante peticiones al Web Service.

### **Mantenimiento de Mails/Móviles**

**Actores:** Administrador

**Propósito:** Creación, modificación o eliminado una persona que recibirá alertas.

**Resumen:** El usuario que está conectado a la consola desea crear, modificar o eliminar una persona que recibirá alertas.

**Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita el alta, modificación o baja de una nueva persona
2. El usuario rellena los datos necesarios.
3. El usuario ve el resultado del alta en el listado de personas.

Respuesta del sistema:

1. El sistema crear, modifica o elimina una persona mediante una petición al Web Service.

### **Robot – Mantenimiento Calendario**

**Actores:** Administrador

**Propósito:** Crear, modificar o eliminar un calendario de ejecución para el robot completo.

**Resumen:** El usuario que está conectado a la consola y tiene seleccionado un robot desea crear, modificar o eliminar un calendario para un robot.

**Sucesión típica de acontecimientos:**

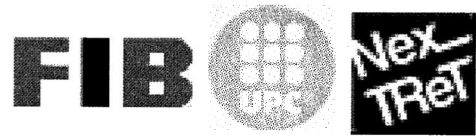
Acciones del actor:

1. El usuario solicita el alta, modificación o baja de un calendario para un robot persona
2. El usuario rellena los datos necesarios.
3. El usuario ve el resultado del alta en el listado de calendarios del robot.

Respuesta del sistema:

1. El sistema crear, modifica o elimina un calendario mediante una petición al Web Service.
2. El agente que disponga de ese robot, cuando se sincroniza mediante una petición al Web Service tiene en cuenta el calendario a la hora de su ejecución.





### **Perfil – Añadir Servicio**

**Actores:** Administrador

**Propósito:** Añadir un servicio al árbol de ejecución

**Resumen:** El usuario que está conectado a la consola, selecciona el perfil del robot del árbol de ejecución y desea añadir un servicio de los existentes.

#### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita añadir, al árbol de ejecución, un servicio de los existentes
2. El usuario ve el resultado del alta en el árbol de ejecución, cuando se añade el servicio y todos los elementos del servicio.

Respuesta del sistema:

1. El sistema obtiene todos los datos del servicio y de sus elementos mediante una petición al Web Service y la consola actualiza el árbol de ejecución.
2. El agente que disponga de ese robot, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.

### **Perfil – Crear Servicio**

**Actores:** Administrador

**Propósito:** Crear un servicio en el árbol de ejecución

**Resumen:** El usuario que está conectado a la consola, selecciona el perfil del robot del árbol de ejecución y desea crear y añadir un servicio.

#### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita crear y añadir, al árbol de ejecución, un servicio de los existentes
2. El usuario rellena los datos necesarios.
3. El usuario ve el resultado del alta en el árbol de ejecución, cuando se añade el servicio sin elementos.

Respuesta del sistema:

1. El sistema realiza el alta del servicio y obtiene todos los datos del servicio mediante una petición al Web Service. La consola actualiza el árbol de ejecución.
2. El agente que disponga de ese robot, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.

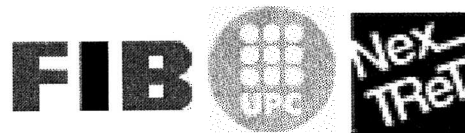
### **Servicio – Añadir ISP**

**Actores:** Administrador

**Propósito:** Añadir un ISP al servicio en el árbol de ejecución

**Resumen:** El usuario que está conectado a la consola, selecciona un servicio del robot en el árbol de ejecución y desea añadir un ISP de los existentes.

#### **Sucesión típica de acontecimientos:**



**Acciones del actor:**

1. El usuario solicita añadir, al árbol de ejecución, un ISP de los existentes
2. El usuario ve el resultado del alta en el árbol de ejecución, cuando se añade el ISP al servicio.

**Respuesta del sistema:**

1. El sistema obtiene todos los datos del ISP mediante una petición al Web Service y la consola actualiza el árbol de ejecución.
2. El agente que disponga de ese robot, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.

**Servicio – Suprimir Servicio**

**Actores:** Administrador

**Propósito:** Suprimir un servicio del árbol de ejecución.

**Resumen:** El usuario que está conectado a la consola, selecciona un servicio del robot en el árbol de ejecución y desea suprimirlo, junto a todos sus elementos, de la monitorización.

**Sucesión típica de acontecimientos:**

**Acciones del actor:**

1. El usuario solicita suprimir, del árbol de ejecución, un servicio de los existentes
2. El usuario ve el resultado del alta en el árbol de ejecución, cuando se suprime el servicio y todos sus elementos.

**Respuesta del sistema:**

1. El sistema envía una petición para suprimir del árbol de al Web Service y la consola actualiza el árbol de ejecución.
2. El agente que disponga de ese robot, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.

**Servicio – Modificar Servicio**

**Actores:** Administrador

**Propósito:** Modificar los parámetros de un servicio del árbol de ejecución.

**Resumen:** El usuario que está conectado a la consola, selecciona un servicio del robot en el árbol de ejecución y desea modificarlo.

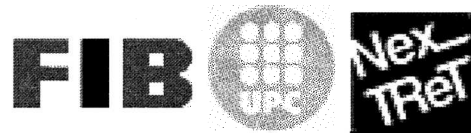
**Sucesión típica de acontecimientos:**

**Acciones del actor:**

1. El usuario solicita modificar un servicio de los existentes
2. El usuario modifica los parámetros.
3. El usuario ve el resultado de la modificación en el árbol de ejecución.

**Respuesta del sistema:**

1. El sistema envía una petición para modificar el servicio al Web Service y la consola actualiza el árbol de ejecución.
2. El agente que disponga de ese servicio, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.



## **Servicio – Mantenimiento Calendario**

**Actores:** Administrador

**Propósito:** Crear, modificar o eliminar un calendario de ejecución para el servicio completo.

**Resumen:** El usuario que está conectado a la consola y tiene seleccionado un servicio desea crear, modificar o eliminar un calendario.

### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita el alta, modificación o baja de un calendario para un servicio.
2. El usuario rellena los datos necesarios.
3. El usuario ve el resultado del alta en el listado de calendarios del servicio.

Respuesta del sistema:

1. El sistema crea, modifica o elimina un calendario mediante una petición al Web Service.
2. El agente que disponga de ese servicio, cuando se sincroniza mediante una petición al Web Service tiene en cuenta el calendario a la hora de su ejecución.

## **ISP – Añadir Test**

**Actores:** Administrador

**Propósito:** Añadir un Test al ISP en el árbol de ejecución

**Resumen:** El usuario que está conectado a la consola, selecciona un ISP del robot, en el árbol de ejecución, y desea añadir un Test de los existentes.

### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita añadir, al árbol de ejecución, un Test de los existentes
2. El usuario ve el resultado del alta en el árbol de ejecución, cuando se añade el Test al ISP, crea una sesión.

Respuesta del sistema:

1. El sistema obtiene todos los datos del Test mediante una petición al Web Service y la consola actualiza el árbol de ejecución.
2. El sistema crea una nueva unidad y relación en la base de datos, llamada sesión.
3. El agente que disponga de ese robot, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.

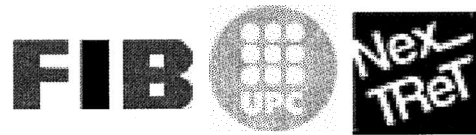
## **ISP – Suprimir ISP**

**Actores:** Administrador

**Propósito:** Suprimir un ISP del árbol de ejecución.

**Resumen:** El usuario que está conectado a la consola, selecciona un ISP del robot en el árbol de ejecución y desea suprimirlo, junto a todos sus elementos, de la monitorización.

### **Sucesión típica de acontecimientos:**



**Acciones del actor:**

1. El usuario solicita suprimir, del árbol de ejecución, un ISP de los existentes
2. El usuario ve el resultado del alta en el árbol de ejecución, cuando se suprime el ISP y todos sus elementos.

**Respuesta del sistema:**

1. El sistema envía una petición para suprimir del árbol el ISP al Web Service y la consola actualiza el árbol de ejecución.
2. El agente que disponga de ese robot, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.

**ISP – Mantenimiento Calendario**

**Actores:** Administrador

**Propósito:** Crear, modificar o eliminar un calendario de ejecución para el ISP completo.

**Resumen:** El usuario que está conectado a la consola y tiene seleccionado un ISP desea crear, modificar o eliminar un calendario.

**Sucesión típica de acontecimientos:**

**Acciones del actor:**

1. El usuario solicita el alta, modificación o baja de un calendario para un ISP.
2. El usuario rellena los datos necesarios.
3. El usuario ve el resultado del alta en el listado de calendarios del ISP.

**Respuesta del sistema:**

1. El sistema crea, modifica o elimina un calendario mediante una petición al Web Service.
2. El agente que disponga de ese ISP, cuando se sincroniza mediante una petición al Web Service tiene en cuenta el calendario a la hora de su ejecución.

**Test – Ejecutar**

**Actores:** Administrador

**Propósito:** Comprobar el funcionamiento de un test.

**Resumen:** El usuario que está conectado a la consola y tiene seleccionado un Test desea comprobar el funcionamiento del test para conocer si funciona correctamente o donde está el fallo en caso de problemas en la monitorización.

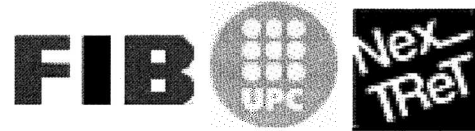
**Sucesión típica de acontecimientos:**

**Acciones del actor:**

1. El usuario solicita la ejecución de un test.
2. El usuario selecciona el ISP por donde desea comprobar el Test.
3. El usuario ve la navegación del test mediante un Internet Explorer.

**Respuesta del sistema:**

1. El sistema lanza la ejecución del test.
2. El sistema informa del resultado correcto de la ejecución o del resultado incorrecto, mostrando el problema encontrado para no llevar a cabo totalmente la prueba.



### **Test – Añadir Persona**

**Actores:** Administrador

**Propósito:** Añadir una persona al Test, para que, esta persona, reciba notificaciones de error del test.

**Resumen:** El usuario que está conectado a la consola, selecciona un Test del robot, en el árbol de ejecución, y desea añadir una persona a la lista de notificaciones en caso de error.

#### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita añadir una persona al Test.
2. El usuario ve la lista de personas y selecciona una de ellas.
3. El usuario ve el resultado del alta en el árbol de ejecución.

Respuesta del sistema:

1. El sistema obtiene todos los datos de las personas y las muestra.
2. El sistema mediante una petición al Web Service, añade a la persona al test.
3. El agente que disponga de ese test, cuando se sincroniza mediante una petición al Web Service actualiza su lista de personas que reciben alertas.

### **Test – Suprimir Persona**

**Actores:** Administrador

**Propósito:** Suprimir una persona de la lista de personas que reciben notificación.

**Resumen:** El usuario que está conectado a la consola, selecciona un Test del robot en el árbol de ejecución y desea suprimir alguna persona seleccionándola de un listado.

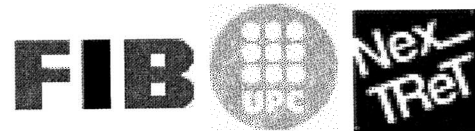
#### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita suprimir una persona del Test.
2. El usuario ve el resultado del alta en el árbol de ejecución, cuando se suprime la persona del Test.

Respuesta del sistema:

1. El sistema envía una petición para suprimir la persona del Test al Web Service y la consola actualiza el árbol de ejecución.
2. El agente que disponga de ese test, cuando se sincroniza mediante una petición al Web Service actualiza su lista de personas que reciben alertas.



## Test – Suprimir Test

**Actores:** Administrador

**Propósito:** Suprimir un Test del árbol de ejecución.

**Resumen:** El usuario que está conectado a la consola, selecciona un Test del robot en el árbol de ejecución y desea suprimirlo, junto a todos sus elementos, de la monitorización.

### Sucesión típica de acontecimientos:

Acciones del actor:

1. El usuario solicita suprimir, del árbol de ejecución, un Test de los existentes
2. El usuario ve el resultado del alta en el árbol de ejecución, cuando se suprime el Test y todos sus elementos.

Respuesta del sistema:

3. El sistema envía una petición para suprimir del árbol el Test al Web Service y la consola actualiza el árbol de ejecución.
4. El agente que disponga de ese robot, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.

## Test – Modificar Test

**Actores:** Administrador

**Propósito:** Modificar los parámetros de un test del árbol de ejecución.

**Resumen:** El usuario que está conectado a la consola, selecciona un test del robot en el árbol de ejecución y desea modificarlo.

### Sucesión típica de acontecimientos:

Acciones del actor:

1. El usuario solicita modificar un test de los existentes
2. El usuario modifica los parámetros.
3. El usuario ve el resultado de la modificación en el árbol de ejecución.

Respuesta del sistema:

1. El sistema envía una petición para modificar el test al Web Service y la consola actualiza el árbol de ejecución.
2. El agente que disponga de ese test, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.

## Test – Mantenimiento Calendario

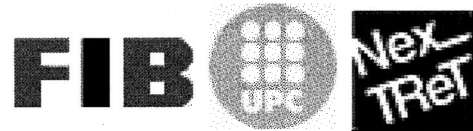
**Actores:** Administrador

**Propósito:** Crear, modificar o eliminar un calendario de ejecución para el Test.

**Resumen:** El usuario que está conectado a la consola y tiene seleccionado un test desea crear, modificar o eliminar un calendario para la tupla Test e ISP (sesión) donde está el Test.

### Sucesión típica de acontecimientos:

Acciones del actor:



1. El usuario solicita el alta, modificación o baja de un calendario para un Test.
2. El usuario rellena los datos necesarios.
3. El usuario ve el resultado del alta en el listado de calendarios del Test.

Respuesta del sistema:

1. El sistema crea, modifica o elimina un calendario mediante una petición al Web Service.
2. El agente que disponga de esa sesión, cuando se sincroniza mediante una petición al Web Service tiene en cuenta el calendario a la hora de su ejecución.

### **Script – Modificar Script**

**Actores:** Administrador

**Propósito:** Modificar los parámetros de un Script de un test.

**Resumen:** El usuario que está conectado a la consola, selecciona un Script de un test en el árbol de ejecución y desea modificarlo.

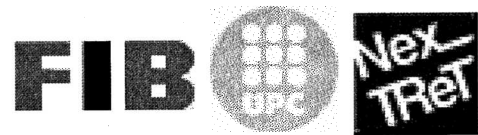
**Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita modificar un script de los existentes
2. El usuario modifica los parámetros.
3. El usuario ve el resultado de la modificación en el árbol de ejecución.

Respuesta del sistema:

1. El sistema envía una petición para modificar el script al Web Service y la consola actualiza el árbol de ejecución.
2. El agente que disponga del test asociado al script , cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.



#### **4.1.2 Grabación**

Los casos de uso que hacen referencia a la grabación, tienen lugar en la consola, en el apartado del Grabador de Scripts, el actor principal es el administrador del ISM.

Para arrancar el Grabador de Scripts es necesario, que entre en escena, el caso de uso “Mantenimiento de Test”.

Una vez lanzado este caso de uso, se abre un interfaz que se es donde se inician los casos de uso de grabación.

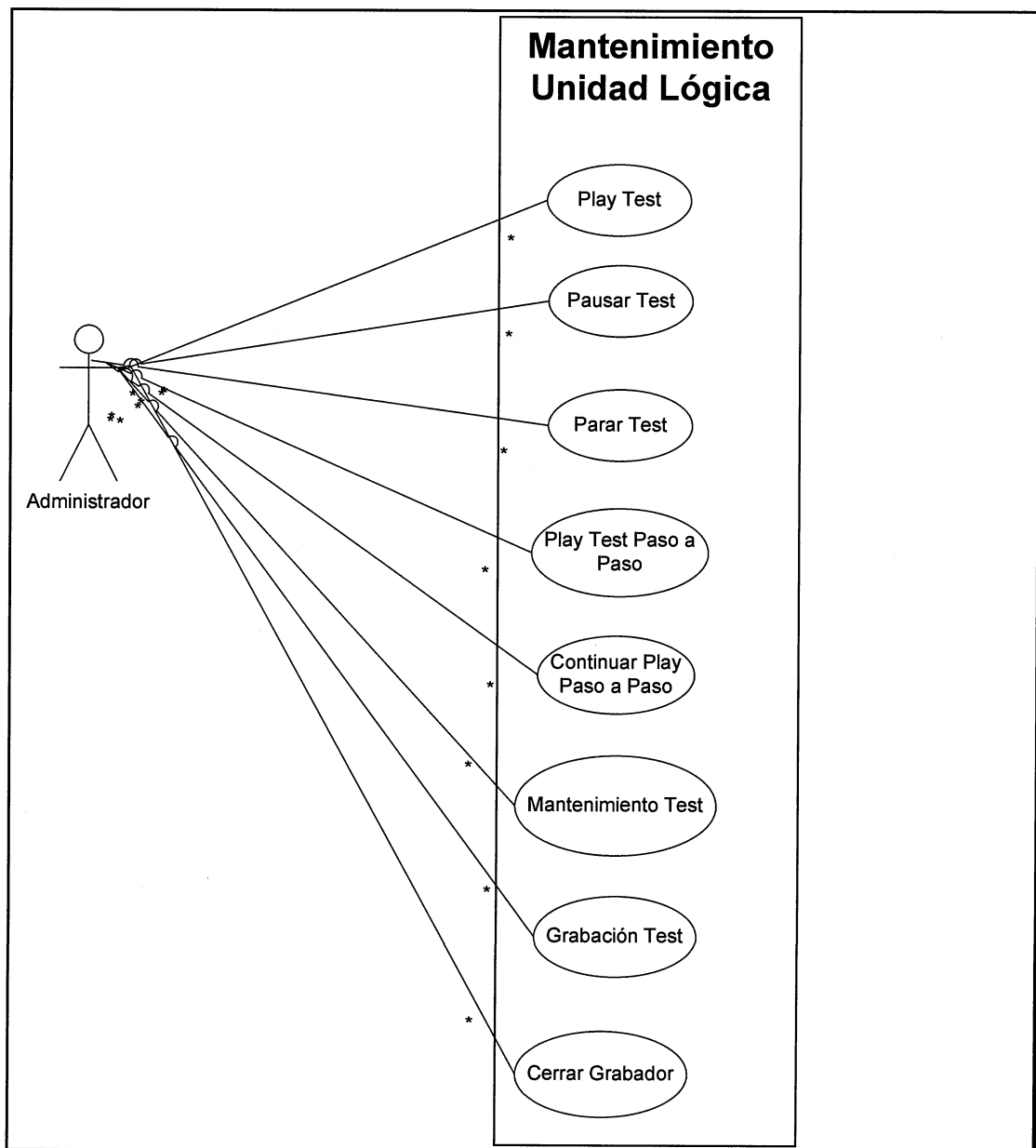
Se dividen en tres tipos de casos de uso, los cuales se dividirán en otros tantos dependiendo del elemento sobre el cual se actúa:

- Diagramas de Mantenimiento de Unidad Lógica, corresponden a los mantenimientos genéricos del ISM, y se actúa sobre la barra principal de la consola de administración de ISM.
- Diagramas del Árbol de Ejecución, estos se dividen por los elementos que intervienen en el árbol de ejecución:
  - Test
  - Script
- Diagramas de Grabación en IE.

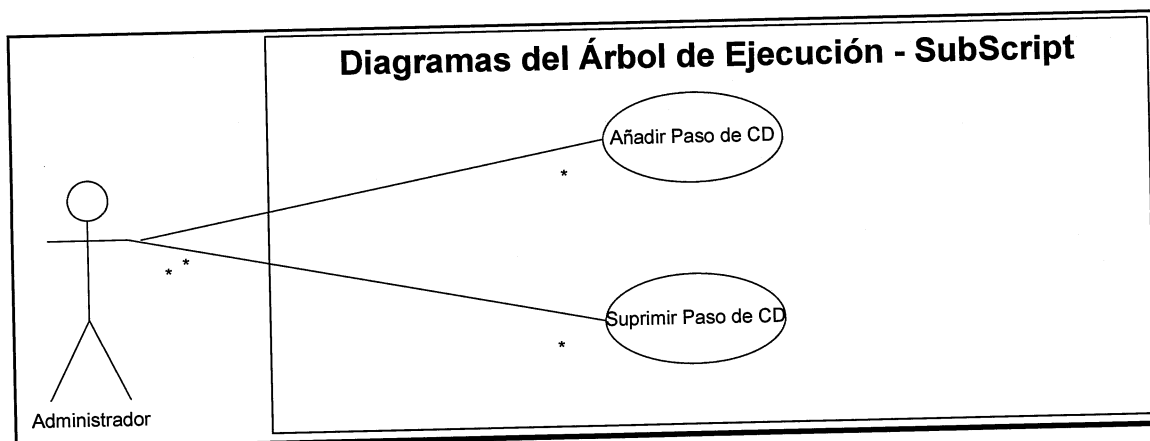
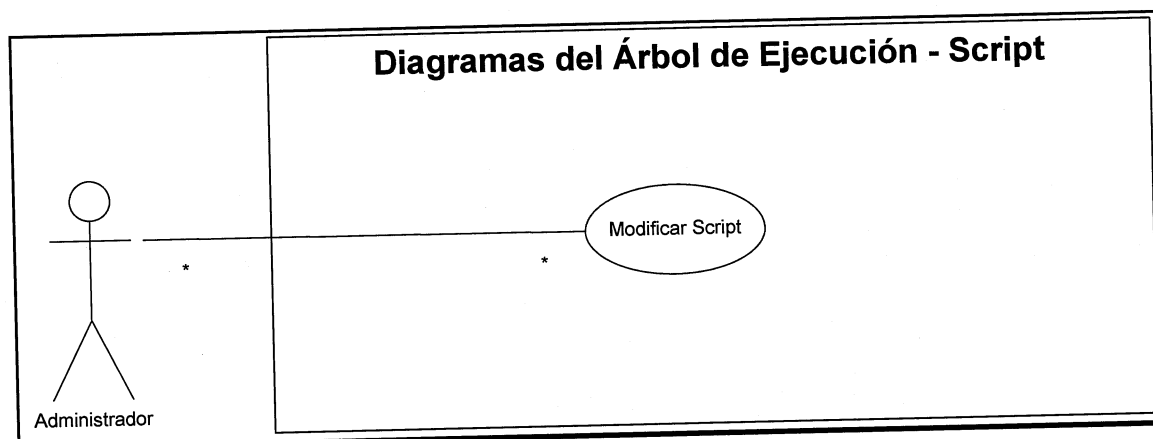
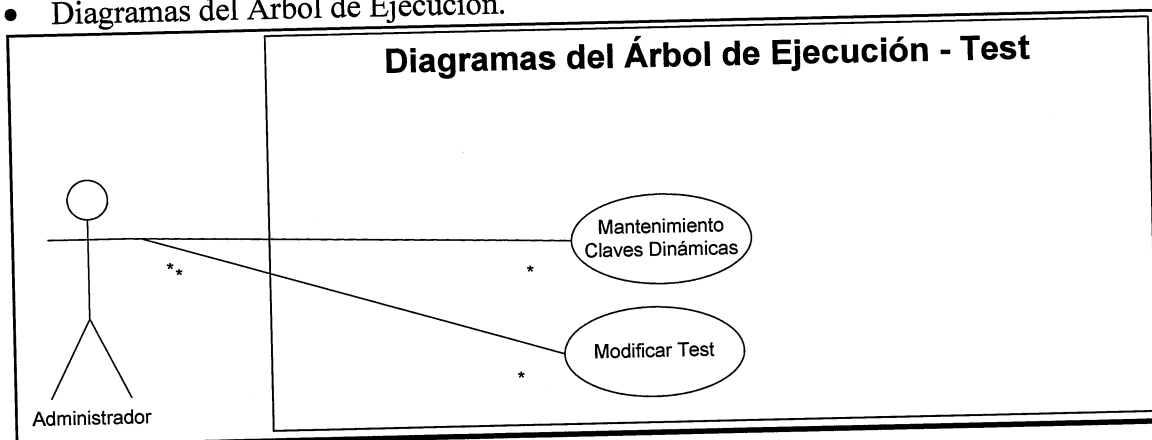


#### 4.1.2.1 Diagrama de Casos de Uso

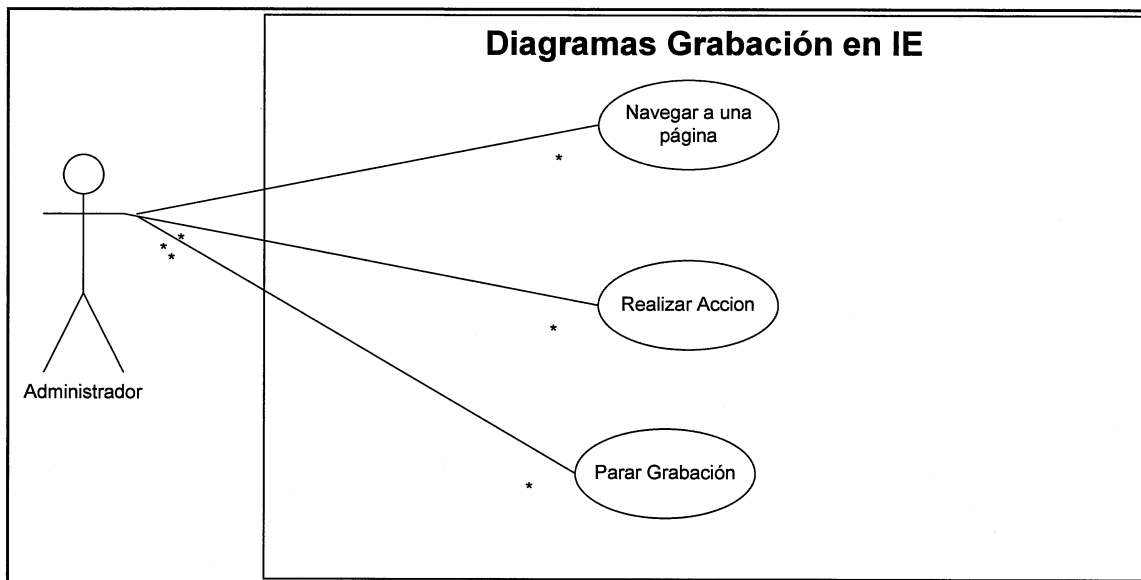
- Diagramas de Mantenimiento de Unidad Lógica.



- Diagramas del Árbol de Ejecución.



- Diagramas de Grabación en IE.



#### 4.1.2.2 Especificación de casos de uso

##### Play Test

**Actores:** Administrador

**Propósito:** Comprobar el funcionamiento de un test.

**Resumen:** El usuario que está conectado al grabador de Test y tiene seleccionado un Test desea comprobar el funcionamiento del test para conocer si funciona correctamente o donde está el fallo en caso de problemas en la monitorización.

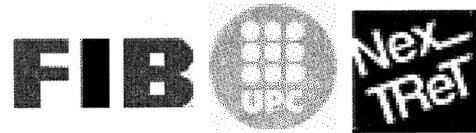
**Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita la ejecución de un test.
2. El usuario selecciona el ISP por donde desea comprobar el Test.
3. El usuario ve la navegación del test mediante un Internet Explorer.

Respuesta del sistema:

1. El sistema lanza la ejecución del test.
2. El sistema informa del resultado correcto de la ejecución o del resultado incorrecto, mostrando el problema encontrado para no llevar a cabo totalmente la prueba.



### **Pausar Test**

**Actores:** Administrador

**Propósito:** Durante la ejecución de test, realizar una pausa.

**Resumen:** El usuario que está conectado al grabador de Test y está ejecutando un test desea realizar una pausa en la ejecución del Test.

#### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario, anteriormente, ha solicita la ejecución de un test.
2. El usuario solicita el pausado de la ejecución de una prueba.

Respuesta del sistema:

1. El sistema pausa la ejecución del test, dejando la navegación en el último punto ejecutado, si el test está cargando una página se espera hasta su finalización y para la navegación.

### **Parar Test**

**Actores:** Administrador

**Propósito:** Durante la ejecución de test, realizar una parada de la ejecución.

**Resumen:** El usuario que está conectado al grabador de Test y está ejecutando un test desea parar la ejecución del Test.

#### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario, anteriormente, ha solicita la ejecución de un test.
2. El usuario solicita parar la ejecución de una prueba.

Respuesta del sistema:

1. El sistema para la ejecución del test, cerrando el navegador.

### **Play Test Paso a Paso**

**Actores:** Administrador

**Propósito:** Ejecutar el test acción a acción

**Resumen:** El usuario desea ejecutar una prueba acción a acción para comprobar el funcionamiento de todas las acciones o subscripts por separado. Típicamente, se utiliza para realizar la regrabación de la prueba desde un punto en concreto.

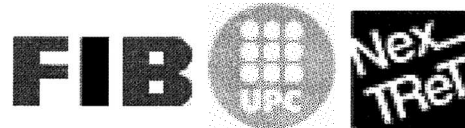
#### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita la ejecución de una prueba paso a paso.

Respuesta del sistema:

1. El sistema ejecuta el test hasta la primera carga de la página.



### **Continuar Play Test Paso a Paso**

**Actores:** Administrador

**Propósito:** Continuar la ejecución del test acción a acción

**Resumen:** El usuario desea continuar la ejecución del test con la siguiente acción a ejecutar.

#### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita la ejecución de la siguiente acción o scripts.

Respuesta del sistema:

2. El sistema ejecuta la siguiente acción y se para.

### **Mantenimiento de Test**

**Actores:** Administrador

**Propósito:** Creación, modificación o eliminado de un nuevo.

**Resumen:** El usuario que está conectado a la consola desea crear, modificar o eliminar un nuevo test.

#### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita el alta, modificación o baja de un nuevo test.
2. El usuario rellena los datos necesarios.
3. El usuario ve el resultado del alta en el listado de test

Respuesta del sistema:

1. El sistema crea, modifica o elimina un nuevo servicio mediante una petición al Web Service.
2. El sistema comprueba si se puede eliminar el test, si el test está dado de alta en alguna sesión no se puede eliminar.

### **Grabación de Test**

**Actores:** Administrador

**Propósito:** Grabar o regrabar una nueva prueba.

**Resumen:** El usuario que está conectado a la consola desea grabar o regrabar.

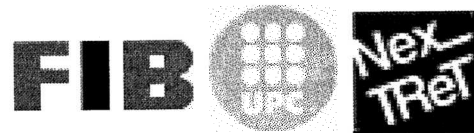
#### **Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita la grabación de una prueba.
2. El usuario selecciona si desea grabar o regrabar.
3. El usuario rellena los datos necesarios y la URL inicial.
4. El usuario comprueba que se inicia un IE y se navega a la URL indicada.

Respuesta del sistema:

1. El sistema registra la navegación a realizar.
2. El sistema abre un IE y navega a la URL indicada.



## Cerrar Generador de Scripts

**Actores:** Administrador

**Propósito:** Cerrar el grabador y volver a la consola principal

**Resumen:** El usuario que está conectado al grabador ha finalizado su tarea y desea volver a la consola.

### Sucesión típica de acontecimientos:

Acciones del actor:

1. El usuario solicita el cierre del generador.
2. El usuario comprueba que se abre la consola principal.

Respuesta del sistema:

1. El sistema cierra el grabador.
2. El sistema, realiza las peticiones necesarias para abrir la consola con el robot preseleccionado.
3. El sistema abre la consola principal.

## Realizar acción sobre IE

**Actores:** Administrador

**Propósito:** Realizar una acción sobre el IE para que se grabe para su posterior monitorización.

**Resumen:** El usuario desea realizar una nueva acción sobre el IE, esta acción genera un nuevo subscript, dependiendo del tipo de acción o una nueva navegación y script si la acción genera una recarga de la página.

### Sucesión típica de acontecimientos:

Acciones del actor:

1. El usuario realiza una acción sobre el IE.
2. El usuario comprueba que en el árbol de ejecución aparece un nuevo subscript o un script si se genera una navegación.

Respuesta del sistema:

1. El sistema crear un nuevo subscript o script.
2. El navegador realiza la acción indicada.

## Navegare a una URL sobre IE

**Actores:** Administrador

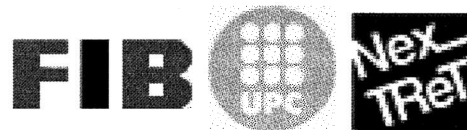
**Propósito:** Realizar navegaciones a URL sin vinculo.

**Resumen:** El usuario desea realizar una navegación a una nueva URL, sin necesidad de realizar una acción sobre el propio IE, el sistema genera un nuevo script, a partir de la nueva URL.

### Sucesión típica de acontecimientos:

Acciones del actor:

1. El usuario solicita la navegación a una nueva URL.
2. El usuario rellena los parámetros necesarios.



3. El usuario comprueba que en el árbol de ejecución aparece un nuevo script, y comprueba que el IE ha navegado a la URL indicada.

Respuesta del sistema:

1. El sistema crear un nuevo script.
2. El navegador realiza la acción indicada.

### **Finalizar la grabación**

**Actores:** Administrador

**Propósito:** Finalizar la grabación.

**Resumen:** El usuario finaliza la grabación de una prueba y el sistema envía la información al repositorio central

**Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita la finalización de la grabación.
2. El usuario acepta el aviso que indica que se van a enviar el test al repositorio central.
3. El usuario comprueba que el test se ha enviado, visualizando un aviso.

Respuesta del sistema:

1. El sistema mediante peticiones al Web Service realizar el envío del nuevo test o de los cambios producidos.
2. Si el test está en algún agente, el sistema actualiza el árbol de ejecución realizando peticiones de sincronización al Web Service.

### **Test – Mantenimiento Claves Dinámicas**

**Actores:** Administrador

**Propósito:** Crear, modificar o eliminar claves dinámicas del Test.

**Resumen:** El usuario que tiene seleccionado un test desea crear, modificar o eliminar las claves dinámicas asociadas al Test.

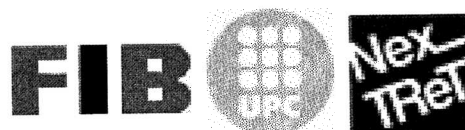
**Sucesión típica de acontecimientos:**

Acciones del actor:

1. El usuario solicita el alta, modificación o baja de las claves dinámicas para un Test.
2. El usuario rellena los datos necesarios.
3. El usuario ve el resultado en el listado de claves dinámicas.

Respuesta del sistema:

1. El sistema crea, modifica o elimina una clave dinámica mediante una petición al Web Service.
2. El agente que disponga del test, cuando se sincroniza mediante una petición al Web Service tiene en cuenta el calendario a la hora de su ejecución.



## Test – Modificar Test

**Actores:** Administrador

**Propósito:** Modificar los parámetros de un test del árbol de ejecución.

**Resumen:** El usuario que está conectado al grabador, selecciona un test del robot en el árbol de ejecución y desea modificarlo.

### Sucesión típica de acontecimientos:

Acciones del actor:

1. El usuario solicita modificar un test de los existentes
2. El usuario modifica los parámetros.
3. El usuario ve el resultado de la modificación en el árbol de ejecución.

Respuesta del sistema:

1. El sistema envía una petición para modificar el test al Web Service y el grabador actualiza el árbol de ejecución.
2. El agente que disponga de ese test, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.

## Subscript – Añadir Paso de CD

**Actores:** Administrador

**Propósito:** Asociar un paso de una clave dinámica a un subscript.

**Resumen:** El usuario desea que ese subscript sea dinámico mediante uno de los pasos de la clave dinámica.

### Sucesión típica de acontecimientos:

Acciones del actor:

1. El usuario solicita la asociación de subscript y paso
2. El usuario ve el resultado de la modificación en el árbol de ejecución.

Respuesta del sistema:

1. El sistema envía una petición para realizar la acción al Web Service y la consola actualiza el árbol de ejecución.
2. El agente que disponga del test asociado al script, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.

## Subscript – Suprimir Paso de CD

**Actores:** Administrador

**Propósito:** Eliminar un paso de una clave dinámica a un subscript.

**Resumen:** El usuario desea que ese subscript deje de ser dinámico mediante uno de los pasos de la clave dinámica.

### Sucesión típica de acontecimientos:

Acciones del actor:

1. El usuario solicita la eliminación de la asociación de subscript y paso
2. El usuario ve el resultado de la modificación en el árbol de ejecución.

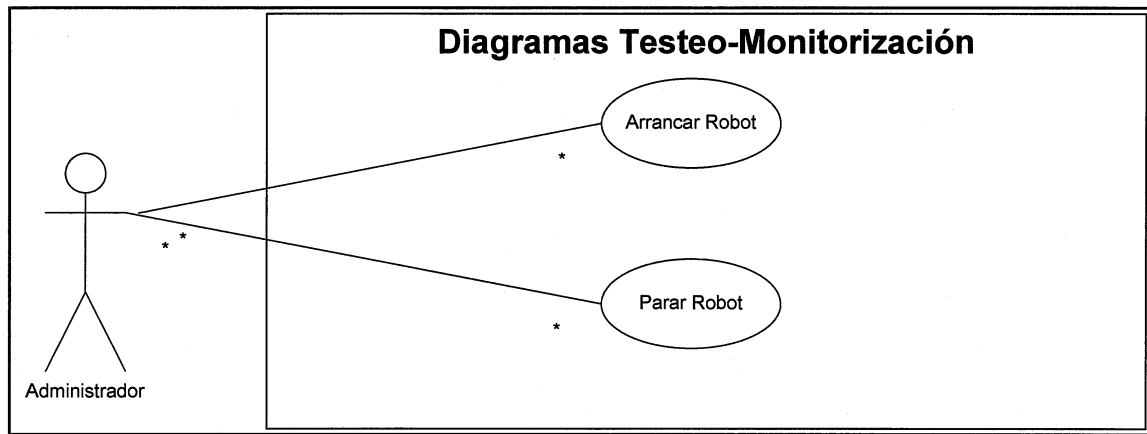


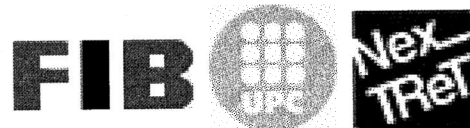
Respuesta del sistema:

1. El sistema envía una petición para realizar la acción al Web Service y la consola actualiza el árbol de ejecución.
2. El agente que disponga del test asociado al script, cuando se sincroniza mediante una petición al Web Service actualiza su árbol de ejecución.

#### 4.1.3 Testeo-Monitorización

##### 4.1.3.1 Diagrama de Casos de Uso





#### 4.1.3.2 Especificación de casos de uso

##### **Arrancar Robot**

**Actores:** Administrador

**Propósito:** Iniciar la monitorización de los tests programados en el robot que ejecuta el agente.

**Resumen:** El usuario desde el agente, con un robot seleccionado, desea que se inicie la monitorización.

##### **Sucesión típica de acontecimientos:**

###### Acciones del actor:

1. El usuario solicita que se arranque la monitorización del agente.
2. El usuario comprueba que se inicia la navegación de l primera prueba programada.

###### Respuesta del sistema:

1. El sistema cambia el estado del Robot en la base de datos mediante una petición al Web Service.
2. El sistema pide realizar una sincronización antes de comenzar con las navegaciones.
3. El Agente asociado periódicamente consulta, por medio del Web Service, su estado y arranca la monitorización en este caso.
4. El sistema arranca la primera navegación

##### **Parar Robot en Agente**

**Actores:** Administrador

**Propósito:** Parar la monitorización de los tests programados en el robot que ejecuta el agente.

**Resumen:** El usuario desde el agente, con un robot seleccionado, desea que se pare la monitorización.

##### **Sucesión típica de acontecimientos:**

###### Acciones del actor:

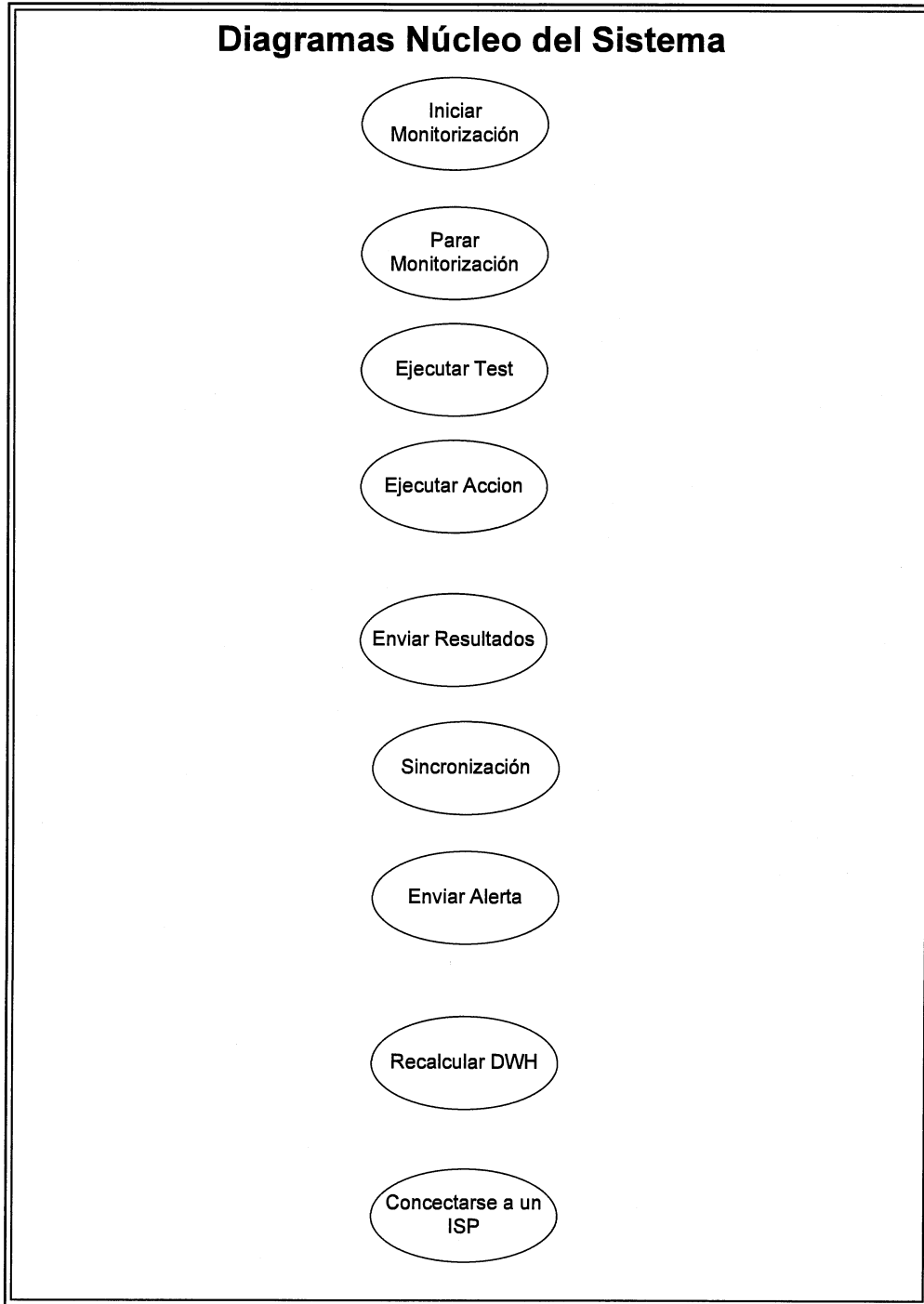
1. El usuario solicita que se pare la monitorización del agente.
2. El usuario comprueba que se para la navegación.

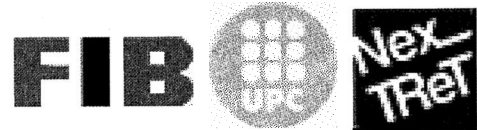
###### Respuesta del sistema:

1. El sistema cambia el estado del Robot en la base de datos mediante una petición al Web Service.
2. El sistema cierra la navegación que estuviera activa.

#### 4.1.4 Núcleo del sistema

##### 4.1.4.1 Diagrama de Casos de Uso





#### 4.1.4.2 Especificación de casos de uso

##### **Iniciar Monitorización**

**Actores:** El sistema

**Propósito:** Iniciar la monitorización de los tests programados en el robot que ejecuta el agente.

**Resumen:** El sistema inicia la ejecución de un robot, al comprobar que el estado es el correcto

##### **Sucesión típica de acontecimientos:**

Acciones del sistema:

1. El sistema comprueba el estado del robot.
2. El sistema inicia la ejecución del robot.

##### **Parar Monitorización**

**Actores:** El sistema

**Propósito:** Parar la monitorización de los tests programados en el robot que ejecuta el agente.

**Resumen:** El sistema para la ejecución de un robot, al comprobar que el estado es el correcto

##### **Sucesión típica de acontecimientos:**

Acciones del sistema:

1. El sistema comprueba el estado del robot.
2. El sistema espera hasta finalizar la monitorización del test.
3. El sistema para la ejecución del robot.

##### **Ejecutar Test**

**Actores:** El sistema

**Propósito:** Ejecutar un test

**Resumen:** El sistema va recorriendo el árbol de ejecución y al entrar en un test lo ejecuta.

##### **Sucesión típica de acontecimientos:**

Acciones del sistema:

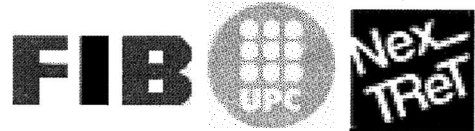
1. El sistema inicia la monitorización de un test.

##### **Ejecutar Acción**

**Actores:** El sistema

**Propósito:** Ejecutar una acción obre el IE de un test

**Resumen:** El sistema va recorriendo el árbol de ejecución y al entrar en una acción de un test la ejecuta.



### **Sucesión típica de acontecimientos:**

Acciones del sistema:

1. El sistema conoce la ejecución de una acción.
2. El sistema busca el elemento asociado a la acción dentro del IE.
3. El sistema ejecuta la acción dentro del IE

### **Conectarse a un ISP**

**Actores:** El sistema

**Propósito:** Conectarse a un ISP para realizar un Test.

**Resumen:** El sistema va recorriendo el árbol de ejecución y al entrar en un nuevo ISP necesita conectarse.

### **Sucesión típica de acontecimientos:**

Acciones del sistema:

1. El sistema conoce la ejecución de una prueba por un nuevo ISP.
2. El sistema busca dentro de RAS el ISP, en caso de ser una conexión telefónica a redes ejecuta la conexión. En caso de ser una conexión LAN continuar continua.
3. El sistema abre la ventana de conexión a un ISP y realiza la conexión.

### **Guardar Resultados en local**

**Actores:** El sistema

**Propósito:** Guardar un resultado en local.

**Resumen:** Una vez, se ha finalizado la carga de un script, se obtienen los resultados de la navegación en local, para su posterior envío al servidor central.

### **Sucesión típica de acontecimientos:**

Acciones del sistema:

1. El sistema obtiene los resultados de la monitorización.
2. El sistema graba los resultados en la base de datos local.

### **Enviar Resultados**

**Actores:** El sistema

**Propósito:** Enviar los resultados de la monitorización al servidor central.

**Resumen:** El sistema tiene planificado el envío de resultados al servidor central. El envío es periódico, y nunca coincide durante la ejecución de una prueba para no interferir en los tiempos de respuesta.

### **Sucesión típica de acontecimientos:**

Acciones del sistema:

1. El sistema lee los resultados de la base de datos local.
2. El sistema crear una petición al Web Service.
3. El sistema envía la petición.

4. El sistema almacena los datos en el servidor central

### **Sincronización**

**Actores:** El sistema

**Propósito:** Conocer el nuevo árbol de ejecución.

**Resumen:** El sistema tiene planificado la sincronización con el repositorio central, para conocer los nuevos tests a ejecutar o las modificaciones de los tests.

#### **Sucesión típica de acontecimientos:**

Acciones del sistema:

1. El sistema realiza una petición de sincronización al Web Service.
2. El sistema actualiza su árbol de ejecución.
3. El sistema continúa con la monitorización.

### **Enviar Alerta**

**Actores:** El sistema

**Propósito:** En caso de error enviar una alerta a una persona.

**Resumen:** El sistema en caso de encontrar un error en la monitorización de un test envía una alerta si el test tiene asociado una persona.

#### **Sucesión típica de acontecimientos:**

Acciones del sistema:

1. Se produce un error en la monitorización
2. El sistema comprueba que existen personas asociadas.
3. El sistema envía una alerta.

### **Recalcular DWH**

**Actores:** El sistema

**Propósito:** Proceso que recalcula el DWH.

**Resumen:** El sistema ejecuta un proceso que recalcula las estadísticas que se ven en el Reporting Web

#### **Sucesión típica de acontecimientos:**

Acciones del sistema:

1. El sistema recalcula el DWH