

**UNIVERSIDAD DE MURCIA**

**FACULTAD DE INFORMÁTICA**



**PROYECTO FIN DE CARRERA**

**DESARROLLO DE UNA APLICACIÓN DE  
GESTIÓN DE INCIDENCIAS E INVENTARIO:  
UNA IMPLEMENTACIÓN CON JSF, SPRING  
E HIBERNATE**

**Alumno:** M<sup>a</sup> Ángeles Murcia Almagro

**Director académico:** José Sáez Martínez

Junio 2009

# UNIVERSIDAD DE MURCIA

## PROYECTO FIN DE CARRERA

**TÍTULO:** “APLICACIÓN DE LAS TECNOLOGÍAS JSF, SPRING E HIBERNATE A LA GESTIÓN DE INCIDENCIAS E INVENTARIO.”

**AUTOR:** M<sup>a</sup> Ángeles Murcia Almagro

**TUTOR ACADÉMICO:** D. José Sáez Martínez.

**DEPARTAMENTO:** Informática y Sistemas.

**RESUMEN:** En este proyecto se realiza un estudio sobre el desarrollo de una aplicación de gestión de incidencias e inventario utilizando las tecnologías JSF, Spring e Hibernate.

Se muestra un análisis de la solución estudiando la metodología utilizada, la viabilidad y los requisitos del sistema.

Se explica el diseño de la aplicación mostrando alguna de las características más importantes como la integración de éstas tres tecnologías.

# ÍNDICE

# Índice

1. INTRODUCCIÓN .....	3
1.1. Sistemas de gestión de incidencias e inventario .....	3
1.2. Contexto .....	4
2 OBJETIVOS .....	7
3. ANÁLISIS DE LA SOLUCIÓN .....	10
3.1. Metodología .....	10
3.2. Estudio de viabilidad de la implantación de software .....	13
3.2.1. Alcance del sistema .....	13
3.2.2. Estudio de alternativas de la solución .....	14
3.2.3. Valoración de las alternativas .....	15
3.2.4. Selección de una solución .....	15
3.3. Planificación del proyecto .....	15
3.4. Análisis de requisitos .....	17
3.4.1. Requisitos funcionales Inventario de sistemas informáticos y telefónicos .....	17
3.4.2. Requisitos funcionales Gestión Web de incidencias de usuario relativas a informática, telefonía y mantenimiento general .....	17
3.4.3. Requisitos funcionales relativos a roles y grupos de usuarios .....	18
3.4.4. Requisitos funcionales Herramienta automática de detección de equipos IP .....	18
3.5. Diagrama de contexto .....	18
4. GESTIÓN DE INCIDENCIAS .....	21
4.1. Gestión de Usuarios .....	21
4.2. Gestión de Incidencias .....	25
4.3. Gestión de Incidencias del Usuario Básico .....	28
5. GESTIÓN DE INVENTARIO .....	31
6. DISEÑO E IMPLEMENTACIÓN .....	40
6.1. Patrones de Diseño .....	40
6.1.1. Abstract Factory .....	40
6.1.2. Data Access Object (DAO) .....	40
6.2. Integración de JSF + Spring + Hibernate .....	42
6.2.1. Capa de presentación .....	43
6.2.2. Capa de negocio .....	44
6.2.3. Capa de persistencia .....	44
6.3. Integración con AD .....	45
6.4. Implementación de demonios en Spring .....	47
6.5. Autogeneración de incidencias a partir de la lectura de correos electrónicos. ....	53
7. CONCLUSIONES Y VÍAS FUTURAS .....	57
8. ANEXO I .....	59
8.1. Funcionalidades actuales de Dumbo .....	59
8.1.1. Solicitud de nuevas tareas .....	59
8.1.2. Exploración de tareas solicitadas .....	59
8.1.3. Atención de tareas recibidas .....	60
8.2. Gestión global .....	62
8.2.1. Gestión de Dumbo por secciones .....	62
8.2.2. Gestión de Dumbo por aplicaciones .....	63
8.2.3. Gestión de Dumbo por informáticos .....	64
8.3. Gestión de la cola genérica de tareas .....	65

8.4. Gestión de la cola de problemas .....	66
8.5. Dumbo rápido .....	66
8.7. Administración del sistema.....	68
8.8. Zonas .....	68
8.9. Consultar/Modificar informáticos.....	70
8.10. Consultar/Modificar edificios.....	70
8.11. Formularios.....	71
8.12. Alias .....	72
8.13. Estadísticas .....	72
8.14. Alertas.....	72
8.15. Notificaciones .....	72
9. ANEXO II .....	74
9.1 Comparativa entre herramientas de inventario .....	74
9.2. Listado de herramientas .....	74
8.3. OCS Inventory NG .....	75
8.4. zCI Computer Inventory System .....	79
8.5. Comparativa entre OCS Inventory System y zCI Computer Inventory System .....	80
10. BIBLIOGRAFÍA.....	82

## **ÍNDICE DE FIGURAS**

# Índice de Figuras

Figura 1. Modelo en cascada.....	11
Figura 2. Planificación de las fases del proyecto.....	16
Figura 3. Diagrama de contexto .....	19
Figura 4. Diagrama jerarquía de Grupos Técnicos.....	21
Figura 5. Pantalla Formulario Grupos Técnicos. ....	22
Figura 6. Pantalla de búsqueda de Usuarios. ....	23
Figura 7. Pantalla resultados búsqueda.....	23
Figura 8. Pantalla Detalle Usuario Pestaña Estaciones de trabajo.....	24
Figura 9. Pantalla Detalle Usuario Pestaña Grupos Usuarios.....	24
Figura 10 Diagrama Jerarquía Tipos de Incidencias.....	25
Figura 11. Pantalla Tipos de Incidencias .....	26
Figura 12. Pantalla búsqueda Incidencias.....	27
Figura 13. Pantalla Detalle Incidencias .....	28
Figura 14. Pantalla Creación nueva incidencia.....	28
Figura 15. Pantalla búsqueda de incidencias .....	29
Figura 16. Pantalla resultados incidencias .....	29
Figura 17. Diagrama Jerarquía elementos de inventario .....	32
Figura 18. Pantalla Búsqueda de Inventarios Software. ....	33
Figura 19. Pantalla Detalle Inventario Software.....	33
Figura 20. Pantalla Inventario Puntos de Red. ....	34
Figura 21. Pantalla Inventario de Equipos.....	34
Figura 22. Pantalla Inventario de Ordenador.....	35
Figura 23. Pantalla Terminales. ....	36
Figura 24. Pantalla Tarjetas SIM.....	37
Figura 25. Pantalla Extensiones.....	38
Figura 26. Diagrama de clases del patrón DAO.....	41
Figura 27. Código de la interfaz IncidenciaDAO .....	42
Figura 28. Diagrama que ilustra la estructura MVC.....	42
Figura 29. Arquitectura multicapa [11]. ....	43
Figura 30. Operación de negocio en una arquitectura con Spring .....	44
Figura 31. Ejemplo de anotaciones. ....	45
Figura 32. Código implementación conexión Directorio Activo.....	45
Figura 33. Código que muestra la recuperación de las unidades.....	46
Figura 34. Código de la configuración de los demonios en el fichero applicationContext.xml. ....	48
Figura 35. Código demonio comprobación de tareas. ....	52
Figura 36. Código conexión buzón correo.....	54
Figura 37. Código clase que lee correo electrónico.....	55
Figura 38. Solicitud de tarea.....	60
Figura 39. Seguimiento de tareas solicitadas. ....	60
Figura 40. Listado de tareas pendientes.....	61
Figura 41. Detalle tarea. ....	62
Figura 42. Acciones sobre una Tarea. ....	62
Figura 43. Gestión de sección. ....	63
Figura 44. Tareas vinculadas a una sección.....	63
Figura 45. Gestión por Aplicación. ....	64
Figura 46. Tareas vinculadas a una aplicación. ....	64
Figura 47. Gestión por informáticos.....	65
Figura 48. Tareas vinculadas a un informático. ....	65

Figura 49. Gestión de cola genérica.....	66
Figura 50. Gestión de cola de problemas.....	66
Figura 51. Dumbo rápido.....	67
Figura 52. Petición Programada. ....	67
Figura 53. Listado de sistemas.....	68
Figura 54. Nuevo sistema.....	68
Figura 55. Listado de zonas. ....	69
Figura 56. Nueva zona. ....	69
Figura 57. Consultar/ Modificar Informáticos. ....	70
Figura 58. Consultar/ Modificar edificios.....	70
Figura 59. Formularios. ....	71
Figura 60. Alias.....	72
Figura 61. Arquitectura OCS Inventory NG.....	76
Figura 62. Elementos inventariables por OCS.....	78
Figura 63. Elementos inventariables por zCI Computer Inventory System.....	79



## **1.- INTRODUCCIÓN**

# 1. INTRODUCCIÓN

## 1.1. Sistemas de gestión de incidencias e inventario

Hasta hace poco tiempo, la práctica habitual en cualquier organización, cuando ocurría una incidencia, era llamar por teléfono al servicio de informática correspondiente para notificarlo.

Por otra parte, los inventarios tecnológicos o bien eran recogidos en papel o bien existían distintas bases de datos descentralizadas e inconexas.

Por lo tanto parece lógica la necesidad de un software que centralice toda esta información y al mismo tiempo facilite el trabajo a los distintos servicios de informática.

La informatización de los sistemas gestores de incidencias e inventario es un campo en auge en los sistemas de información de muchas organizaciones en la actualidad y son evidentes algunas de las ventajas:

- Resolución de incidencias de forma rápida y ordenada.
- Registro exacto de los datos referentes a inventario.

En un principio la gestión de las incidencias vía web podía plantear ciertas dudas, desde el punto de vista del usuario final, que está acostumbrado a llamar por teléfono para comunicar su incidencia. Ahora deberá acceder vía web a la aplicación e introducir los datos que se requieren para dar de alta de incidencia.

Por lo tanto, al implantar un sistema gestor de estas características si queremos obtener el éxito se debe de realizar una tarea docente con el personal de la organización que lo utilizará.

## 1.2. Contexto

Para comprender el contexto de este proyecto es necesario hablar sobre Metaenlace.

MetaEnlace (<http://www.metaenlace.com>) es una empresa de Servicios Integrales para Sistemas de Información, que comprenden desde consultoría tecnológica hasta ingeniería e integración de sistemas. Constituida en 2001, ha conseguido excelentes resultados en diversas vertientes, destacando especialmente los Sistemas de Información para Internet e Intranet.

Esta empresa trabaja en muchos casos para la administración pública y ha observado la necesidad que tienen muchas instituciones de una herramienta web que integre la gestión de incidencias, inventario y telefonía.

Los Sistemas de Información de estas instituciones necesitan encargarse de gestionar toda la información relativa a:

- Incidencias generadas, resueltas.
- Inventario existente.
- Telefonía: terminales, extensiones, tarjetas SIM.

Toda esta información será tratada por el Servicio de Informática.

Para la gestión del Sistema de Información se trabaja en base a incidencias.

Una incidencia es la notificación de cualquier anomalía (rotura, mal funcionamiento) o la petición de una tarea determinada:

Nuevos procesos/desarrollos o modificación/actualización de los ya existentes.

Este proyecto surge tras la necesidad que se le presenta a Metaenlace de construir una herramienta software que gestione las incidencias que se generan en los Sistemas de Información de las instituciones con las que trabaja.

Además se ha aprovechado, para incluir una gestión del inventario que facilitará la integración en una sola herramienta de múltiples gestiones.

Como primer paso de este proyecto, se hizo un estudio del software ya existente para implantarlo directamente, pero debido a ciertas características específicas que se querían incluir se optó por la creación de un nuevo sistema gestor de incidencias e inventario.

Una de estas características específicas es la gestión de Usuarios desde un sistema de Directorio Activo que está implantando en la organización destinataria de la herramienta a construir. En él, los usuarios se distribuyen

en las distintas unidades o divisiones de las que pueden formar parte, cada usuario tendrá asignados unos determinados grupos que marcan permisos de acceso a la propia aplicación o a otras aplicaciones que posea la institución.

Otra característica específica será la visibilidad del usuario de ciertas partes de la aplicación debido a la creación de unos “grupos técnicos” que limitan la visibilidad atendiendo al grupo al que se pertenezca (desarrolladores, CAU, telefonía, administrador).

## **2.- OBJETIVOS**

## 2 OBJETIVOS

Este proyecto fin de carrera tiene como objetivos principales:

1. Análisis de necesidades y diseño de un sistema de gestión de incidencias.
2. Análisis y diseño de un gestor de inventario.
3. Implementación de una herramienta integrada para gestión de incidencias + inventario.
4. Estudio y conocimiento detallado y uso de las tecnologías JSF + Spring + Hibernate.

El primer objetivo se plantea fácil en principio, ya que este tipo de aplicaciones se han hecho muy populares en los últimos años y los requisitos están bastante bien definidos. Todas ellas suelen contemplar una gestión de usuarios, de equipos, etc.

En nuestro caso se nos han planteado unas características específicas que han hecho necesaria esta nueva implementación. Entre dichas características se encuentran la gestión de usuarios con Directorio Activo, incluir una determinada jerarquía de tipos de incidencias, información adicional sobre incidencias de telefonía. Se ha partido de un estudio previo valorando distintos gestores de incidencias existentes, entre ellos Dumbo [1], utilizado por la universidad de Murcia.

En primer lugar, se realizó un estudio sobre diversos gestores de inventario existentes, eligiendo finalmente OCS para integrarlo con esta herramienta. Las herramientas consideradas fueron:

- Open Computers and Software (OCS) Inventory NG [2].
- zCI Computer Inventory System [3].
- Windows Inventory/Open-AudIT [4].

- h-inventory [5].

De entre las herramientas anteriores se preseleccionaron las dos primeras por tratarse de software libre, herramientas con cierta madurez y no abandonadas.

El cuarto objetivo se concreta en la implementación de la herramienta, donde se hacen uso de estas tres tecnologías.

### **3.- ANÁLISIS DE LA SOLUCIÓN**



### **3. ANÁLISIS DE LA SOLUCIÓN**

Se va a realizar un estudio de cómo va a ser la solución adoptada para este proyecto. En primer lugar, explicaremos la metodología aplicada durante el proceso de desarrollo, estudio de viabilidad, análisis de requisitos y de la planificación de este proyecto.

#### **3.1. Metodología**

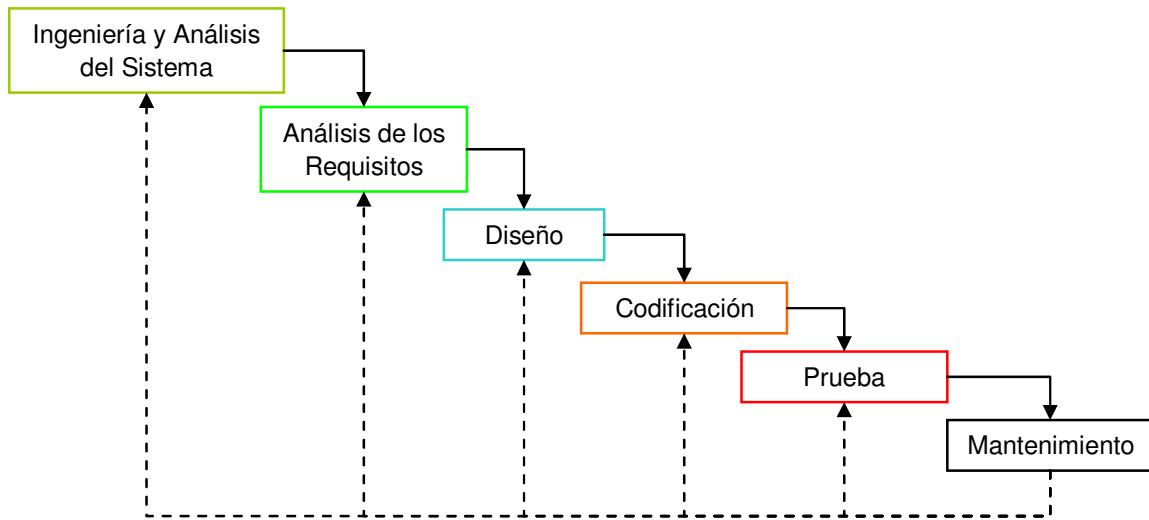
En la realización de este trabajo se ha seguido un proceso iterativo basado en el enfoque metodológico clásico denominado ciclo de vida en cascada [6].

Dicho enfoque consiste en ordenar las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior. Tras finalizar cada una de las fases se puede pasar a la siguiente o reutilizar su salida como realimentación para refinar el trabajo realizado en alguna de las etapas anteriores.

Se ha seguido el modelo en cascada pero no de una manera rígida, sino que el proceso ha sido iterativo. Es decir, no se ha necesitado esperar la finalización completa de una tarea para pasar a la siguiente.

Se ha realizado una implementación parcial de cada una de las etapas y se ha vuelto a empezar por el principio del ciclo, depurando y testando, reajustando de esta forma el proceso a nuestro objetivo final.

En la figura 1 se muestran de forma esquematizada las principales etapas del modelo en cascada, incluyendo las fases de realimentación mencionadas en forma de líneas discontinuas.



**Figura 1.** Modelo en cascada

Las principales características de cada una de las etapas mostradas en la figura 1 son las siguientes:

**Ingeniería y Análisis del Sistema:** debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

En nuestro proyecto esta fase consistió en comprender la estructura y funciones de la organización destinataria del software y cómo el Servicio de Informática se enmarca dentro de ella.

**Análisis de los requisitos del software:** el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software.

En nuestro caso, se nos proporcionó un documento inicial de análisis, en el que se especificaban los requisitos software y hardware que precisaría la aplicación. En este documento se detalla en profundidad los formularios que conformarán la aplicación, así como los campos que se encuentran en cada uno de ellos. Así mismo, se identificaban las características particulares mencionadas anteriormente: integración con el Directorio Activo, definición de la jerarquía de incidencias a considerar y la información adicional que incorpora una incidencia de telefonía frente a las incidencias más “clásicas”.

**Diseño:** el diseño del software se enfoca en cuatro atributos distintos del programa:

- la estructura de los datos,
- la arquitectura del software,
- el detalle procedimental
- y la caracterización de la interfaz.

El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

En este trabajo hemos realizado un diagrama de clases que especifica cuál va a ser el diseño de la estructura de datos. En posteriores iteraciones hemos obtenido a partir de aquí el esquema E/R de los datos, de manera que se pudiera traducir directamente a una BD relacional.

La arquitectura del software está definida en tres niveles:

- Capa de presentación, se usa la versión que MyFaces hace de JSF (Java Server Faces) para desarrollar las vistas de la aplicación [7].
- Capa de negocio, nos apoyamos en el framework Spring [8].
- Capa de persistencia de datos, se desarrolla utilizando el framework de mapeo objeto-relacional Hibernate [9]. En cualquier caso el acceso a los datos se implementa mediante el patrón DAO

La interconexión entre los distintos subsistemas se realiza mediante servicios web (subsistema de autenticación de usuarios, autorización de usuarios). La gestión de los Usuarios se realiza desde el Directorio Activo.

En cuanto a la caracterización de la interfaz, en nuestro caso partimos del documento de análisis proporcionado inicialmente. Este documento sirvió de guía para la creación inicial de los formularios, que hubo que modificar posteriormente.

En la primera iteración, se implementaron tal cual se detallaban en el documento de análisis. En posteriores iteraciones estos formularios fueron cambiando a medida que se refinaron los requisitos.

**Codificación:** consiste en traducir de una forma legible para la máquina la etapa anterior (diseño). Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

En este apartado, hemos llevado a cabo la implementación del esquema E/R en un SGBD relacional, SQL Server y se ha desarrollado el código en el entorno de desarrollo Eclipse Ganymede [14].

**Prueba:** una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren. En el proyecto que nos ocupa, la fase de pruebas se realizó al terminar cada uno de los módulos: incidencias, inventario, telefonía.

**Mantenimiento:** el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que se hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

En este proyecto, el mantenimiento se llevará a cabo una vez que el software se implante en la organización a la que esté destinado. Habrá

reuniones semanales en las que el jefe del servicio de informática o similar informará de los cambios/adaptaciones que precisa la aplicación.

Finalmente, hemos desarrollado documentación siguiendo la estructura definida por Métrica Versión 3 [10].

La principal ventaja de la utilización de la utilización del modelo en cascada radica en su sencillez, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.

Y como principales desventajas se podrían exponer las dos siguientes:

- Normalmente, es difícil para el cliente establecer explícitamente al principio todos los requisitos. En teoría, el ciclo de vida clásico lo requiere y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos productos. Por eso, nuestro proceso incorpora el elemento iterativo, para permitir redefinir y reajustar los requisitos y necesidades conforme se va desarrollando el proceso.
- El cliente debe tener paciencia hasta llegar a las etapas finales del proyecto, ya que hasta ese momento no estará disponible una versión operativa del programa. Un error importante no detectado hasta que el programa esté funcionando puede ser desastroso. Para intentar paliar este inconveniente en este trabajo se han ido haciendo entregas semi-funcionales, incluyendo gradualmente partes de la funcionalidad. También hemos intentado acortar en lo posible la duración de los ciclos, de manera que se pudieran probar versiones prototípicas de las aplicaciones cuanto antes.

## **3.2 .Estudio de viabilidad de la implantación de software**

En este apartado vamos a realizar un estudio de qué alternativa es mejor para la empresa, si desarrollar el proyecto entero desde un principio o adquirir un gestor de incidencias ya implementado e intentar integrarlo con nuestra aplicación web.

Elegir una solución u otra puede ahorrar mucho trabajo y tal vez el éxito del proyecto dependa de ello.

### **3.2.1. Alcance del sistema**

El software debe permitir la gestión de incidencias e inventario. La aplicación deberá poder integrarse con un servicio de Directorio Activo para la gestión de usuarios y con una herramienta gestora de inventario.

### 3.2.2. Estudio de alternativas de la solución

#### **Alternativa A. Compra de gestor de incidencias ya desarrollado, por ejemplo Mantis.**

##### VENTAJAS

- No será necesario emplear tanto tiempo en implementar la solución software.
- El esfuerzo se centrará en dominar la solución software y la documentación.

##### INCONVENIENTES

- Puede ser que no dispongamos del código fuente de la aplicación.
- Existirá una dependencia de la empresa a la que se ha comprado el software.
- Hay que realizar un desembolso económico que encarecerá el precio total del producto.
- No se podrán realizar cambios en el software ya que no se conoce como está construido el software.

#### **Alternativa B. Utilización de gestor de incidencias ya implementado y libre: Dumbo.**

##### VENTAJAS

- Disponemos del código fuente de la aplicación.
- No será necesario emplear tanto tiempo en implementar la solución software.
- El esfuerzo se centrará en dominar la solución software y la documentación.
- Al ser software libre se pueden realizar las modificaciones que se consideren convenientes.

##### INCONVENIENTES

- Cualquier modificación que se precise será costosa ya que habrá que realizar un estudio de la aplicación y la documentación existente.
- Habrá módulos de la aplicación que no nos resulten necesarios.

#### **Alternativa C. Desarrollo propio en Metaenlace.**

##### VENTAJAS

- Se dispondrá del conocimiento necesario para realizar un cambio en el software cuando sea necesario.
- Metaenlace podrá ofrecer este producto a otras empresas u organismos públicos.

- Este código se podrá particularizar dependiendo del destino.

#### INCONVENIENTES

- Será necesario emplear más tiempo en implementar la solución software.
- Habrá que invertir recursos en el desarrollo del nuevo producto.

#### **3.2.3. Valoración de las alternativas**

Desde un principio la alternativa A fue la menos valorada, ya que exigía más recursos económicos y era una opción que limitaba la extensibilidad de la aplicación en un futuro. Por lo tanto fue descartada.

La alternativa B fue largamente valorada y estudiada. En el Anexo I se realizó un estudio detallado de la aplicación, viendo la funcionalidad de la aplicación y sus distintas pantallas.

La aplicación tiene muchas opciones que no van a ser necesarias aunque también tiene la ventaja que ha sido durante mucho tiempo probada y utilizada.

Si lo que buscamos es una opción a medida para introducir la funcionalidad necesaria e integrar de una manera limpia la gestión de incidencias con el inventario la opción más adecuada es la alternativa C.

#### **3.2.4. Selección de una solución**

Finalmente se optó por la alternativa C, ya que Dumbo tiene demasiados módulos y opciones que no se precisan. Además para adaptarla a nuestras necesidades habría que modificar mucho código, lo cual requeriría mucho tiempo.

Con lo cual dado que la empresa puede afrontar los recursos que se precisan para implementar la aplicación desde un principio, se optó por la alternativa C.

### **3.3. Planificación del proyecto**

En la figura 2 se muestra una planificación aproximada de las fases en las que se ha dividido este proyecto y en qué fechas se ha ido afrontando cada una de las partes del mismo.

Como ya se explicó anteriormente este proceso sigue un ciclo iterativo, por lo que las etapas se han ido reajustando en cada una de las iteraciones. Por lo que las fechas anteriores son orientativas, ya que al realizar la fase de pruebas hubo que modificar codificación y así sucesivamente.

<b>FASE</b>	<b>OBJETIVOS</b>	<b>COMIENZO</b>	<b>FIN</b>	<b>DURACIÓN</b>
<b>Ingeniería y Análisis del Sistema</b>	Estudio de la aplicación de incidencias Dumbo	15/09/2008	19/09/2008	1 semana
<b>Ingeniería y Análisis del Sistema</b>	Estudio de los distintos gestores de inventario.	15/09/2008	19/09/2008	1 semana
<b>Ingeniería y Análisis del Sistema</b>	Reuniones con mi jefe de proyecto en la empresa para enmarcar el proyecto y delimitar la funcionalidad.	22/09/2008	26/09/2008	1 semana
<b>Análisis de los requisitos del software</b>	Refinamiento del documento de requisitos inicial.	29/09/2008	03/10/2008	1 semana
<b>Diseño</b>	Realización del modelo E/R	06/10/2008	10/10/2008	1 semana
<b>Diseño</b>	Elección de la BD	06/10/2008	10/10/2008	1 semana
<b>Diseño</b>	Elección de la arquitectura sw.	13/10/2008	17/10/2008	1 semana
<b>Diseño</b>	Estudio sobre si evolucionar a Hibernate con anotaciones y a JSF 1.2.	20/10/2008	24/10/2008	1 semana
<b>Diseño</b>	Diseño del esqueleto básico de la aplicación.	20/10/2008	24/10/2008	1 semana
<b>Codificación</b>	Creación de BD y tablas del modelo	27/10/2008	31/10/2008	1 semana
<b>Codificación</b>	Implementación de formularios	03/11/2008	02/02/2009	3 meses
<b>Pruebas</b>	Probar funcionalidad de todos los formularios	03/02/2009	01/06/2009	4 meses
<b>Mantenimiento</b>	Solucionar posibles errores y ampliar funcionalidad.	03/02/2009	01/06/2009	4 meses

**Figura 2.** Planificación de las fases del proyecto.

### **3.4. Análisis de requisitos**

Se nos proporcionó un documento de requisitos al comenzar el proyecto, que tras la sucesión de las distintas iteraciones se fue refinando hasta obtener el documento final. A continuación se presentan los requisitos del software.

#### **3.4.1. Requisitos funcionales Inventario de sistemas informáticos y telefónicos**

RF1. Deberá poder inventariar ordenadores, impresoras, monitores, escáneres, equipos de comunicaciones, software, teléfonos móviles y fijos, líneas telefónicas, extensiones, tarjetas SIM, cajas de telefonía, etc.

RF2. El sistema debe incluir un control de caducidad de licencias y mantenimientos hardware.

RF3. Se realizará una evaluación de las distintas herramientas de inventario automático para determinar cuál de ellas es la más adecuada a los objetivos de este desarrollo.

RF4. Gestión de stocks de informática y mantenimiento.

RF5. Se ejecutará diariamente un demonio, que conectándose a la herramienta de inventario automática, actualizará los datos del ordenador así como el software que tiene instalado.

RF6. Los objetos de stock y los cartuchos de impresora tendrán una cantidad mínima de existencias. Cuando el stock de un producto esté en existencias mínimas se genera una incidencia automática avisando de que es necesaria la compra de más material.

RF7. Se podrá visualizar para cada terminal, extensión y tarjeta SIM los usuarios que las han utilizado, así como la fecha de alta y de baja.

#### **3.4.2. Requisitos funcionales Gestión Web de incidencias de usuario relativas a informática, telefonía y mantenimiento general**

RF8. El sistema deberá de ser capaz de integrarse con un sistema automático de inventario de todos aquellos elementos que sea posible y con el directorio activo de Windows.

RF9. Permitirá a los usuarios el alta de incidencias a través del entorno web, y la asignación automática de cada tipo de incidencia a los distintos responsables.

RF10. Consultar las incidencias resueltas o pendientes por parte de los usuarios.

RF11. Los usuarios podrán rechazar la resolución de la incidencia en caso de no estar conformes.

RF12. Los responsables de cada tipo de incidencia podrán realizar un seguimiento de las tareas resueltas por su equipo de trabajo.



RF13. Integrar las incidencias que se sigan realizando vía telefónica.

RF14. Integrar las incidencias que el equipo informático realice a sus suministradores.

RF15. Integrar las tareas periódicas y planificadas del equipo informático.

RF16. Documentar la resolución de las incidencias más complejas, permitiendo incluir los documentos elaborados.

### **3.4.3. Requisitos funcionales relativos a roles y grupos de usuarios.**

RF17. Los roles de los usuarios determinan la visibilidad de las distintas opciones del menú.

RF18. El usuario con rol 'Administrador' tendrá acceso total.

RF19. Cada tipo de incidencia tiene asociado un grupo técnico resolvedor.

RF20. El usuario que pertenece a un grupo técnico sólo visualiza las incidencias del tipo asociado a su grupo.

### **3.4.4. Requisitos funcionales Herramienta automática de detección de equipos IP**

RF21. Será necesario realizar un estudio comparativo de las herramientas disponibles en el mercado. Se le dará preferencia a aquellas de libre distribución

RF22. Permitirá la detección automática de equipos que se puedan identificar a través de IP. Para ello, realizará un rastreo automático de los mismos.

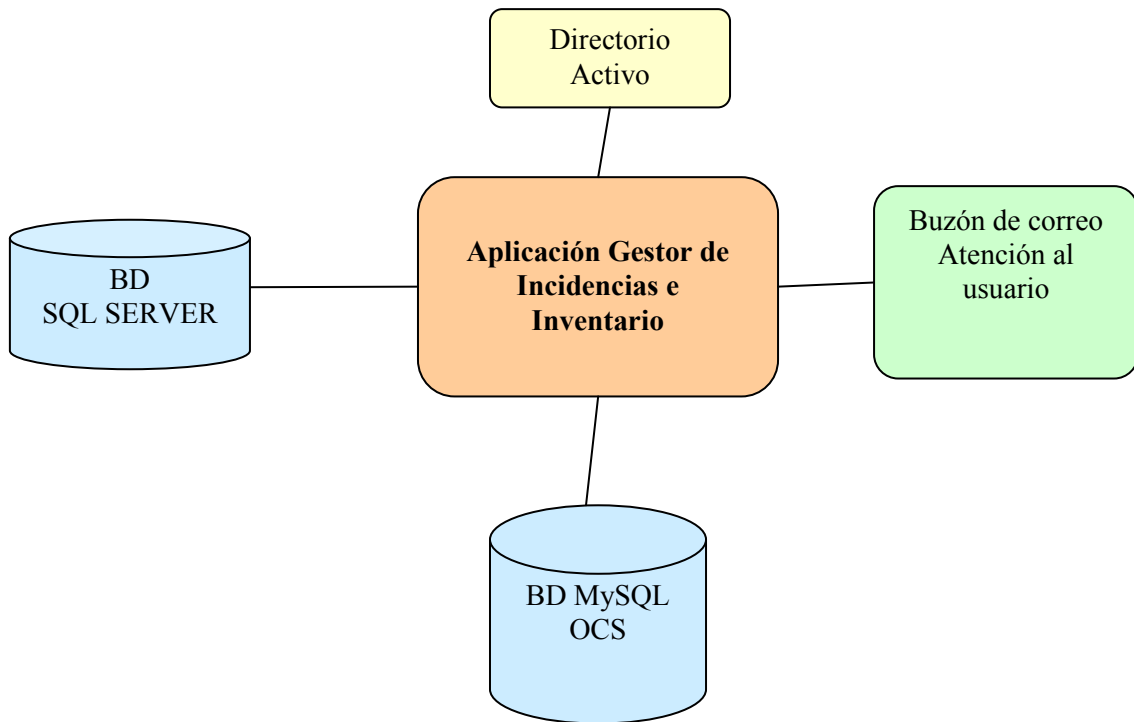
RF23. Los datos de los equipos IP que contabilice la herramienta deberán de estar disponibles a través de un interfaz de comunicación al que se pueda acceder mediante programación.

## **3.5. Diagrama de contexto**

El diagrama de contexto de la figura 3 muestra la conexión del sistema con dos bases de datos y la integración que debe existir con el Directorio Activo y con el buzón de correo.

Esta integración será transparente al usuario que únicamente tendrá que conectarse a la aplicación web y tendrá todos los datos disponibles.

En el capítulo 6, Diseño e Implementación, se explica detalladamente como se realiza la integración con el Directorio Activo y con el buzón de correo.



**Figura 3.** Diagrama de contexto

## **4.- GESTIÓN DE INCIDENCIAS**

## 4. GESTIÓN DE INCIDENCIAS

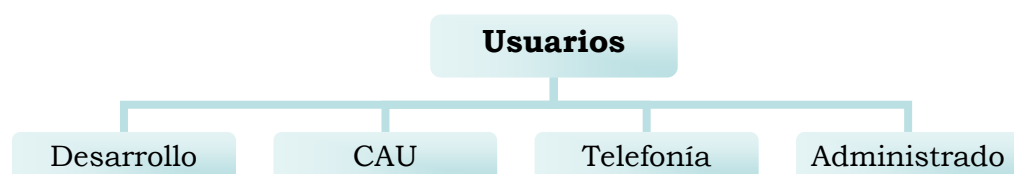
A continuación vamos a analizar en detalle las funcionalidades más importantes de la aplicación de Gestión de Incidencias. Esta funcionalidad se puede dividir en dos aspectos fundamentales:

- Gestión de Usuarios
- Gestión de Incidencias

### 4.1. Gestión de Usuarios

Como ya hemos dicho, la gestión de usuarios ha de estar integrada con el Directorio Activo existente en la organización. Los usuarios que gestiona la aplicación se clasifican en cuatro grupos:

- Desarrollo, a este grupo pertenecerán los informáticos que se encargan de desarrollar/mantener aplicaciones.
- CAU, personal perteneciente al servicio de informática que se encargan del mantenimiento informático y del inventario de equipos.
- Telefonía, personal encargado de gestionar la telefonía.
- Administrador, persona con permisos para acceder a todos los formularios.



**Figura 4.** Diagrama jerarquía de Grupos Técnicos

- **Grupos Técnicos**

Este formulario gestionará el tipo de incidencias que va a ver cada usuario atendiendo al grupo técnico al que pertenezca.

Es decir, en el formulario de incidencias se detecta el usuario que ha entrado a la aplicación y podrá ver sólo incidencias de un tipo determinado.

En la figura 5 se muestra la pantalla de grupos técnicos, a cada grupo técnico le podemos asignar los usuarios que pertenecerán a él.

Grupos de Técnicos

Búsqueda y selección:  
Nombre:

Restablecer | Buscar

	Nombre	Descripción
	Aplicación Gestión Incidencias e Inventario	Desarrolladores de la aplicación de gestión de incidencias e inventario.
	CAU	Personal CAU
	Desarrollo Secretaría	Desarrolladores Secretaría General (Plataforma CHS, Justo, Nedaes, etc.)
	GIS	Sistema de Información Geográfica
	Telefonía	Personal Telefonía

Añadir

Usuarios

Cuenta dominio ↑	Nombre
DESARROLLO	A. T. Informática desarrollo (Jose Antonio)
DESARROLLO1	A.T. de informática para desarrollo 1

**Figura 5.** Pantalla Formulario Grupos Técnicos.

- **Usuarios Pantalla Búsqueda/Resultados**

La herramienta de gestión de usuarios permite realizar búsquedas y resultados incluyendo información existente en la BD del Directorio Activo. La figura 6 muestra el formulario que permite realizar una búsqueda de usuarios en base a ciertas características del mismo.

En la figura 7 se muestran los resultados en forma de tabla paginada.

Usuarios | Incidencias | Telefonía | Informática | Stock | Contratos | Formularios Auxiliares | Usuario: desarrollo1

**Usuarios**

Búsqueda y Selección

Cuenta Dominio:   
 Nombre:   
 Nombre a Mostrar:   
 Email:   
 Dependencia: ...  
 Despacho: ...  
 Grupo: ...  
 F.Baja desde/hasta:  -   
 Mail público:  Sí  No  Indiferente

Apellidos:   
 NIF:   
 Unidad: ...  
 Planta: ...  
 Nombre A.T.: ...  
 F.Alta desde/hasta:  -   
 Inicio de Sesión: No seleccionado  
 Externo CHS:  Sí  No  Indiferente

Restablecer Buscar

**Figura 6.** Pantalla de búsqueda de Usuarios.

Usuarios | Incidencias | Telefonía | Informática | Stock | Contratos | Formularios Auxiliares | Usuario: desarrollo1

**Usuarios**

Búsqueda y Selección

Encontrados 909 resultados.  
 Mostrando del 191 al 200

16 17 18 19 20 21 22 23 24

**Usuarios**

	Cuenta Dominio	NIF	Nombre	Email	Externo a CHS	Email público intranet	Fecha Alta	Fecha Baja
●	CABAÑAS		CABAÑAS		<input type="checkbox"/>	<input type="checkbox"/>	01/02/2009 00:00:00	02/01/2009
●	CALATAYUD		CALATAYUD		<input type="checkbox"/>	<input type="checkbox"/>	01/01/2009 00:00:00	02/01/2009
●	CALATAYUD JOSÉ		CALATAYUD JOSÉ		<input type="checkbox"/>	<input type="checkbox"/>	01/01/2009 00:00:00	02/01/2009
●	CAMARILLAS		CAMARILLAS		<input type="checkbox"/>	<input type="checkbox"/>	01/01/2009 00:00:00	02/01/2009
●	CAMPUZANO		López Campuzano, Manuel Ángel	manuel.lopez@chs.mma.es	<input type="checkbox"/>	<input type="checkbox"/>	18/07/2008 07:32:54	
●	CANALCREVILLENTE		CANALCREVILLENTE		<input type="checkbox"/>	<input type="checkbox"/>	01/01/2009 00:00:00	02/01/2009
●	CANDEL		CANDEL		<input type="checkbox"/>	<input type="checkbox"/>	01/01/2009 00:00:00	02/01/2009
●	CANO AGUSTIN		CANO AGUSTIN		<input type="checkbox"/>	<input type="checkbox"/>	01/01/2009 00:00:00	02/01/2009
●	CANTÓ		CANTÓ		<input type="checkbox"/>	<input type="checkbox"/>	01/01/2009 00:00:00	02/01/2009
●	CAPARROS		CAPARROS		<input type="checkbox"/>	<input type="checkbox"/>	01/01/2009 00:00:00	02/01/2009

Añadir

**Figura 7.** Pantalla resultados búsqueda

- **Usuarios Pantalla Detalle**

Cuando seleccionamos un usuario de la tabla de resultados la aplicación nos lleva a otra pantalla, en la que vemos las características particulares y específicas de cada usuario.

Como se puede observar en las figuras 8 y 9, desde este formulario además de poder visualizar las características simples del usuario, se pueden gestionar muchas de sus propiedades.

Si es un Usuario procedente del Directorio Activo podremos ver:

- Estaciones de trabajo en las que este usuario tiene permiso para iniciar sesión.
- Grupos a los que pertenece, esto le dará permisos acceso en la organización.

Usuarios | Incidencias | Telefonía | Informática | Stock | Contratos | Formularios Auxiliares | Usuario: desarrollo1

**Usuarios**

**Detalles del usuario:**

Cuenta Dominio: CAMPUZANO NIF: Apellidos: Nombre: López Campuzano, Manuel Ánge  
 Externo CHS: [ ] Apellidos: Nombre M.: López Campuzano, Manuel Ánge  
 Dependencia: [ ] Planta: [ ] Despacho: [ ]  
 Fecha Alta: 18/07/200 Unidad: COMISARIA  
 Fecha Baja: [ ] Email público: [ ] Usuario AD: [x] Email: manuel.lopez@chs.mma.es

Baja Usuario Guardar

Iniciar Sesión Grupos Terminal Tarjetas SIM Extensiones Incidencias Equipos Ordenadores Teléfonos Directos Anotaciones

**Estaciones de Trabajo**

Nombre	Sistema Operativo	Service Pack
CAMPUZANO	Windows XP Professional	Service Pack 3

**Figura 8.** Pantalla Detalle Usuario Pestaña Estaciones de trabajo

Usuarios | Incidencias | Telefonía | Informática | Stock | Contratos | Formularios Auxiliares | Usuario: desarrollo1

**Usuarios**

**Detalles del usuario:**

Cuenta Dominio: CAMPUZANO NIF: Apellidos: Nombre: López Campuzano, Manuel Ánge  
 Externo CHS: [ ] Apellidos: Nombre M.: López Campuzano, Manuel Ánge  
 Dependencia: [ ] Planta: [ ] Despacho: [ ]  
 Fecha Alta: 18/07/200 Unidad: COMISARIA  
 Fecha Baja: [ ] Email público: [ ] Usuario AD: [x] Email: manuel.lopez@chs.mma.es

Baja Usuario Guardar

Iniciar Sesión Grupos Terminal Tarjetas SIM Extensiones Incidencias Equipos Ordenadores Teléfonos Directos Anotaciones

**Grupos de Usuarios**

Nombre	Descripción
GASUPERFICIALES	Grupo de usuarios de aguas superficiales
GCOMISARIA	Grupo de Comisaría
Usuarios de CHS	
Usuarios de Comisaría de Aguas	
Usuarios de Mahonesas	Usuarios de Mahonesas

**Figura 9.** Pantalla Detalle Usuario Pestaña Grupos Usuarios.

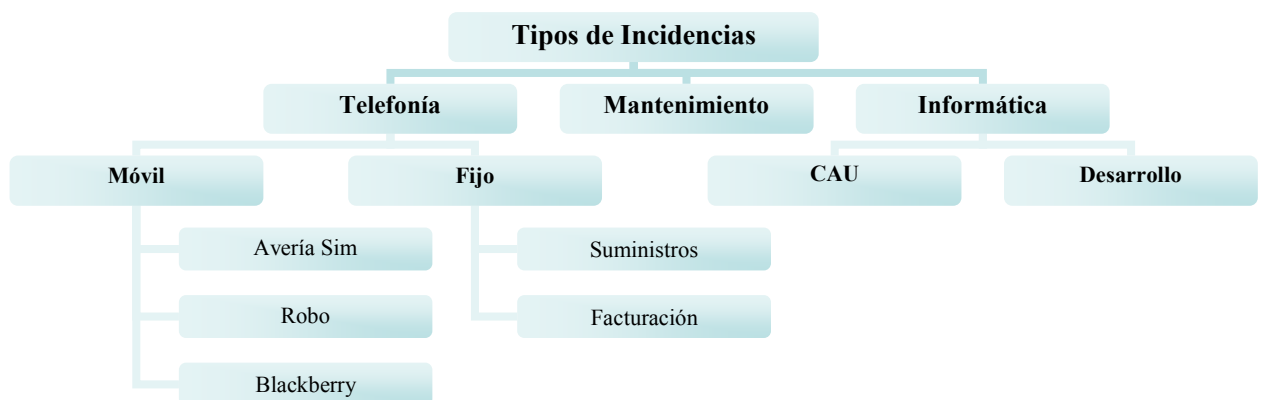
Se pueden visualizar los datos referentes a telefonía: los terminales, tarjetas SIM y extensiones de las que dispone el usuario.

Además se muestran las incidencias introducidas por el usuario y los ordenadores y equipos de los que dispone.

## 4.2. Gestión de Incidencias

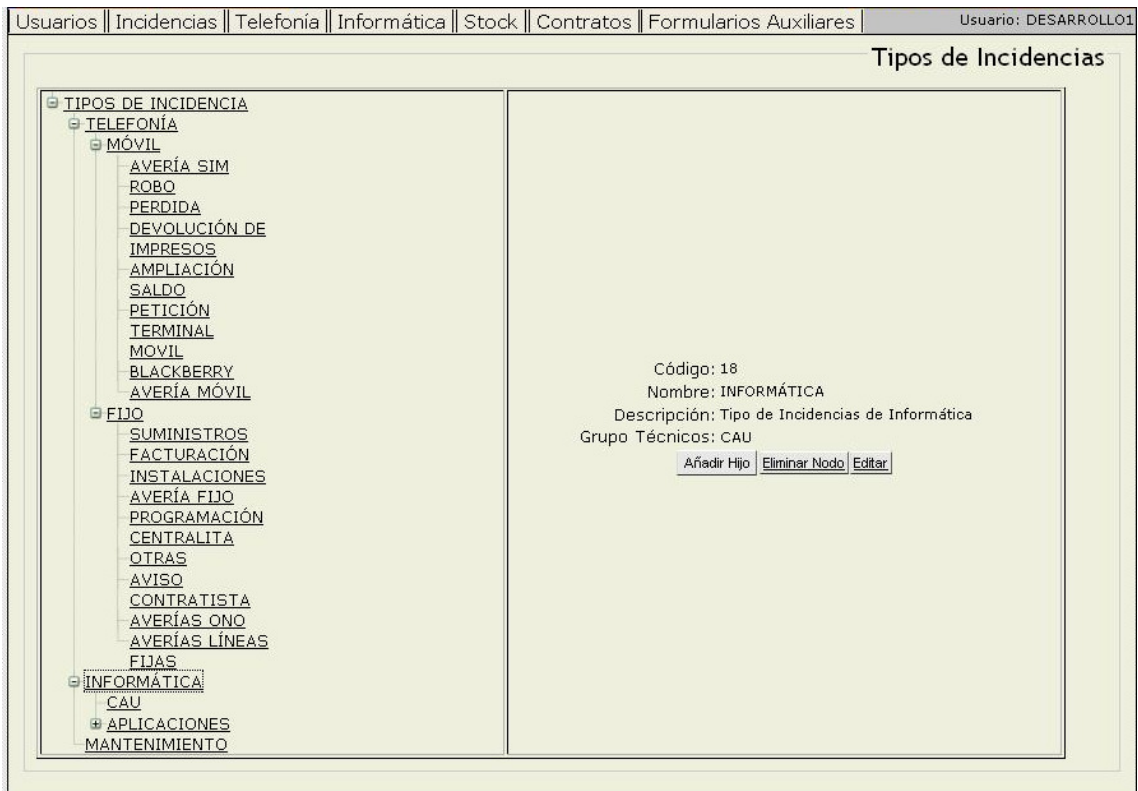
- **Tipos de Incidencias**

Desde este formulario se gestionan los distintos tipos de incidencias. Como se puede observar, se muestran en forma de árbol, ya que existe una jerarquía anidada como se observa en la figura 10. A cada tipo de incidencia le asignamos el grupo técnico “resolvidor”. A continuación, en la figura 11 se muestra la vista del árbol.



**Figura 10** Diagrama Jerarquía Tipos de Incidencias





**Figura 11.** Pantalla Tipos de Incidencias

- **Incidencias Pantalla Búsqueda/Resultados**

En este formulario se pueden filtrar las incidencias, en base a unos criterios, y se muestran los resultados en una tabla.

El usuario sólo verá las incidencias que tenga permitidas atendiendo a su grupo técnico.

Un usuario CAU no verá las incidencias de tipo telefonía y viceversa.

En la figura 12 vemos una tabla paginada con las incidencias resultado de la búsqueda, al seleccionar cualquiera de ellas entraremos en la pantalla detalle que se muestra en la figura 13, donde visualizaremos las características específicas de la incidencia.

Usuarios || Incidencias || Telefonía || Informática || Stock || Contratos || Formularios Auxiliares | Usuario: desarrollo1

### Incidencias

**Búsqueda y selección:**

Año/Nº:  -  F.Entrada (desde - hasta):  -

Clase de Incidencia:  F.Cierre (desde - hasta):  -

Tipo de Incidencia: No seleccionado Asignada a: No seleccionado

Usuario: No seleccionado

Estado: Pendientes Asig.  Asignada  En espera  Resueltas  Anuladas  Rechazada  Rechazadas Usuario

Descripción:  Texto Anotación:

Restablecer | Buscar

Encontrados 3150 resultados.  
Mostrando del 1 al 10

◀◀ 1 2 3 4 5 6 7 8 9 ▶▶

Año	Número	Fecha Entrada	Actualiz.	Usuario	Asignada A	Descripción	Estado	Clase	Tipo Incidencia
2009	502	29/06/09	02/06/09 09:19	CONTRERAS	DESARROLLO2	No se puede cargar el fichero del plan de pensiones 2009.	Resuelta	INC	NEDAES
2009	454	26/06/09	01/06/09 12:51	MAQUILON	TCRUZ	El teléfono no funciona bien	Resuelta	INC	AVERÍA FIJO
2009	649	15/06/09	15/06/09 13:52	ORDENANZAS 3ª MAHONESAS		lol: no va la kyocera de la tercera planta	Pendiente de Asignación	INC	CAU
2009	648	15/06/09	15/06/09 14:29	TCRUZ	TCRUZ	Se solicita en abril presupuesto de cambio de centralita en Casa de Agua, Santomera, sin noticias hasta hoy	Asignada	SUM	TELEFONÍA
2009	647	15/06/09	15/06/09 13:08	Administrador	MJRAMIREZ	Da error al actualizar por http y ftp para coger los nuevos dat diarios. He entrado en el Explorador y le he puseo las dos rutas de donde se lo descarga (ftp.nai.com... y el http://updateer...) y permito en la seguridad mejorada entrar en esas páginas y parece que tampoco va	Asignada	INC	INFORMÁTICA
2009	646	15/06/09	15/06/09 13:07	TCRUZ	TCRUZ	Se solicita 14/06/09 mejora de cobertura móvil para los embalses Camarillas, Fuensanta, Cenajo y Talave	Anulada	SUM	MÓVIL
			15/06/09			Se solicitan terminales por alta de			

Añadir

**Figura 12.** Pantalla búsqueda Incidencias

Además se muestra un multiregistro de Anotaciones. El usuario puede introducir una anotación manual, escribiendo texto y adjuntando un archivo o bien se puede generar una anotación de forma automática:

- Al crear una incidencia y en ella se indica fecha y usuario que la creó.
- Al cambiar el tipo de incidencia, indicando el tipo de incidencia anterior.
- Al cambiar el usuario asignado de su resolución, indicando el usuario asignado anterior.

Usuarios || Incidencias || Telefonía || Informática || Stock || Contratos || Formularios Auxiliares Usuario: desarrollo1

### Incidencias

**Detalles de la Incidencia:**

Año/Nº: 2009/332      Fecha Entrada: 14/05/2009      Fecha Cierre: 18/05/2009

Clase de Incidencia: trabajo      Tipo de Incidencia: INCIDENCIAS E INVENTARIO      Estado: Resuelta

Asignada a: DESARROLLO      Sumistrador: ...      Número Móvil: |

Usuario: DESARROLLO      Terminal: No seleccionado      ICC: No seleccionado

Centralita: ...      Caja: ...      Extensión: ...

Número Fijo: |      Ordenador: ...      Equipo: ...

Descripción: Subir a pruebas nueva versión del gestor de incidencias con el nuevo mecanismo de autorización (roles, aplicaciones, etc.) activado. Crear la/s aplicaciones y asignar los roles adecuados a los usuarios.

Nota al Usuario:

Anular Incidencia Guardar

### Anotaciones

Fecha	Anotación	Fichero
18/05/2009	El usuario desarrollo ha cerrado la incidencia.	
14/05/2009	El usuario desarrollo ha creado la incidencia.	

Añadir

⏪ ⏩ ⏴ ⏵

**Figura 13.** Pantalla Detalle Incidencias

### 4.3. Gestión de Incidencias del Usuario Básico

Las pantallas que se muestran a continuación en las figuras 14,15 y 16 se corresponden con las ventanas desde las que el usuario básico introducirá, buscará y listará sus incidencias.

Cuando hablamos de usuario básico nos referimos a aquel usuario que solo va a utilizar la aplicación para solicitar la resolución de incidencias.

Nueva Incidencia Usuario || Solicitadas Usuario: desarrollo1

### Incidencia

**Nueva Incidencia**

Clase de Incidencia:  Se trata de una incidencia, error o fallo.  
 Se trata de una petición, novedad, mejora o sugerencia.

Tipo de Incidencia: No seleccionado

Descripción:

Guardar

**Figura 14.** Pantalla Creación nueva incidencia

Nueva Incidencia Usuario | Solicitadas | Usuario: desarrollo1

### Búsqueda Incidencias

**Búsqueda y selección:**

Año/Nº

F. Entrada (desde - hasta):  -

Clase de Incidencia:  Se trata de una incidencia, error o fallo.  
 Se trata de una petición, novedad, mejora o sugerencia.  
 Indiferente

Tipo de Incidencia: No seleccionado

Descripción:

**Figura 15.** Pantalla búsqueda de incidencias

Nueva Incidencia Usuario | Solicitadas | Usuario: desarrollo1

### Resultados Incidencias

Encontrados 1 resultados.

Fecha Entrada	Descripción	Año	Número	Tipo Incidencia
26/06/07	Manuel: No Inicia Windows, se queda en la ventana de windows cargando y de hay no pasa. Con el ERD COMMANDER lo he restaurado a un punto anterior, y le he hecho un scandisk y ya arranca, pero el disco esta mal, tiene sectores defectuosos. -Subir el antiguo equipo de Cerro y volcarle la copia de seguridad, para montarlo.	2007	552	INFORMÁTICA

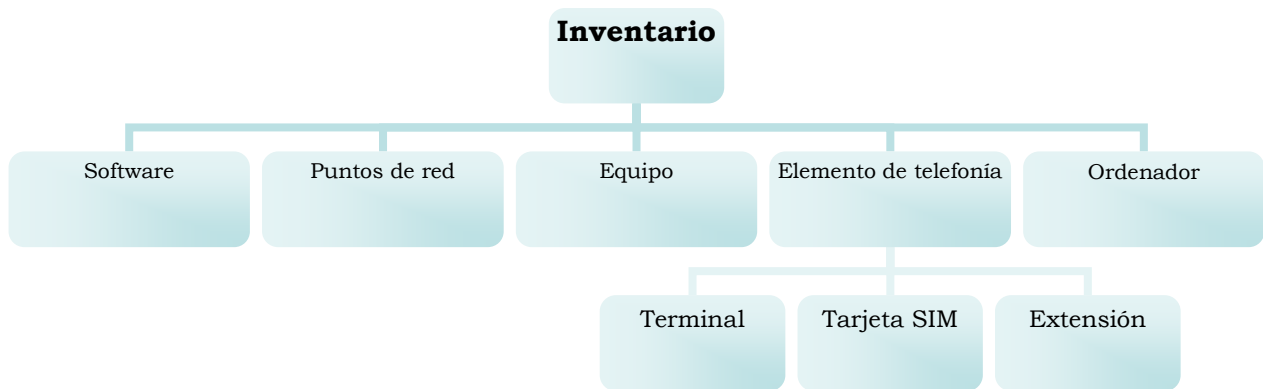
**Figura 16.** Pantalla resultados incidencias

## **5.- GESTIÓN DE INVENTARIO**

## **5. GESTIÓN DE INVENTARIO**

Las funciones del Gestor de Inventario son llevar un registro actualizado del material, equipos informáticos y de red y el software existente la organización en todo momento. De esta manera, se facilitan las tareas de mantenimiento y se puede obtener en cualquier momento una “instantánea” del estado de la organización.

Los elementos inventariables se han identificado tras el proceso de análisis de los requisitos, y se pueden reflejar en la siguiente jerarquía que muestra la figura 17.



**Figura 17.** Diagrama Jerarquía elementos de inventario

Para ello, se han desarrollado una serie de formularios web que se irán explicando a continuación:

- Inventario Software
- Inventario de Puntos de Red
- Inventario de Equipos (no ordenadores)
- Inventario de Ordenadores
- Terminales
- Tarjetas SIM
- Extensiones

- **Inventario Software**

El inventario software es el formulario desde el que se gestiona el software existente en la organización y las licencias disponibles. En la figura 18 se muestra la tabla paginada con los inventarios resultados de la búsqueda. Al seleccionar un inventario en particular visualizamos la pantalla que se muestra en la figura 19.

Usuarios || Incidencias || Telefonía || Informática || Stock || Contratos || Formularios Auxiliares Usuario: desarrollo1

### Inventario de Software

**Búsqueda y selección:**

Archivador:       Nombre:       Descripción:   
 Nº de Cds:       Nº de Caja:       Nº de Licencias:   
 F.Adquisición desde:       F.Adquisición hasta:       F.Validez desde:   
 F.Validez hasta:

Encontrados 156 resultados.  
Mostrando del 1 al 10

	Archivador	Nombre	Descripción	Nº de Cds	Nº de Cajas	Nº de Licencias	Fecha adquisición	Fecha validez
🗑️	OTRAS LICENCIAS ACTIVAS II	Acronis True Image (ATI)	MAS DETALLES EN ANOTACIONES			40	19/10/2007	
🗑️	LICENCIAS MICROSOFT ACTIVAS (V)	ACT. QUOTA ADVISOR EE v 4.0				1	15/05/2000	
🗑️	OTRAS LICENCIAS ACTIVAS II	Activación de Soporte de HP para VMWARE				0	14/11/2007	
🗑️	Licencias Microstation,Geomeia, Autocad Activas	Actualización de GeoMedia Profesional v5.2 Spanish				1	27/09/2004	
🗑️	Licencias Microstation,Geomeia, Autocad Activas	Actualización de GeoMedia SMMS v5.2				1	27/09/2004	
🗑️	Licencias Microstation,Geomeia, Autocad Activas	Actualización de GeoMedia Spanish v5.2				1	27/10/2004	
🗑️	OTRAS LICENCIAS ACTIVAS II	Adobe Acrobat v4.0		2	2	1		
🗑️	OTRAS LICENCIAS ACTIVAS II	Adobe Golive		1	3	1		
🗑️	OTRAS LICENCIAS ACTIVAS II	AppliFilter Add-On v4.0				1	20/06/2006	31/12/2007
🗑️	OTRAS LICENCIAS ACTIVAS II	AppliFilter Add-On v6.0				1	27/11/2007	31/12/2008

Figura 18. Pantalla Búsqueda de Inventarios Software.

Usuarios || Incidencias || Telefonía || Informática || Stock || Contratos || Formularios Auxiliares Usuario: desarrollo1

### Inventario de Software

**Detalles del Producto:**

Archivador:       Nº de Cds:   
 Nombre:       Nº de Caja:   
 Descripción:       Nº de Licencias:       Fecha Adquisición:   
 Fecha Validez:

### Anotaciones

Fecha ↑	Anotación	Fichero

Figura 19. Pantalla Detalle Inventario Software.



- **Inventario Puntos de Red**

El inventario puntos de red es un formulario para inventariar los puntos de red disponibles en la organización, detallando la planta y la dependencia donde se halla ubicado. En figura 20 se muestra el formulario.

**Figura 20.** Pantalla Inventario Puntos de Red.

- **Inventario de Equipos**

Desde este formulario se gestionan todos los equipos, es decir, todo lo que no son ordenadores, como por ejemplo: monitores, faxes, impresoras, modems, como se observa en la figura 21.

Nombre ↑	FechaDesde	FechaHasta
ACALIDAD	24/03/2009	
ADMARC	25/03/2009	

**Figura 21.** Pantalla Inventario de Equipos

- **Inventario de Ordenadores**

La gestión del inventario de ordenadores se lleva a cabo integrando nuestra aplicación con OCS Inventory NG [2]. Esta herramienta consta de agentes instalados en los equipos que se encargan de transmitir la información completa sobre cada ordenador a un servidor con una BD MySQL donde se almacena toda esta información.

Cada día se ejecuta un demonio que actualiza los datos de los ordenadores desde OCS, la fecha de actualización muestra la fecha de la última vez que se completaron los datos.

Algunos de los datos que se actualizan son: N° serie, procesador, memoria, sistema operativo.

Por su parte, en nuestra aplicación hemos desarrollado un formulario de gestión de los ordenadores. Los datos que se muestran en la pestaña Software son también procedentes de OCS y actualizados a diario. En esta pestaña se muestran todos los programas, parches o actualizaciones que tiene instalado cada ordenador. Esta información les será muy útil al Servicio de informática, que sabrá en cada momento la versión de cualquier programa que tiene instalado cada ordenador. En la figura 22 se observa como queda el formulario.

Software

Nombre	Empresa	Versión
7-Zip 4.64	N/A	N/A
Actualización de seguridad para el Reproductor de Windows Media (KB952069)	Microsoft Corporation	N/A
Actualización de seguridad para el Reproductor de Windows Media 10 (KB911565)	Microsoft Corporation	N/A
Actualización de seguridad para el Reproductor de Windows Media 11 (KB936782)	Microsoft Corporation	N/A

**Figura 22.** Pantalla Inventario de Ordenador.

- **Terminales / Tarjetas SIM / Extensiones**

Estos tres formularios tienen las mismas características. Constan de dos multiregistros, en el superior se muestran las entidades que se están gestionando, cuando se selecciona una de ellas se muestra en el inferior los préstamos de usuario que ha tenido, es decir, qué usuarios los han utilizado y durante cuánto tiempo.

Terminales

**Búsqueda y selección:**

IMEI: <input type="text"/>	Fabricante: <input type="text"/>	Modelo: <input type="text"/>
Gama: <input type="text"/>	Tipo: <input type="text"/>	Observaciones: <input type="text"/>
F.Alta desde: <input type="text"/>	F.Alta hasta: <input type="text"/>	F.Baja desde: <input type="text"/>
F.Baja hasta: <input type="text"/>	Situación: <input type="text"/>	Tipo de Recepción: <input type="text"/>

Cuenta Dominio: No seleccionado

Encontrados 230 resultados.  
Mostrando del 11 al 20

1 2 3 4 5 6 7 8 9

Terminal									
	IMEI	Fecha de Baja	Fabricante	Modelo	Gama	Tipo	Situación	Recepción	Observaciones
	351892015442069			E 50	Media		En Servicio		Para Juan Ginés
	351892015973568	17/10/2008		E 50	Media		En Baja		Lo devuelve sin cargador
	352042022352484			6300	Media		En Servicio		
	352042022352674			6300	Media		En Servicio		**

Usuario

Incidencias

Usuario			
	Nombre ↑	FechaEntrega	FechaRecogida
	MARCO	27/02/2007	11/09/2008

**Figura 23.** Pantalla Terminales.

Tarjetas SIM

**Búsqueda y selección:**

ICC:  N° Móvil:  PIN:   
 PUK:  Límite Facturación:  F.Recepción desde:   
 F.Recepción hasta:  Permisos:  Situación:   
 Abreviado:  Observaciones:  Oculto:  Sí  No  Indiferente

Encontrados 405 resultados.  
Mostrando del 1 al 10

Tarjetas SIM													
			ICC	N° Móvil	PIN	PUK	Límite Facturación	Fecha Recepción	Permisos	Abreviado	Observaciones	Situación	Oculto
			01	649945590			0	01/01/2001	INTERNAC	3590		En Baja	<input checked="" type="checkbox"/>
			015	606941154			0	27/07/2004	INTERNAC	3154		En Baja	<input checked="" type="checkbox"/>
			0195000334701	627572910	1756	87508004	0	29/09/2003	30E	3910		En Servicio	<input type="checkbox"/>
			0195000334719	627571954	7331	20086472	0	29/09/2003	45?	3954		En Servicio	<input type="checkbox"/>
			0195000334703	627572910	1756	87508004	0	29/09/2003	30E	3910		En Servicio	<input type="checkbox"/>

---

**Usuario**

	Nombre ↑	FechaDesde	FechaHasta
	ADASA	17/03/2009	
	FUENTES	01/01/2000	04/05/2005

Figura 24. Pantalla Tarjetas SIM.

Usuarios | Incidencias | Telefonía | Informática | Stock | Contratos | Formularios Auxiliares Usuario: desarrollo1

### Extensiones

**Búsqueda y selección:**

Extensión:       Caja:       Categoría:   
 Dependencia:       Planta:       Despacho:   
 Directo:       Contraseña:       Observaciones:

Encontrados 372 resultados.  
Mostrando del 61 al 70

Extensión									
		Extensión	Caja	Categoría	Tipo Extensión	Despacho	Contraseña	Directo	Observaciones
		1059	3		Pública		187531	968965059	
		1060	3		Pública		187087	968965060	
		1061	7		Pública		185529	968965061	
		1062	7		Pública		181700	968965062	
		1063	7		Pública		185248	968965063	
		1064	7		Pública		183026	968965064	

### Usuario

	Nombre ↑	FechaDesde	FechaHasta
	MIRO	01/01/2000	

**Figura 25.** Pantalla Extensiones

## **6.- DISEÑO E IMPLEMENTACIÓN**

## **6. DISEÑO E IMPLEMENTACIÓN**

### **6.1. Patrones de Diseño**

Los patrones de diseño [17] son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Los patrones resuelven con efectividad un problema concreto, y para ello debemos identificar primero éstos problemas y qué patrón aplicar en cada caso.

El uso de patrones de diseño es una técnica comúnmente aceptada y proporciona numerosos beneficios (reusabilidad, claridad, sencillez al implementar, etc.). A continuación detallamos los principales patrones empleados.

#### **6.1.1. Abstract Factory**

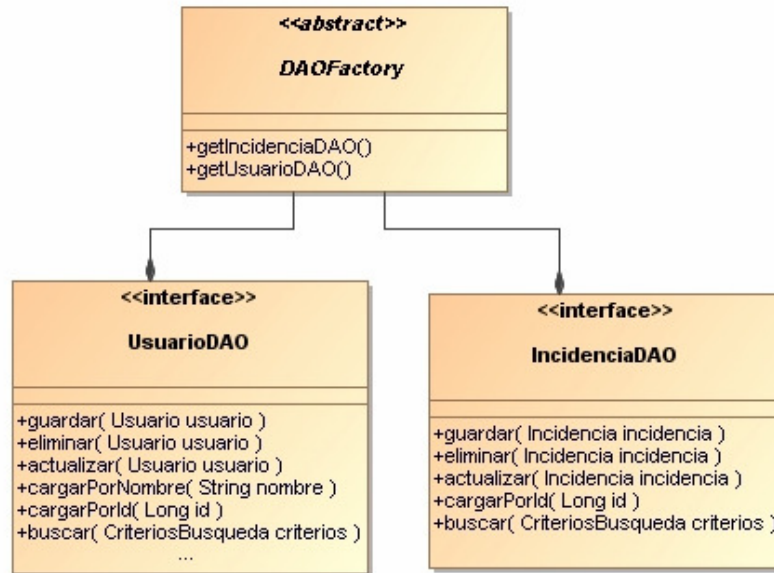
Este patrón intenta crear diferentes familias de objetos de forma que no se mezclen entre sí. Lo utilizaremos en el patrón DAO, para crear los objetos DAO de una misma familia.

Los clientes utilizan una misma factoría abstracta común que será implementada por cada una de las familias que se desee crear.

#### **6.1.2. Data Access Object (DAO)**

Este patrón ofrece una interfaz común entre la aplicación y varios dispositivos de almacenamiento de datos, como una base de datos, o un archivo. De esta manera ningún objeto de negocio requiere conocimiento directo del destino final de la información manipulada, abstrayéndolo de la tecnología de persistencia subyacente. Será necesario crear por cada objeto del

que se necesite realizar persistencia un objeto DAO con las acciones de persistencia sobre éste, como crear, buscar, actualizar, eliminar, etc. En la figura 26 se muestra el diagrama de clases que ilustra cómo implementan el patrón DAO las clases Incidencia y Usuario.



**Figura 26.** Diagrama de clases del patrón DAO.

En la figura 27 podemos ver el código de la interfaz de IncidenciaDAO. Se muestran las operaciones que se pueden realizar sobre la clase Incidencias.



```

package es.chs.gestorIncidencias.dao;

import java.util.List;

import org.hibernate.Criteria;

import es.chs.gestorIncidencias.búsqueda.CriteriosBusquedaIncidencia;
import es.chs.gestorIncidencias.modelo.Incidencia;

public interface IncidenciaDAO {
    public void guardar(Incidencia obj);
    public void actualizar(Incidencia obj);
    public void eliminar(Incidencia obj);
    public List<Incidencia> cargarTodos();
    public Incidencia cargarPorId(long id);
    public List<Incidencia> cargarPorNombre(Integer nombre);
    public List<Incidencia> buscar(CriteriosBusquedaIncidencia criterios);
    public int obtenerNumeroMaximo(int año);

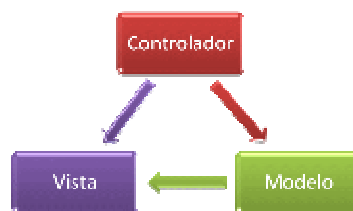
    public Criteria creaConsulta(CriteriosBusquedaIncidencia criterios);
    public List buscarRango(CriteriosBusquedaIncidencia criterios,int
posicionInicial, int maxResults);
    public Integer
obtenerNumeroResultadosBuscar(CriteriosBusquedaIncidencia criterios);

```

**Figura 27.** Código de la interfaz IncidenciaDAO

## 6.2. Integración de JSF + Spring + Hibernate

El proyecto presenta una arquitectura en tres capas implementando el MVC (Modelo-Vista-Controlador). En la figura 28 se muestra un pequeño diagrama que ilustra la interacción entre éstas tres capas.

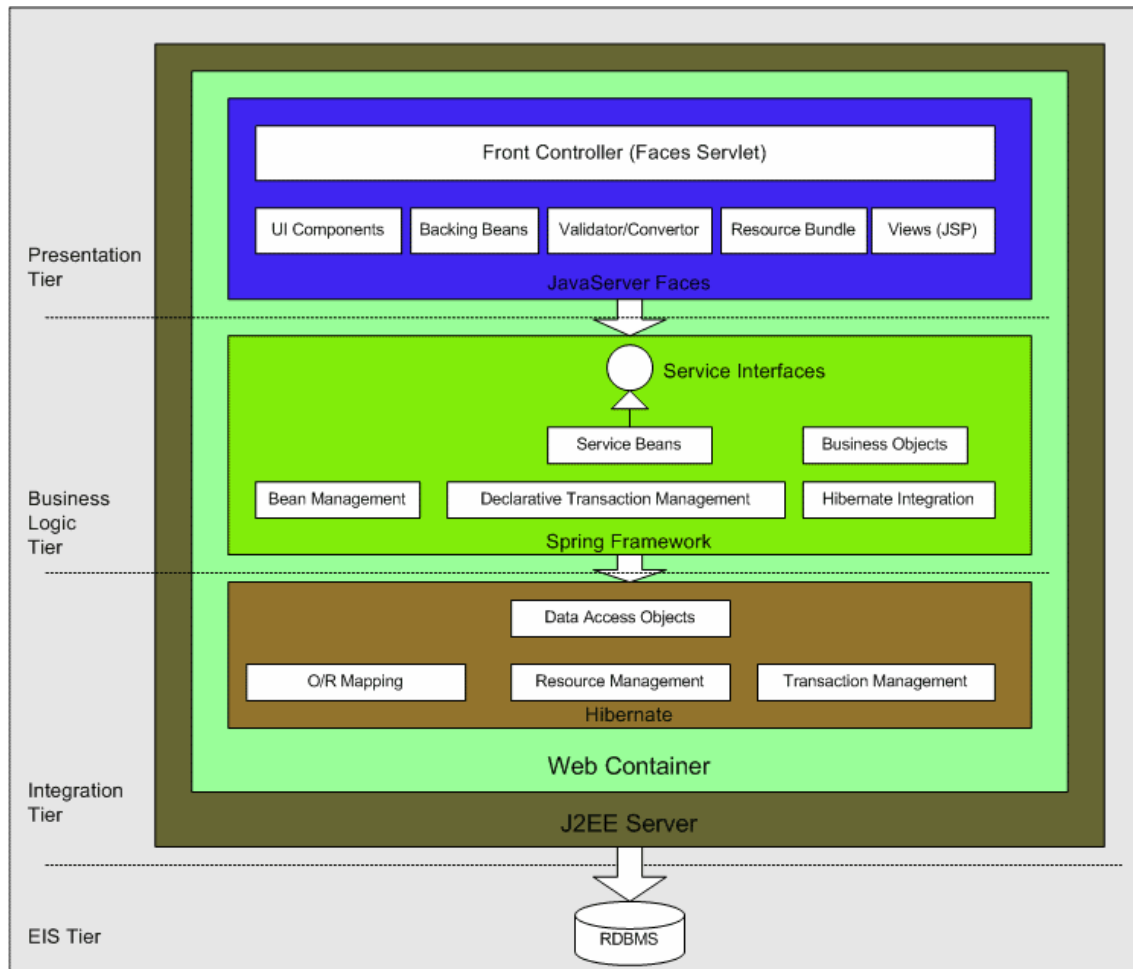


**Figura 28.** Diagrama que ilustra la estructura MVC.

El objetivo principal de la arquitectura es separar, de la forma más limpia posible, las distintas capas de desarrollo facilitando el mantenimiento y evolución de las aplicaciones.

Se deseaba una arquitectura que permitiese trabajar en capas y que sirviese tanto para las aplicaciones en la intranet como en Internet, para lograr esto se eligió el patrón MVC (Modelo-Vista-Controlador) que permite una separación limpia entre las distintas capas de una aplicación.

En la figura 29 se muestra cómo se concreta esta capa para el caso concreto de la integración entre JSF, Spring, Hibernate y un servidor J2EE.



**Figura 29.** Arquitectura multicapa [11].

### 6.2.1. Capa de presentación

Para la capa de presentación (la vista) se buscaba un framework que nos proporcionase una mayor facilidad en la elaboración de pantallas, mapeo entre los formularios y sus clases en el servidor y que facilitase también el incluir componentes complejos (menús, árboles, ajax, etc) de una forma sencilla y – sobre todo- fácil de mantener. Para esta capa se ha elegido JSF (JavaServer Faces) [12].

Se ha utilizado la implementación que hace MyFaces[7] de JSF, utilizándose la librería de componentes Tomahawk, también de MyFaces, y la librería Ajax for JSF [13]

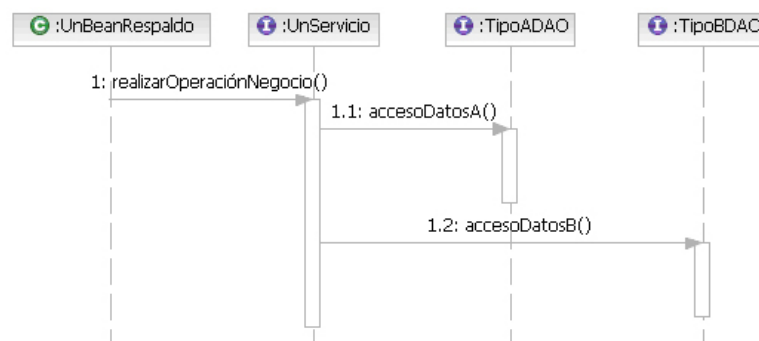
### 6.2.2. Capa de negocio

En la capa de negocio y persistencia, se decidió desde el primer momento no emplear EJB's por su elevado coste de desarrollo y mantenimiento así como su falta de flexibilidad y coste en tiempo para los cambios.

Usamos el Framework Spring [8] que es de código abierto e incluye muchas extensiones, por ejemplo, en este proyecto se hace uso de Spring LDAP y de Spring Integration.

- Spring LDAP [15], utilizado para conectar la aplicación con el servicio de Directorio Activo.
- Spring Integration [16] utilizado para conectar con un buzón de correo.

La persistencia de las clases se sustenta en DAO's (Objetos de Acceso a Datos), manteniendo aislada la capa de persistencia de la capa de negocio. Tanto los servicios como los DAO's así como el propio modelo son realmente POJOs (clases simples de Java), con la simplicidad que conllevan y sin dependencias reales con ningún framework concreto. Para realizar esta integración se usa Spring. En la figura 30 se muestra cómo sería una operación de negocio con Spring.



**Figura 30.** Operación de negocio en una arquitectura con Spring

### 6.2.3. Capa de persistencia

Para la capa de persistencia se pensó en utilizar alguna herramienta ya existente, que permitiese realizar el mapeo objeto-relacional de una forma cómoda pero potente, sin tener que implementarlo directamente mediante JDBC.

La herramienta elegida finalmente fue Hibernate [9].

Hibernate ofrece un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente, Criteria.

Se utiliza Criteria para realizar las consultas a BD. Criteria puede complicarse en caso de consultas anidadas o muy complicadas, pero nuestras consultas son simples.

Hemos utilizado el componente Hibernate Annotations que implementa el estándar JPA, que es parte de esta plataforma, de esta forma no necesitamos crear los archivos de mapeo para crear la correspondencia con los campos de la tabla. En este caso en la clase del modelo se incluyen anotaciones, de la forma que se muestra en la figura 31.

```

@Entity
@Table(name = "INCIDENCIA")
public class Incidencia implements Cloneable{

    private Long id;

    @Id
    @GeneratedValue
    @Column(name = "id", unique = true, nullable = false)
    public Long getId() {
        return id;
    }

    private Usuario usuario;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name="id_usuario")
    public Usuario getUsuario() {
        return usuario;
    }
}

```

**Figura 31.** Ejemplo de anotaciones.

### 6.3. Integración con AD

La aplicación está preparada para la comunicación con el Directorio Activo. Se utiliza para ello la librería Spring LDAP del Framework Spring, que permite una comunicación sencilla entre la aplicación y el directorio Activo.

El código que se muestra en la figura 32 se encuentra ubicado en el fichero applicationContext.xml y configura la conexión.

```

<bean id="contextSource" c
class="org.springframework ldap.core.support.LdapContextSource">
    <property name="url" value="ldap://dominator01.confederacion.dom:389" />
    <property name="base" value="dc=CONFEDERACION,dc=DOM" />
    <property name="userDn"
        value="CN=A.T. de informática para
desarrollo,OU=ASISTENCIAS,OU=SECRETARIA,OU=USUARIOS DEL
DOMINIO,DC=CONFEDERACION,DC=DOM" />
    <property name="userDn" value="CN=Usuario para la aplicación Gestión de
Incidencias,OU=USUARIOS VARIOS,DC=CONFEDERACION,DC=DOM" />
    <property name="password" value="estuche" />

</bean>

```

**Figura 32.** Código implementación conexión Directorio Activo.



## 6.4. Implementación de demonios en Spring

Se han implementado 3 demonios de importación de datos diarios. Todos los días estos demonios se ejecutan y actualizan datos en la BD.

- *Demonio actualización Usuarios.*  
Se conecta al Directorio Activo y comprueba si se han producido modificaciones en LDAP respecto de la BD. Si se han actualizado los datos de cualquier usuario o se ha creado alguno nuevo se modifica en BD.
- *Demonio actualización datos procedentes de OCS.*  
Se conecta a la BD MySQL de OCS, para actualizar los datos de los pcs: características hardware, software instalado.
- *Demonio comprobación de tareas y generación de incidencias automática.*  
Este demonio recorre la tabla Tareas, comprobando si hay tareas pendientes para la fecha indicadas. Si las hay se genera una incidencia con los datos que se indican en la tarea.

El código que se muestra en la figura 34 corresponde a la configuración que se incluye en el fichero applicationContext.xml, en el que se indica la hora a la que se realizará la importación de datos y el tiempo período de comprobación que se le asigna al demonio, cuando se cumpla este tiempo el demonio se despierta para volver a ejecutarse.

En la figura 35 mostramos el código que se utiliza para implementar uno de los demonios, el de comprobación de tareas diarias para ejecutar ese día.

```

<bean id="demonioImportacionUsuariosLDAP"
class="es.chs.gestorIncidencias.demonios.DemonioImportacionUsuariosLDAP"
    scope="singleton" lazy-init="false" init-method="start">
    <property name="nombreDemonio">
        <value>Demonio de importación de usuarios del Active
Directory</value>
    </property>
    <property name="horaEjecucion">
        <value>06:00</value> <!-- Hora a la que se activa el demonio
(habrà que tener en cuenta que hay un error maximo de +periodoComprobacion) --
>
    </property>
    <property name="periodoComprobacion">
        <value>1</value> <!-- Minutos que duerme el demonio antes
de despertar y comprobar/ejecutar -->
    </property>
</bean>

    <bean id="demonioImportacionOCS"
class="es.chs.gestorIncidencias.demonios.DemonioImportacionOCS"
    scope="singleton" lazy-init="false" init-method="start">
    <property name="nombreDemonio">
        <value>Demonio de importación de información recogida por
OCS</value>
    </property>
    <property name="horaEjecucion">
        <value>06:10</value> <!-- Hora a la que se activa el demonio
(habrà que tener en cuenta que hay un error maximo de +periodoComprobacion) --
>
    </property>
    <property name="periodoComprobacion">
        <value>15</value> <!-- Minutos que duerme el demonio
antes de despertar y comprobar/ejecutar -->
    </property>
</bean>

    <bean id="demonioComprobacionTareas"
class="es.chs.gestorIncidencias.demonios.DemonioComprobacionTareas"
    scope="singleton" lazy-init="false" init-method="start">
    <property name="nombreDemonio">
        <value>Demonio de comprobación tareas y generación
automática de incidencias</value>
    </property>
    <property name="horaEjecucion">
        <value>06:20</value> <!-- Hora a la que se activa el demonio
(habrà que tener en cuenta que hay un error maximo de +periodoComprobacion) --
>
    </property>
    <property name="periodoComprobacion">
        <value>3</value> <!-- Minutos que duerme el demonio antes
de despertar y comprobar/ejecutar -->
    </property>
</bean>

```

**Figura 34.** Código de la configuración de los demonios en el fichero applicationContext.xml.

```

package es.chs.gestorIncidencias.demonios;

import java.util.Calendar;
import java.util.Date;
import java.util.List;

import javax.annotation.Resource;

import org.springframework.context.annotation.Scope;

import es.chs.gestorIncidencias.modelo.AnotacionIncidencia;
import es.chs.gestorIncidencias.modelo.ClaseIncidencia;
import es.chs.gestorIncidencias.modelo.Estado;
import es.chs.gestorIncidencias.modelo.Incidencia;
import es.chs.gestorIncidencias.modelo.Tarea;
import es.chs.gestorIncidencias.servicios.ServicioClaseIncidencia;
import es.chs.gestorIncidencias.servicios.ServicioEstado;
import es.chs.gestorIncidencias.servicios.ServicioIncidencia;
import es.chs.gestorIncidencias.servicios.ServicioTarea;
import es.chs.gestorIncidencias.util.Constantes;
import es.chsegura.comun.demonios.AbstractDemonioDiario;
import es.chsegura.comun.excepciones.ErrorAccesoADatosException;

/**
 * Demonio que se ejecuta una vez al día y comprueba las tareas activas (fecha fin <
 fecha actual)
 * Para aquellas tareas cuyo periodo de creación se cumpla, generará la incidencia
 correspondiente.
 * @author DESARROLLO
 */

@Scope("singleton")
public class DemonioComprobacionTareas extends AbstractDemonioDiario {

    private static final org.apache.log4j.Logger logger =
org.apache.log4j.Logger.getLogger(DemonioComprobacionTareas.class);

    @Resource
    private ServicioTarea servicioTarea;
    @Resource
    private ServicioIncidencia servicioIncidencia;
    @Resource
    private ServicioEstado servicioEstado;
    @Resource
    private ServicioClaseIncidencia servicioClaseIncidencia;

    public ServicioTarea getServicioTarea() {
        return servicioTarea;
    }
    public void setServicioTarea(ServicioTarea servicioTarea) {
        this.servicioTarea = servicioTarea;
    }

    public ServicioIncidencia getServicioIncidencia() {
        return servicioIncidencia;
    }
}

```



```

public void setServicioIncidencia(ServicioIncidencia servicioIncidencia) {
    this.servicioIncidencia = servicioIncidencia;
}

public ServicioEstado getServicioEstado() {
    return servicioEstado;
}

public void setServicioEstado(ServicioEstado servicioEstado) {
    this.servicioEstado = servicioEstado;
}

public ServicioClaseIncidencia getServicioClaseIncidencia() {
    return servicioClaseIncidencia;
}

public void setServicioClaseIncidencia(
    ServicioClaseIncidencia servicioClaseIncidencia) {
    this.servicioClaseIncidencia = servicioClaseIncidencia;
}

protected void realizarAccion() {

    logger.info("Comprobación de Tareas para generación automática de
Incidencias...");
    if (this.getServicioTarea()==null){
        logger.error("No se encontró el servicio Tareas");
        return;
    }
    if (this.getServicioIncidencia()==null){
        logger.error("No se encontró el servicio Incidencias");
        return;
    }

    // hoy tendrá la fecha de hoy con exactitud hasta el día y la hh:mm:ss
y mils a 0.
    Calendar c = Calendar.getInstance();
    Calendar hoy = Calendar.getInstance();
    hoy.clear();
    hoy.set(c.get(Calendar.YEAR), c.get(Calendar.MONTH),
c.get(Calendar.DATE),0,0);
    List<Tarea> listaTareas=null;
    try {
        listaTareas = this.getServicioTarea().cargarActivas();
    } catch (ErrorAccesoADatosException e) {
        logger.error("Error al cargar tareas de BBDD: " + e.getMessage());
        return;
    }
    int generadas=0;
    for (Tarea t:listaTareas){

        // Si la tarea ya ha generado una incidencia hoy pues nos la
saltamos

        if (t.getFechaUltimaActivacion()!=null)
            if (t.getFechaUltimaActivacion().getTime() ==
hoy.getTime().getTime())
                continue;

        logger.debug("Procesando tarea activa: " + t.getId() + " - " +
t.getTitulo());

```

```

// Comprobamos si la tarea debe generar una incidencia hoy.
Para ello, hay que ver si // los días transcurridos desde la fecha de inicio de la tarea son
múltiplo del periodo // de la tarea.
long dias = ( hoy.getTimeInMillis() - t.getFechaInicio().getTime() )
/ (1000 * 60 * 60 * 24);
if ((dias % t.getDias().intValue()) == 0) {
    logger.debug("Generando incidencia para la tarea...");
    Incidencia incid = new Incidencia();

// Asignamos los campos de la Incidencia que podemos
sacar de la Tarea o averiguar.
    incid.setNumero(new Integer(0)); // Al guardarla se le
asignará automáticamente el número que le corresponda.
    incid.setAnyo(Calendar.getInstance().get(Calendar.YEAR));
    incid.setFechaEntrada(new Date()); // le asignamos la
fecha actual
    incid.setDescripcion(t.getDescripcion());
    incid.setTipoIncidencia(t.getTipoIncidencia()); // El tipo
de incidencias que se asignó en la tarea.
    incid.setAsignada(t.getAsignado()); //Persona a la que se
le asigna la tarea
    incid.setUsuario(t.getUsuario()); // El usuario que dio de
alta la tarea.

// FIXME Falta esto !!!

    Estado estad;
    try {
        if(incid.getAsignada()!=null){
            estad=(Estado)this.getServicioEstado().cargarPorNombre(Constantes.ESTADO_I
NCIDENCIA_ASIGNADA).get(0);
        }
        else estad =
(Estado)this.getServicioEstado().cargarPorNombre("Pendiente de Asignación").get(0);
    } catch (Exception e1) {
        logger.error("No se pudo recuperar el estado
'Pendiente de Asignación': " + e1.getMessage());
        continue;
    }
    incid.setEstado(estad);

    ClaseIncidencia clase;
    try {
        clase =
(ClaseIncidencia)this.getServicioClaseIncidencia().getClaseIncidenciaPorNombre("TRAB
AJO").get(0);
    } catch (Exception e1) {
        logger.error("No se pudo recuperar la clase de
incidencia 'TRABAJO': " + e1.getMessage());
        continue;
    }
    incid.setClaseIncidencia(clase);

```

```

//                               incid.setClaseIncidencia(...); // Hay que asignarle
clase trabajo.                               incid.setEstado(...); // hay que asignarle el estado
//
Pendiente de asignacion.

                                AnotacionIncidencia ai= new AnotacionIncidencia();
                                ai.setAnotacion("Incidencia autogenerada por tarea: " +
t.getTitulo());                                ai.setFecha(new Date()); // Fecha de hoy
                                                incid.addAnotacion(ai);

                                // Guardamos la Incidencia
                                try {
                                    this.getServicioIncidencia().guardar(incid);
                                    logger.debug("Incidencia Generada: " +
incid.getNumero()+"/"+incid.getAnyo());
                                    generadas++;
                                } catch (ErrorAccesoADatosException e) {
                                    logger.error("Error al guardar incidencia en BBDD
para tarea con id=" + t.getId() + ": " + e.getMessage());
                                    continue;
                                }

                                // Actualizamos la tarea para indicar que hoy ya se ha
generado la incidencia asociada.
                                // Esto es necesario por si, por ejemplo, se reinicia la
aplicación. En ese caso, no habría forma
                                // de saber si se ha ejecutado la tarea o no ya en este día
sin esta fecha.

                                t.setFechaUltimaActivacion(hoy.getTime());
                                try {
                                    this.getServicioTarea().actualizar(t);
                                } catch (ErrorAccesoADatosException e) {
                                    logger.error("Error al actualizar fecha ultima
activación tarea con id=" + t.getId() + " en BBDD: " + e.getMessage());
                                    continue;
                                }
                            }
                    }

                    if (generadas>0)
                        logger.info("Autogeneradas " + generadas + " incidencias por
tareas.");
                    else
                        logger.info("Ninguna incidencia autogenerada por tareas.");
                }
            }
}

```

**Figura 35.** Código demonio comprobación de tareas.

## 6.5. Autogeneración de incidencias a partir de la lectura de correos electrónicos.

Dado que en la mayoría de organismos la atención al usuario se gestiona mediante peticiones al correo de atención al usuario, pareció útil ampliar la funcionalidad del proyecto de forma que éste se conecte al buzón de correo de atención al usuario y transforme el texto de petición del correo en una incidencia en nuestra aplicación.

Para implementar esta funcionalidad vamos a utilizar Spring Integration [14].

Spring Integration permite mensajería entre nuestra aplicación que utiliza Spring y sistemas externos vía simples adaptadores.

Estos adaptadores proveen un nivel de abstracción alto para soportar Spring remoting, mensajería y scheduling.

Se puede explicar de forma resumida de la siguiente forma:

- **Mensajes (Messages)**  
Los mensajes son la abstracción principal de cualquier middleware orientado a mensajes. Los Mensajes pueden ser hechos con Java. Estos son pasados de endpoints a endpoints usando canales (channels).
- **Endpoints(puertos)**  
Endpoints son los componentes que desean enviar o recibir un mensaje. Se pueden hacer grandes cadenas de endpoints a través de canales (pipes) desde el punto en que es enviado, hasta que es entregado.
- **Canales (Channels)**  
Estos son los pipes que almacenan nuestros mensajes durante el tránsito de un endpoint a otro. Los canales pueden ser: Direct channels (Los mensajes se pasan de un endpoint a otro endpoint en el mismo hilo), este es el comportamiento por defecto, o PublishAndSubscribe channels (Los mensajes son enviados a todos los consumidores suscritos de manera asíncrona).

En la figura 36 se muestra el código de cómo se configura la conexión:

```

<!-- Canal para leer mensajes desde el buzón -->
    <mail:inbound-channel-adapter    channel="mailIn"
        store-
uri="pop3://gestionIncidencias:estuche@cronos.confederacion.dom/INBOX">
        <integration:poller max-messages-per-poll="5">
            <integration:interval-trigger interval="90" time-
unit="SECONDS"/>
        </integration:poller>
    </mail:inbound-channel-adapter>

    <!-- Todos los mensajes de entrada son recibidos en este canal -->
    <integration:channel id="mail" />

    <!-- this service activator gets activated    for any inbound mail arrive at
mainIn
        and calls the receiveMail method -->
    <integration:service-activator input-channel="mailIn" ref="servicioCorreo"
method="recibeCorreo">
    </integration:service-activator>

```

**Figura 36.** Código conexión buzón correo

La clase que se muestra en la figura 37 ilustra cómo se recupera el mensaje y se lee.

```

import javax.annotation.Resource;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import es.chs.gestorIncidencias.dao.FactoriaDAO;
import es.chs.gestorIncidencias.servicios.ServicioCorreo;

import javax.mail.Address;
import javax.mail.BodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;

@Scope("singleton")
@Service("servicioCorreo")
@Transactional (rollbackFor={ErrorAccesoADatosException.class})
public class ServicioCorreoImpl implements ServicioCorreo {
    @Resource
    private FactoriaDAO factoriaDAO;

    public FactoriaDAO getFactoriaDAO() {
        return factoriaDAO;
    }

    public void setFactoriaDAO(FactoriaDAO factoriaDAO) {
        this.factoriaDAO = factoriaDAO;
    }
}
@Transactional

```

```
public void recibeCorreo(MimeMessage mimemsg){
    try {
        Address[] from=mimemsg.getFrom();
        String sender=from[0].toString();

        String senderMailId=MailFormatUtils.getUserMailId(sender);
        String senderName=MailFormatUtils.getUserName(sender);
        String subject=mimemsg.getSubject();
        String bugId=MailFormatUtils.getBugId(subject);

        System.out.println("Sender: " + sender);
        System.out.println("Subject: " + mimemsg.getSubject());

        MimeMultipart mul=(MimeMultipart)mimemsg.getContent();
        BodyPart bp= mul.getBodyPart(0);
        String mailContent=bp.getContent().toString();
        String comments=MailFormatUtils.getComments(mailContent);
        System.out.println("Contentet: " + mailContent);

        try{
            ;
            incidencias.storeHistoryByMail( bugId, comments, senderMailId,
senderName, mimemsg.getSentDate());
            System.out.println("history was added!!!");
        }
        catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
    catch(Exception e){
        e.printStackTrace();
    }
}
}
```

**Figura 37.** Código clase que lee correo electrónico.

## **7.- CONCLUSIONES Y VÍAS FUTURAS**

## 7. CONCLUSIONES Y VÍAS FUTURAS

Según lo expuesto en este proyecto se ha llevado a cabo la creación de un gestor de incidencias e inventario. Se ha unificado funcionalidad que en muchas organizaciones se gestiona de forma independiente.

Se ha desarrollado una aplicación, que además de lo anteriormente expuesto, se integra con:

- El servicio de Directorio Activo para gestionar los Usuarios.
- Una herramienta automática de detección de ip para inventariar los equipos.
- Se conecta con una cuenta de correo generando incidencias automáticas a partir del mensaje de correo.

En primer lugar se realizó un análisis de la solución, estudiando la viabilidad, metodología y los requisitos del sistema.

Se mostraron las capturas de pantalla de las diferentes vistas de la aplicación donde se puede ver la información recopilada.

Se realizó un diseño e implementación del sistema donde se muestran los aspectos más importantes de diseño como por ejemplo la integración de las tres tecnologías: JSF, Spring e Hibernate.

Esta aplicación es altamente extensible y adaptable a nuevos requisitos. Los frameworks que se utilizan son actuales y están en pleno desarrollo por lo que un desarrollo futuro no debe de suponer ningún problema.



## **8.- ANEXO I**

## **8. ANEXO I**

### **8.1. Funcionalidades actuales de Dumbo**

#### **8.1.1. Solicitud de nuevas tareas**

Tanto para los usuarios como para los expertos informáticos. Comprende la solicitud on-line de tareas de distinto tipo (incidencia, petición, problema), así como el dumbo rápido (tareas de atención telefónica que se abren y finalizan al mismo tiempo) y, la asignación automática al experto informático que tiene que resolver la tarea.

El aspecto del formulario de solicitud de tarea se puede observar en la figura 38

#### **8.1.2. Exploración de tareas solicitadas**

Todos los usuarios de sistema Dumbo podrán realizar un seguimiento de la evolución de las tareas solicitadas y saber, en cada caso, quién es el experto a cargo de las mismas, así como el estado en que se encuentra una tarea. Incluye un sistema de filtros avanzados y opciones para personalizar los contenidos que desea visualizar en cada momento. Un ejemplo de esta aplicación se muestra en la figura 39.

» Nueva Tarea (\*) campos obligatorios.

**Origen/Destino**

Solicitante(\*):  Informático:

**Datos Tarea**

Asunto(\*):

Texto Tarea(\*):

Tipo de Tarea(\*):  Se trata de una petición, novedad, mejora o sugerencia.  
 Se trata de una incidencia, error o fallo.

Aplicación(\*):

Urgencia:

Email Copia:

Ext.Contacto(\*):

Ubicación Tarea:

Anexos:  Examinar...  Examinar...  
 Examinar...  Examinar...  
 Examinar...  Examinar...

Figura 38. Solicitud de tarea.

» Solicitadas por resolver

	Estado	Tarea	Aplicación	Asunto	Usuario	Informático	Creada
[1]		245	JUS	<a href="#">Prueba</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	18/07/2006
[2]		242	JUS	<a href="#">Va con retarodkas</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	06/07/2006
[3]		241	JUS	<a href="#">Prueba con Thread</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	06/07/2006
[4]		231	JUS	<a href="#">Otra Prueba</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	06/07/2006

Incidencia  Petición

Figura 39. Seguimiento de tareas solicitadas.

### 8.1.3. Atención de tareas recibidas

Los expertos que trabajen con sistema Dumbo podrán gestionar todas las tareas que les hayan sido asignadas, teniendo a su alcance información completa sobre ellas, así como opciones para crear subtareas, reasignar una tarea, anularla, devolverla a la cola genérica, gestionarla (modificación datos, historial actuaciones, cuaderno bitácora, adjuntar archivos), categorizar la tarea, pasarla a espera, priorizarla, planificarla o finalizar su actividad. En la figura 40 se muestra un listado de las tareas pendientes y en la figura 41 se muestra un detalle de la tarea.

DUMBO7. Área de Tecnologías de la Información y las Comunicaciones Aplicadas de la Universidad - Microsoft Internet Exp...

### Filtro de Información: Listado atender

**Tipo Tarea:**  Petición  Incidencia

**Estado Tarea:**  Asignada  En Espera  Rechazada  Anulada

**Aplicación:**  Sin Aplicación Asignada

[Sección: DESARROLLO, APLICACIONES Y METODOLOGIA]  
[Sección: SECCION DE PROYECTOS Y APLICACIONES]  
[Sección: CURIE/RIGA/EUC/SPE Y OTROS]  
DUMBO  
[Sección: SECCION DE DESARROLLO]  
SUBAPL. SUMA  
[Sección: SECCION INICIAL]  
APLICACION DE PRUEBA 7

**Urgencia:**  Urgente  
 No Urgente  
 Sin Urgencia

**Prioridad:**  Alta  Media  Baja  
 Crítica  Planificada  Sin  
Prioridad

**Fecha Inicio:** Desde   Hasta

**Usuario:**

**Asunto:**

**Texto:**

**Código:**

**Figura 40.** Listado de tareas pendientes.

**Figura 41.** Detalle tarea.

Las acciones que podemos llevar a cabo con una tarea dependerán del estado de esta y se dividen en tres bloques que se muestran en la figura 42.

<u>Ubicar Tarea</u>	<u>Gestionar Tarea</u>	<u>Finalizar Tarea</u>
<ul style="list-style-type: none"> <li>· <a href="#">Reasignar Tarea</a></li> <li>· <a href="#">Anular</a></li> <li>· <a href="#">Devolver a la Cola</a></li> </ul>	<ul style="list-style-type: none"> <li>· <a href="#">Es Incidencia</a></li> <li>· <a href="#">Pasar a Espera</a></li> <li>· <a href="#">Crear Subtarea</a></li> </ul>	<ul style="list-style-type: none"> <li>· <a href="#">Finalizar Tarea</a></li> </ul>

**Figura 42.** Acciones sobre una Tarea.

## 8.2. Gestión global

El sistema Dumbo permite la definición de secciones y aplicaciones/servicios los cuales tienen asignados responsables y expertos. Los responsables, mediante esta opción, pueden gestionar las tareas de cualquier experto de la aplicación/servicio al cual coordina, así como todas las tareas a nivel de aplicación/servicio o a nivel de sección, según sea su grado de responsabilidad. Esta opción comprende una gestión global por sección, por aplicación y por experto e incluye además una opción de suplantación de identidad (informático/usuario). Cada una de estas opciones se detallan más extensamente a continuación.

### 8.2.1. Gestión de Dumbo por secciones

La gestión por secciones, permite ver y gestionar tareas vinculadas a aplicaciones de una determinada sección y sus subsecciones. Desde el listado

podemos ver las aplicaciones y los informáticos vinculados. En la figura 43 se muestra un ejemplo de gestión de Dumbo por secciones.

Gestión Global de Dumbos Por Secciones		
MAESTRO: GESTIONA DE CUALQUIER SECCIÓN		
REDES	<a href="#">Aplicaciones</a>	<a href="#">Informáticos</a>
SECCION DE DESARROLLO	<a href="#">Aplicaciones</a>	<a href="#">Informáticos</a>
METODOLOGIAS	<a href="#">Aplicaciones</a>	<a href="#">Informáticos</a>
PROYECTOS	<a href="#">Aplicaciones</a>	<a href="#">Informáticos</a>
SUB PROYECTOS	<a href="#">Aplicaciones</a>	<a href="#">Informáticos</a>
SECCIÓN INICIAL	<a href="#">Aplicaciones</a>	<a href="#">Informáticos</a>
SISTEMAS	<a href="#">Aplicaciones</a>	<a href="#">Informáticos</a>

**Figura 43.** Gestión de sección.

Una vez seleccionada la sección, podemos obtener la lista de tareas vinculadas a la misma. En la figura 44 se muestra un ejemplo de tareas vinculadas a una sección.

GESTIÓN GLOBAL DE DUMBOS								
Sección: SECCION DE DESARROLLO								
Tareas Pendientes		<a href="#">Tareas Resueltas</a>						
» Atender tareas pendientes								
	Estado	Tarea	Aplicación	Asunto	Usuario	Informático	Creada	
[1]		244	SUM	<a href="#">Prueba anular</a>	<a href="mailto:elenar@um.es">elenar@um.es</a>	<a href="mailto:jjimerono@um.es">jjimerono@um.es</a>	12/07/2006	
[2]		239	SUM	<a href="#">SUB: Otra mas</a>	<a href="mailto:jjimerono@um.es">jjimerono@um.es</a>	<a href="mailto:jjimerono@um.es">jjimerono@um.es</a>	06/07/2006	
[3]		229	SUM	<a href="#">Esta si</a>	<a href="mailto:jjimerono@um.es">jjimerono@um.es</a>	<a href="mailto:jjimerono@um.es">jjimerono@um.es</a>	06/07/2006	
[4]		205	APSUBPR	<a href="#">Super Prueba</a>	<a href="mailto:jjimerono@um.es">jjimerono@um.es</a>	<a href="mailto:jjimerono@um.es">jjimerono@um.es</a>	16/06/2006	
<a href="#">Incidencia</a>		<a href="#">Petición</a>						

**Figura 44.** Tareas vinculadas a una sección.

### 8.2.2. Gestión de Dumbo por aplicaciones

La gestión por aplicaciones, permite ver y gestionar tareas vinculadas a determinadas aplicaciones. Desde el listado podemos también ver los informáticos vinculados a cada aplicación. En la figura 45 se muestra un ejemplo de gestión de Dumbo por aplicaciones

Gestión Global de Dumbos Por Aplicaciones	
MAESTRO: GESTIONA DE CUALQUIER APLICACIÓN	
<a href="#">APLICACION</a>	<a href="#">Informáticos</a>
<a href="#">APLICACION PUN</a>	<a href="#">Informáticos</a>
<a href="#">APLICACION SUBPROYECTO</a>	<a href="#">Informáticos</a>
<a href="#">ARCON</a>	<a href="#">Informáticos</a>
<a href="#">CURIE</a>	<a href="#">Informáticos</a>
<a href="#">ERASMUS</a>	<a href="#">Informáticos</a>
<a href="#">FIREWALL (PRUEBA UPCT)</a>	<a href="#">Informáticos</a>
<a href="#">GESTIÓN DE RECURSOS HUMANOS</a>	<a href="#">Informáticos</a>
<a href="#">ILA</a>	<a href="#">Informáticos</a>
<a href="#">JUSTO</a>	<a href="#">Informáticos</a>
<a href="#">PROGRAMA MOVILIDAD ESTUDIANTES ISEP</a>	<a href="#">Informáticos</a>
<a href="#">PROGRAMA MOVILIDAD NACIONAL</a>	<a href="#">Informáticos</a>
<a href="#">PUBLICACIONES</a>	<a href="#">Informáticos</a>
<a href="#">SUMA</a>	<a href="#">Informáticos</a>

**Figura 45.** Gestión por Aplicación.

Una vez seleccionada la aplicación, podemos obtener la lista de tareas vinculadas a la misma, como se muestra en la figura 46.

GESTIÓN GLOBAL DE DUMBOS							
Aplicación: SUMA							
Tareas Pendientes		Tareas Resueltas					
» Atender tareas pendientes							
	Estado	Tarea	Aplicación	Asunto	Usuario	Informático	Creada
[1]		244	SUM	<a href="#">Prueba anular</a>	<a href="mailto:elenar@um.es">elenar@um.es</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	12/07/2006
[2]		239	SUM	<a href="#">SUB: Otra mas</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	06/07/2006
[3]		229	SUM	<a href="#">Esta si</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	06/07/2006
[4]		196	SUM	<a href="#">Otra Prueba</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	<a href="mailto:ijimerono@um.es">ijimerono@um.es</a>	14/06/2006
Incidencia		Petición					

**Figura 46.** Tareas vinculadas a una aplicación.

### 8.2.3. Gestión de Dumbo por informáticos

La gestión por informáticos, permite ver y gestionar tareas vinculadas a determinados informáticos. En la figura 47 se muestra un ejemplo de la gestión de Dumbo por informáticos.

**Gestión Global de Dumbos Por Informáticos**

**INFORMÁTICOS QUE PUEDE GESTIONAR**

**Informáticos Por Secciones**

**SECCIÓN:[DES] SECCION DE DESARROLLO**

[mccm@um.es](mailto:mccm@um.es) [CARCELES MARTINEZ, MARIA CARMEN](#)  
[ic@um.es](mailto:ic@um.es) [CARRION PLAZA, JOSE CARLOS](#)  
[juanio@um.es](mailto:juanio@um.es) [GONZALEZ SANCHEZ, JUAN JOSE](#)  
[jjmerono@um.es](mailto:jjmerono@um.es) [MEROÑO SANCHEZ, JUAN JOSE \(Responsable\)](#)  
[jamolina@um.es](mailto:jamolina@um.es) [MOLINA ESPIN, JOSE ANTONIO](#)  
[inl@um.es](mailto:inl@um.es) [NADAL LORENTE, JOSE MARIA](#)  
[jdpa@um.es](mailto:jdpa@um.es) [PEREZ DEL AGUILA, JUAN DAVID](#)  
[frd@um.es](mailto:frd@um.es) [ROBLES DIAZ, FRANCISCO](#)  
[sabina@um.es](mailto:sabina@um.es) [RRRRRRRRRRRRRRR, ANA SABINA DEL](#)  
[vrp@um.es](mailto:vrp@um.es) [RUIZ PAREDES, VIRGINIA](#)  
[SANCHEZ MARTINEZ, ROSA ANA](#)

**Informáticos Por Aplicaciones**

**APLICACIÓN:[APL] APLICACION**

[mccm@um.es](mailto:mccm@um.es) [CARCELES MARTINEZ, MARIA CARMEN](#) (Responsable)  
[juanio@um.es](mailto:juanio@um.es) [GONZALEZ SANCHEZ, JUAN JOSE](#)  
[hhidalgo@um.es](mailto:hhidalgo@um.es) [HIDALGO CESPEDES, JOSE FRANCISCO](#) (Responsable)  
[jjmerono@um.es](mailto:jjmerono@um.es) [MEROÑO SANCHEZ, JUAN JOSE](#) (Responsable)  
[jdpa@um.es](mailto:jdpa@um.es) [PEREZ DEL AGUILA, JUAN DAVID](#)  
[SANCHEZ MARTINEZ, ROSA ANA](#) (Responsable)

**Figura 47.** Gestión por informáticos.

Una vez seleccionado el informático, podemos obtener la lista de tareas vinculadas al mismo como se observa en la figura 48.

**GESTIÓN GLOBAL DE DUMBOS**  
 Informático: [mccm@um.es](mailto:mccm@um.es)

Tareas Pendientes [Tareas Resueltas](#)

» Atender tareas pendientes

	Estado	Tarea	Aplicación	Asunto	Usuario	Informático	Creada	
[1]		202	CUR	<a href="#">prueba dumbov7 no hacer caso</a>	<a href="mailto:frd@um.es">frd@um.es</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	15/06/2006	
[2]		201	ARC	<a href="#">prueba tarea programada 13/06/2006</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	15/06/2006	
[3]		199	ARC	<a href="#">prueba tarea programada 13/06/2006</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	14/06/2006	
[4]		198	ARC	<a href="#">prueba tarea programada 13/06/2006</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	14/06/2006	

[Incidencia](#) [Petición](#)

**Figura 48.** Tareas vinculadas a un informático.

### 8.3. Gestión de la cola genérica de tareas

La cola genérica de tareas contiene todas las tareas que los usuarios han solicitado y que no tienen vinculada la aplicación/servicio a la que afecta la tarea; mediante esta opción se podrán asignar las tareas al experto más adecuado previa vinculación a la aplicación/servicio. En la figura 49 se muestra un ejemplo de la gestión de la cola genérica de tareas.



» Tareas por asignar

	Estado	Tarea	Aplicación	Asunto	Usuario	Informático	Creada
[1]		228		<a href="#">Esto es incidencia</a>	<a href="mailto:elenar@um.es">elenar@um.es</a>	Por Asignar	04/07/2006
[2]		227		<a href="#">dfsdfsdf</a>	<a href="mailto:elenar@um.es">elenar@um.es</a>	Por Asignar	04/07/2006
[3]		226		<a href="#">sdfsdf</a>	<a href="mailto:elenar@um.es">elenar@um.es</a>	Por Asignar	04/07/2006
[4]		131		<a href="#">aa</a>	<a href="mailto:jimerono@um.es">jimerono@um.es</a>	Por Asignar	18/05/2006

Incidencia    Petición

**Figura 49.** Gestión de cola genérica.

## 8.4. Gestión de la cola de problemas

La cola de problemas contiene tareas que informan de problemas y que están dirigidas a los supervisores del sistema DUMBO, llamados COMITÉ DUMBO. Un problema es la causa, conocida o desconocida, que origina la ocurrencia de una o más incidencias. Un ejemplo de este tipo de gestión se muestra en la figura 50.

» Listado de problemas

	Estado	Tarea	Aplicación	Asunto	Usuario	Informático	Creada
[1]		106	APL	<a href="#">prueba solicitud problema</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	Por Asignar	09/05/2006
[2]		86	CUR	<a href="#">he detectado un problema en.....</a>	<a href="mailto:jnl@um.es">jnl@um.es</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	03/05/2006
	<input checked="" type="checkbox"/>	88	CUR	<a href="#">asunto</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	03/05/2006
		87	CUR	<a href="#">asunto</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	<a href="mailto:mccm@um.es">mccm@um.es</a>	03/05/2006

Incidencia    Petición    Problema

**Figura 50.** Gestión de cola de problemas.

## 8.5. Dumbo rápido

Los dumbos rápidos son tareas resueltas de manera al mismo tiempo que son planteadas, suele ser el caso de tareas planteadas y resueltas durante una llamada telefónica.


El aspecto del formulario de solicitud de dumbos rápidos se observa en la figura 51

## 8.6. Petición programada

Las peticiones programadas son tareas que generan peticiones de forma automática, para cubrir aquellas actuaciones que son susceptibles de realizarse de manera repetitiva en un determinado periodo. En la figura 52 se muestra un ejemplo de petición programada.

» Nuevo Dumbo Rápido (\*) campos obligatorios.

**Origen/Destino**


Solicitante(\*):   Informático:   
 JUAN JOSE MEROÑO SANCHEZ

**Datos Tarea**

Asunto(\*):

Texto Tarea(\*):

Texto Solución(\*):

Aplicación(\*):  

Tiempo Empleado:  Horas  Minutos

Email Copia:



Tipo Tarea:

**Enviar**

Figura 51. Dumbo rápido.

» Nueva Petición Programada (\*) campos obligatorios.


**Origen/Destino**


Solicitante(\*):   Informático:    
 JUAN JOSE MEROÑO SANCHEZ

**Datos Tarea**

Asunto(\*):

Texto Tarea(\*):

Fecha Max. Resol.:  

Aplicación(\*):  

Urgencia:

Email Copia:



Ext.Contacto(\*):

Ubicación Tarea:

Anexos: 

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

**Temporización**

Fecha/Hora Inicio(\*):   Fecha/Hora Fin(\*):  

Intervalo(\*):  Horas Fin de Semana(\*):

**Enviar**

Figura 52. Petición Programada.

## 8.7. Administración del sistema

Permite la actualización de la información relativa a la estructura base del sistema, tal como secciones, aplicaciones, informáticos, categorías de resolución, tipos de coste, sistemas, zonas etc. En las figuras 53 y 54 se muestran un listado de sistemas y un detalle de un nuevo sistema, respectivamente.



Código	Anagrama	Descripción	URL
SIN	SISTEMASIN	<a href="#">Servicio de Informática</a>	http://dumbo.upct.es

**Figura 53.** Listado de sistemas.



**Datos del sistema**

Código (\*):

Anagrama (\*):

Descripción (\*):

URL:

(URL obligatoria si el sistema es externo)

Tipo (\*):

**Maestro del sistema (Sólo si el sistema es interno)**

Identificador:

**Figura 54.** Nuevo sistema.

## 8.8. Zonas

El módulo de administración de zonas de DUMBO permite gestionar las zonas de nuestro sistema. En la figura 55 se muestra un listado de zonas y en la figura 56 un ejemplo de una nueva zona.

Administración | Gestión | Atender | Solicitar | Solicitadas | Métricas | Notificaciones | Alertas | Inicio

> Dumbo ATICA > Administración > Zonas

JUAN JOSE MEROÑO SANCHEZ(jjmerono@um.es)

»Listado de Zonas [Insertar Zona](#)

Código	Descripción				
1	ESPINARDO NORTE				
2	ESPINARDO CENTRO-ESTE				
3	ESPINARDO CENTRO-OESTE				
4	ESPINARDO SUR-OESTE				
5	ESPINARDO SUR-ESTE				
6	ESPINARDO OESTE				
7	MURCIA NORTE				
8	MURCIA CENTRO				
9	MURCIA SUR				
10	MURCIA EL PALMAR				
11	CARTAGENA				

Figura 55. Listado de zonas.

»Nueva zona

**Datos de la zona (\*)**

Código :

Descripción :

**Vincular informaticos**

Identificador:  **Añadir**

Informaticos (Apellidos, Nombre):

<-- Añada aquí los informaticos -->

**Eliminar**

**Vincular edificios**

ZONAS :

Edificios asociados a la zona seleccionada:

- Edificio C Complejo de Espinardo.
- Edificio D Complejo de Espinardo
- CEBAS Espinardo
- Nave del Animalario
- Animalario. Servicio de Animales de Laboratorio
- Facultad de Medicina
- Almacén (CARM)
- Edificio Carpintería Issorm
- Centro Social Universitario
- Aulario Norte

Edificios añadidos a la zona:

<-- No se han añadido edificios -->

**Añadir** **Eliminar**

(\*) campos obligatorios.

**Aceptar** **Cancelar**

Figura 56. Nueva zona.

## 8.9. Consultar/Modificar informáticos

»Vincular informáticos a "ZONA 12"

Id	Apellidos, Nombre	Email	Tlf
34821340	<a href="#">MEROÑO SANCHEZ, JUAN JOSE</a>		

**Vincular informático**

Identificador:

Informaticos (Apellidos, Nombre):

[1] CARCELES MARTINEZ, MARIA CARMEN

Figura 57. Consultar/ Modificar Informáticos.

## 8.10. Consultar/Modificar edificios

»Vincular edificios a "ZONA 12"

Id	Descripción
028	Facultad de Matemáticas y Aulario General

**Vincular edificios**

ZONAS :

Edificios asociados a la zona seleccionada:

- Escuela de Negocios, Fundación Universidad Empresa
- Edificio Luis Vives
- Decanato de Psicología y Salón de Actos L. Vives
- Aulario Giner de los Ríos
- Facultad de Educación
- Facultad de Ciencias del Trabajo y E.U. de Trabajo Social

Edificios añadidos a la zona "ZONA 12" :

Hospital Virgen de la Arrixaca  
Pabellón Docente Arrixaca  
Centro Integrado de Investigación Biomédica (CEIB), PROYECTO

Figura 58. Consultar/ Modificar edificios.

## 8.11. Formularios

El módulo de administración de formularios de Dumbo permite construir, modificar o eliminar los distintos formularios de nuestro sistema. Estos formularios se emplean para solicitar tareas desde fuera de la aplicación Dumbo. En la figura 59 se muestra un ejemplo de formulario.

**Datos Formulario**

*Id(\*)*:

*URL Formulario*:

*Titulo(\*)*:

*Subtítulo(\*)*:

*Activo(\*)*:

**Datos Tarea**

*Email*:

*Informático*:

*Aplicación (\*)*:

*Asunto*:

*Texto Tarea*:

*Tipo de Tarea*:

*Urgencia*:

*Email Copia*:

*Tlfno. Contacto*:

*Ubicación Tarea*:

*Anexos*:

**Enviar**

**Figura 59.** Formularios.

## 8.12. Alias

El módulo de administración de alias de DUMBO permite crear distintos alias para el sistema. Al entrar a la pantalla obtenemos el listado de todos los alias, como se muestra en la figura 60.

Listado de Alias Para Direcciones de Correo			<a href="#">Nuevo Alias</a>	
Alias	Dni	Nombre		
dmaster	34821340	MEROÑO SANCHEZ, JUAN JOSE	Eliminar	Modificar
junbio	48505775	LOZANO RUIZ, MANUEL	Eliminar	Modificar
junder	48518028	NAVARRO ALIAGA, ERNESTO	Eliminar	Modificar
juneco	73574768	IRANZO PONCE, TAMARA	Eliminar	Modificar
junedu	47722247	LORENTE GARCIA, JOSE CARLOS	Eliminar	Modificar
junlet	48551491	COLL CAMARA, JESUS	Eliminar	Modificar
junlv	48399453	GARCIA PINO, DAVID	Eliminar	Modificar
junmat	48543960	EGEA MUÑOZ, DAVID	Eliminar	Modificar
junmed	23027116	SORIANO HERNANDEZ, ANTONIO	Eliminar	Modificar
junmerm	48396461	FERNANDEZ RUIZ, RAFAEL GUILLERMO	Eliminar	Modificar
junmert	77717149	BENITEZ VEGA, OSCAR	Eliminar	Modificar
junqui	23288112	LOPEZ VALDES, CARLOS	Eliminar	Modificar
junvet	20467522	MORA BERNABE, FERNANDO	Eliminar	Modificar

Figura 60. Alias.

## 8.13. Estadísticas

Sistema Dumbo integra un módulo de estadísticas, tanto personales como estratificadas por aplicaciones/servicios y secciones, y también globales.

## 8.14. Alertas

Dumbo dispone de un sistema de alertas a 3 niveles: experto, gestor responsable y supervisor, que avisa de tareas que están en estado atípico, bien por que deberían estar resueltas y no lo están, bien porque falta cumplimentar alguna información de especial relevancia.

## 8.15. Notificaciones

Dumbo contempla un sistema de notificaciones por e-mail, tanto a experto informático como a usuario, cuando se producen eventos de relevancia en la vida de la solicitud de una tarea que les afecta.

## **9.- ANEXO II**



## **9. ANEXO II**

### **9.1 Comparativa entre herramientas de inventario**

Para facilitar las tareas de gestión de inventario de equipamiento de ordenadores, impresoras y otros dispositivos se precisa un gestor de inventario.

Para ello, ha sido necesario realizar un estudio comparativo entre las distintas herramientas existentes en el mercado, que permitan inventariar dichos equipos. En el presente documento se recogen los resultados obtenidos del estudio.

### **9.2. Listado de herramientas**

Como punto de partida se han preseleccionado las siguientes herramientas para la gestión de inventario:

- Open Computers and Software (OCS) Inventory NG
- zCI Computer Inventory System
- NetInvent
- Windows Inventory/Open-Audit
- Computer Inventory Management System
- h-inventory
- HW Audit System
- phpMyInventory
- os:inventory
- PoxPos
- Feta
- PHP Inventory Manager

- SpiceWorks

Se ha realizado una criba del listado anterior de forma que se cumplieran las siguientes premisas:

- a) Que sean de licencia pública, lo que permitirá hacerles cambios para adaptarlas a las necesidades de la Confederación.
- b) Que sea una herramienta con cierta madurez.
- c) Que sea una herramienta que no haya dejado de mantenerse.

Partiendo de estas premisas, el listado de herramientas se ha reducido a solamente dos, que son:

- Open Computers and Software (OCS) Inventory NG
- zCI Computer Inventory System

Ambas basan su arquitectura en un agente (pieza software) que se ejecuta en los equipos cliente y un servidor de gestión, que centraliza los resultados del inventario.

### **8.3. OCS Inventory NG**

Como se ha indicado anteriormente, la arquitectura del sistema se basa en un agente software que se instala en los equipos a inventariar y un servidor de gestión que recoge la información de los agentes.

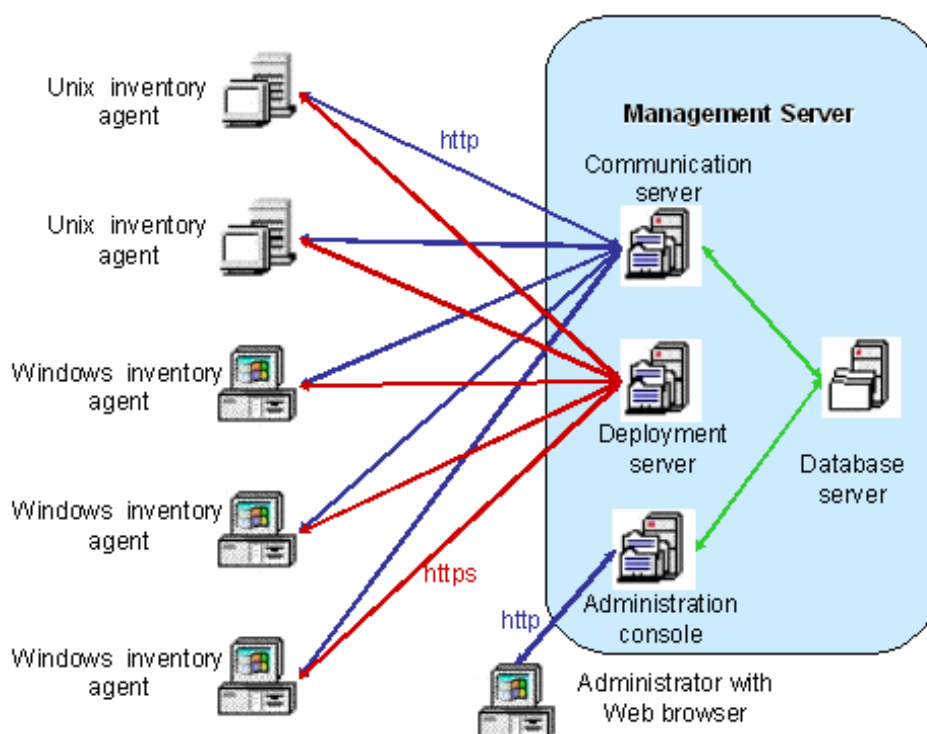
La comunicación entre los agentes y el servidor se realiza utilizando los protocolos HTTP/HTTPS. Los datos se formatean en XML comprimido para reducir la carga de tráfico por la red.

Existe una herramienta para instalarlos a través de scripts de conexión (login) o de Directorio Activo bajo Windows. Bajo Linux hay que instalarlos manualmente.

El servidor de gestión está formado por 4 componentes:

- servidor de base de datos
- servidor de comunicación, que maneja la comunicación HTTP entre el servidor de base de datos y los agentes.
- servidor de despliegue, que almacena la configuración de los paquetes para instalar.
- consola de administración, que permite administrar el sistema a través de un navegador

En la figura 1 se muestra la arquitectura de la herramienta.



**Figura 61.** Arquitectura OCS Inventory NG

La principal ventaja de esta herramienta es la escalabilidad. Los 4 componentes de la arquitectura se pueden alojar en un único servidor o en varios para permitir el balanceo de carga. Para más de 10 mil equipos inventariados es mejor utilizar al menos dos servidores distintos, uno para el servidor de la base de datos + el servidor de comunicación y otro para réplica de la base de datos + consola de administración + servidor de despliegue.

Otra ventaja de esta herramienta es que permite el despliegue de aplicaciones, es decir, crear paquetes de instalación de software que se distribuirán a los ordenadores para ser instalados de forma desatendida.

Está construida utilizando herramientas de libre distribución ampliamente utilizadas como son:

- Servidor de base de datos: MySQL 4.1 o superior
- Servidor de comunicación: escrito en PERL. Necesita Apache 1.3.x/2.x
- Servidor de despliegue: necesita un servidor web con SSL habilitado
- Consola de administración: escrita en PHP 4.1. Necesita Apache 1.3.x/2.x
- Agente Windows: escrito en C++
- Agente Linux: escrito en PERL

El agente se ha probado en los siguientes sistemas operativos:

- Windows 95 con DCOM95 (o IE 4 o superior)
- Windows Windows 98
- Windows Windows 98 Second Edition

- Windows Windows NT4 Workstation con IE 4 o superior
- Windows Windows NT4 Server con IE 4 o superior
- Windows Windows 2000 Professional
- Windows Windows 2000 Server/Advanced Server
- Windows Windows XP Home Edition
- Windows Windows XP Professional Edition
- Windows Windows Server 2003
- Windows Windows Vista
- Linux Centos Linux
- Linux Debian Linux
- Linux Fedora Core Linux
- Linux Gentoo Linux
- Linux Knoppix Linux
- Linux Mandriva Linux
- Linux RedHat Linux
- Linux Slackware Linux
- Linux SuSE Linux
- Linux Trustix Linux
- Linux Ubuntu Linux

El servidor de gestión se ha probado en las siguientes plataformas:

- Windows 2000 Professional
- Windows Windows 2000 Server
- Windows Windows XP Professional Edition
- Windows Windows Server 2003
- Linux Centos Linux
- Linux Debian Linux
- Linux Fedora Core Linux
- Linux Mandriva Linux
- Linux RedHat Linux
- Linux SuSE Linux
- Linux Ubuntu Linux

OCS Inventory NG además de realizar un inventario hardware permite realizar un inventario software. En la figura 2 se muestran los distintos elementos que es capaz de inventariar.

<b>BIOS</b>		System serial number, System manufacturer, System model, BIOS manufacturer, BIOS version, BIOS date.
<b>Processors</b>		Type (Pentium with MXX, Penitum II, Penitum III, Pentium IV, Celeron, Athlon, Duron...), Processor speed, Number of Processors.
<b>Memory slot arrays</b>		Caption and description, Capacity in MB, Purpose (system memory, flash memory...), Type of memory (SDRAM, DDR...), Speed in MHz, Slot number.
<b>Total physical memory</b>		in MB
<b>Total paging file or swap size</b>		in MB
<b>Input devices</b>		Type (keyboard or pointing), Manufacturer, Caption, Description, Interface used (PS/2, USB...).
<b>System Ports</b>		Type (serial or parallel), Name, caption and description.
<b>System Slots</b>		Name, Description, Designation (AGP1, PCI1, PCI2, ISA1...).
<b>System Controllers</b>		Manufacturer, Name, Type (Floppy, IDE, SCSI, USB, PCMCIA, IEEE1394, Infrared).
<b>Storage peripherals</b>		Manufacturer, Model, Description, Type (Floppy, Hard disk, CD-Rom, Removable, Tape...), Size in MB.
<b>Logical drives / partitions</b>		Logical drive letter, Type (removable, hard drive, cd-rom, network, RAM...), File system (FAT, FAT32, NTFS...), Total size in MB, Free space in MB.
<b>Sound devices</b>		Manufacturer, Name, Description.
<b>Video adapters</b>		Name, Chipset, Memory in MB, Screen resolution.
<b>Display monitors</b>		Manufacturer, Caption, Description, Type, Serial number.
<b>Telephony modems</b>		Name, Model, Description, Type (internal, external...).
<b>Network Adapters</b>		Description, Type (Dialup, Ethernet, Token Ring, ATM...), Speed (in Mb/s or Gb/s), MAC Address, IP Address, IP Network Mask, IP gateway, DHCP server used.
<b>Printers</b>		Name, Driver, Connection port.
<b>Operating System</b>		Name (Linux, Windows 95/98/Me/NT Workstation/NT Server/2000 Professional/2000 Server...), Version (4.0.1381 for NT4, 4.10.2222 for 98 SE...), Comments (Operating System Release for Windows 9X, Service Packs for NT/2000/XP), Registered company, Registered owner, Registered product ID.
<b>Software</b>		extracted from the registry as shown in the "Add/Remove software" control panel applet: Name, Publisher, Version.
<b>Registry values</b>		if you want to query for some key under one of Windows registry hives.
<b>Computer description</b>		as shown in the Identification Tab of the Network Settings dialog box.

**Figura 62.** Elementos inventariables por OCS

## 8.4. zCI Computer Inventory System

Al igual que OCS Inventory NG, la arquitectura se basa en un agente que se instala en los ordenadores y un servidor que recoge la información.

Respecto a las herramientas sobre las que está construido, en el caso de los clientes están disponibles para MS Windows (basados en Jscript) y Linux (basados en Java)

La parte del servidor se puede instalar en cualquier plataforma hardware y en cualquier sistema operativo. Necesita un servidor web que soporte PHP, PHP versión 4.1 o superior y una base de datos. Se recomienda MySQL versión 4.1 o superior.

The screenshot shows a web browser window titled "Computer Details - Microsoft Internet Explorer". The address bar shows the URL: `http://localhost/zc/frontend/computer_details.php?compid=1`. The page content includes a search bar with the text "Computer Name:" and a "SEARCH" button. Below the search bar, there is a red header with the text "ID: 3ET-MZAMRONI-1". The main content is a table with the following data:

Registration Date	2005-06-11
User Name	MZamroni-A
Network Domain	MYDOMAIN1
Computer Model	IBM 23731HX ThinkPad T41
Serial Number	99TBTXK
Ownership Status	Lease
Reserve Status	active usage, not reserved
Operating System	Microsoft Windows 2000 Professional (2004-03-16)
Processor	1. Intel(R) Pentium(R) M processor 1400MHz (1398 MHz)
Memory Size	512 MB
Display Adapter	ATI MOBILITY RADEON 7500 AGP (0x4C57)
Monitor	1. ID 7: Default Monitor (Ownership: Lease SN: KSJFSDK93284)
Floppy Drive	not exists
Harddisk	1. IC25N030ATMR04-0 (28 GB)
CD Drive	1. HL-DT-ST DVD-ROM GDR8083N
SCSI Controller	
Network Adapter	
Audio Controller	1. SoundMAX Integrated Digital Audio
Peripheral	1. ID 15: Canon BJC20ef (Ownership: Lease SN: YUEYR9328409) 2. ID 10: FX Document Centre 235 PCL 6 (Ownership: Lease SN: 112121298793) 3. ID 9: HP Laserjet 1220 (Ownership: Lease SN: 8645986897456) 4. ID 17: HP Laserjet 5000 (Ownership: Own SN: YREIUT9348093) 5. ID 11: Lexmark Z3 (Ownership: Lease SN: 768432423) 6. ID 13: Lexmark Z3 (Ownership: Lease SN: EQWE23213)

At the bottom of the table, there are three buttons: "Show History", "Show/Hide Installed Softwares", and "Show Supervising Page".

Figura 63. Elementos inventariables por zCI Computer Inventory System

## **8.5. Comparativa entre OCS Inventory System y zCI Computer Inventory System**

Como se ha indicado anteriormente, ambas herramientas basan su arquitectura en agentes instalados en los ordenadores (Windows y Linux) y un servidor en el que queda registrada la información que se recopila de los agentes.

Ambas utilizan herramientas de uso público como son servidor web Apache + PHP para la gestión y MySQL para la base de datos, si bien OCS Inventory System requiere además del intérprete de PERL ya que para acelerar la comunicación entre los agentes y el servidor se utilizan CGIs escritos en este lenguaje que quedan cargados en memoria

Ambas permiten utilizar una base de datos distinta a MySQL, como puede ser MS SQL Server o incluso ORACLE, aunque recomiendan que sea MySQL.

Están bajo licencia GNU General Public License, lo que permite modificarlas para adaptarlas a las necesidades cualquier organización.

Finalmente, las diferencias a favor de OCS Inventory System son tres:

- 1) Que permite el despliegue de aplicaciones para instalarlas en los ordenadores.
- 2) Que se menciona una escalabilidad del sistema, repartiendo la carga de los servicios en varios servidores
- 3) La comunicación entre los agentes y el servidor es más rápida al estar desarrollado el script en PERL y quedar residente.

## **10.- BIBLIOGRAFÍA**



## 10. BIBLIOGRAFÍA

- [1] DUMBO, Gestión automatizada de atención a usuarios.  
<http://www.um.es/atica/dumbo/>
- [2] OCS, Inventory NG. Herramienta de inventario.  
<http://www.ocsinventory-ng.org>
- [3] zCI Computer Inventory System. Herramienta de inventario.  
<http://zci.sourceforge.net>
- [4] Windows Inventory/Open-Audit. Herramienta de inventario.  
<http://www.open-audit.org>
- [5] h-inventory. Herramienta de inventario..  
<http://demo.h-inventory.com>
- [6] El Ciclo de Vida en Cascada. Curso de doctorado  
“El proceso software”. Jose Juan Pazos Arias.  
[http://idtv.det.uvigo.es/~jose/doctorado/proceso\\_sw/sld006.htm](http://idtv.det.uvigo.es/~jose/doctorado/proceso_sw/sld006.htm)
- [7] MyFaces. Proyecto de Apache Software Foundation  
<http://myfaces.apache.org/>
- [8] Spring. Framework de código abierto de desarrollo de aplicaciones para  
la plataforma Java.  
<http://www.springsource.org/>
- [9] Hibernate. Framework de desarrollo para desarrollar la persistencia.  
<https://www.hibernate.org/>

- [10] MÉTRICA. Versión 3. Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información.  
<http://www.csae.map.es/csi/metrica3/index.html>
- [11] Tutorial sobre integración JSF + Spring + Hibernate  
[http://www.programacion.com/tutorial/jap\\_jsfwork/](http://www.programacion.com/tutorial/jap_jsfwork/)
- [12] Java Server Faces (JSF)  
<http://java.sun.com/javaee/javaserverfaces/>
- [13] Ajax For JSF (ajax4jsf). Librería Ajax para JSF.  
<https://ajax4jsf.dev.java.net/>
- [14] Entorno de desarrollo. Eclipse Ganymede.  
<http://www.eclipse.org/ganymede/>
- [15] Spring Integration.  
<http://static.springframework.org/spring-integration/reference/htmlsingle/spring-integration-reference.html#mail>
- [16] Spring LDAP.  
<http://www.springsource.org/ldap>
- [17] Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.