

Gestión del LDAP

Memoria

Autor: F. Jorge Blanca Sánchez

Director: Jordi Ceballos Villach

Estudios: Ingeniería Técnica en Informática de Sistemas – UOC

Enero 2006


	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

AGRADECIMIENTO

Quiero agradecer la ayuda y comprensión que he recibido de Fina, mi mujer, que ha sabido conducir con gran destreza la cantidad de horas que he tenido que dedicar al reto personal que ha supuesto sacar esta carrera.

TABLA DE CONTENIDOS

• Tabla de contenidos	3
• Descripción del proyecto	4
• Objetivos	5
• Planificación temporal	6
• Plataforma de desarrollo	8
• Struts Framework	11
• Implementación del modelo de datos (LDAP)	13
• Diseño del LDAP	15
• Diseño de clases	19
El Contrlador	19
El Modelo	22
La Vista	24
• Clases de la vista	26
• Funcionalidades de la aplicación	34
• Conculsiones	42
• Líneas de futuro desarrollo	44
• Bibliografía	45

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

DESCRIPCIÓN DEL PROYECTO

El proyecto se basa en la creación de una aplicación web que permita la gestión del inventario de los servidores e impresoras de un dominio informático.

En dominios Windows 2000 & 2003, la información sobre personas, aplicaciones y recursos está contenida en el componente denominado 'Active Directory' lo cual permite la capacidad de centralizar la administración de la red con el fin de compartir recursos en un conjunto de directorios. Para ello el Active Directory trabaja y utiliza servicios estándares propios de redes IP. Es el caso de las especificaciones LDAP, DNS, DHCP y otros.

'Active Directory' utiliza un LDAP como base de datos para almacenar la información referente a los recursos que controla, en particular y en lo que afecta a este proyecto, nos interesa la información de Servidores e Impresoras. Cuando alguno de estos recursos se incorpora o se desinstala en alguno de los dominios que controla el 'Active Directory' automáticamente queda registrado en el LDAP.

La aplicación 'Gestión del LDAP' debe permitir el acceso al servicio LDAP, de forma que a través de una aplicación web podamos gestionar recursos del dominio, dando altas y bajas de servidores e impresoras en el LDAP de forma manual para aquellos recursos que no sean de dominios Windows o bien poder hacer las manipulaciones en el registro que el departamento de Sistemas estime convenientes. Además la aplicación debe permitir la consulta por de los recursos gestionados y su posible modificación.

Aunque 'Gestión LDAP' se conecta inicialmente a un LDAP por defecto de manera automática, también permite conectar a cualquier otro LDAP. Para ello debemos especificar el host y puerto por el que está escuchando, así como las credenciales de un usuario que tenga permisos para conectar.

Esta aplicación está personalizada para dos tipologías de usuario. Una denominada 'Administrador' que permite Altas, Bajas, Modificaciones y Consultas del LDAP. El otro tipo denominado 'Usuario' sólo tiene acceso a la lectura de los registros.

Además, el administrador puede modificar los perfiles de usuario del LDAP y puede dar o quitar permisos para el acceso a esta aplicación a cualquier usuario del dominio. No debemos olvidar que los usuarios también son recursos del LDAP, por ello los administradores de la aplicación sólo deben modificar un atributo para permitir como Administrador o Usuario, o no permitir el acceso a la aplicación.

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

OBJETIVOS

Con este trabajo pretendo abordar los siguientes objetivos:

1. Realizar una aplicación web basada en el paradigma de diseño MVC (Model, View, Controller) mediante el uso de un Framework.
2. Ampliar el conocimiento de acceso a otro tipo de base de datos distinto a las relacionales, concretamente a los servicios de Directorio LDAP. Estas estructuras de datos se pueden definir como bases de datos de estructura jerárquicas y optimizadas para lecturas.
3. Conocer y manejar las API's J2EE para desarrollo de aplicaciones web y acceso a recursos de servicios de directorios JNDI.
4. Además desearía profundizar en los sistemas de seguridad de los servidores de aplicaciones J2EE.
5. Conocer y ampliar las diferentes posibilidades que ofrece el mercado en cuanto a patrones de trabajo para diseño e implementación de aplicaciones basado en el modelo MVC.
6. Ampliar las posibilidades que tenemos en la actualidad para adquirir un sistema de bases de datos basados en Directorio.

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

PLANIFICACIÓN TEMPORAL

La planificación del proyecto se ha respetado escrupulosamente. Las 4 entregas que debían realizarse durante el desarrollo del trabajo se han cumplido en fechas.

Las entregas prefijadas son:

PAC 1 - Plan de trabajo	26/09/2005
Análisis – Prototipo primera versión	17/10/2005
PAC 2 – Diseño – Prototipo definitivo	07/11/2005
PAC 3 – Implementación	09/12/2005
Memoria + Presentación	09/01/2006

Primera entrega: Plan de Trabajo

Se describe el alcance del proyecto, así como los objetivos, requisitos y planificación.

Segunda entrega: Documento de análisis

Se especifican las funcionalidades de la aplicación del proyecto y una primera versión del prototipo (sólo interface gráfica).


Tercera entrega: Diseño en prototipo definitivo.

Se especifica cómo se implementan las funcionalidades descritas en la fase anterior, con sus diagramas y definición del esquema LDAP y la interface gráfica ya finalizada.

Cuarta entrega: implementación.

Debe hacerse entrega de la aplicación prácticamente finalizada. Es decir, quedan algunos detalles para acabar de retocar durante la siguiente entrega.

Quinta y última entrega: Memoria del Trabajo Final de Carrera y presentación PowerPoint. Entrega la implementación definitiva de la aplicación, así como de una memoria de la misma y una presentación.

 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

Finalmente, adjunto el diagrama Gantt donde se puede ver de forma esquemática la planificación del proyecto a lo largo del cuatrimestre.



Los documentos obtenidos durante el desarrollo del aplicativo son:

1. Pan de Trabajo: Breve descripción de trabajo y planificación temporal.
2. Análisis: Requerimientos y especificación de las funcionalidades mediante casos de uso.
3. Prototipo del aplicativo: Diseño de la interface y ejemplo de funcionalidades en HTML estático.
4. Diseño: Diagrama de clases y diagramas y especificaciones de los esquemas LDAP.
5. Implementación: Aplicación totalmente operativa.
6. Manual de instalación: Descripción detallada de los pasos a realizar para la total instalación del aplicativo.
7. Memoria del proyecto: El presente documento.
8. Presentación: complemento de la memoria destacando los puntos esenciales.

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

PLATAFORMA DE DESARROLLO

El desarrollo del aplicativo se ha realizado sobre una CPU Intel Pentium 4 1.7 GHz con 512 MB de memoria RAM aunque sería recomendable trabajar por lo menos con 1 GB.

El sistema operativo es una distribución Red Hat – Fedora Core (2.6.10-1.771_FC2) instalada con una configuración workstation.

La arquitectura de trabajo es J2EE, por lo que ha habido que instalar un JDK de Java de la versión 1.5 (*jdk-1_5_0_04-linux-i586.rpm*).

Como contenedor de servlets he utilizado Apache Tomcat por ser la implementación de referencia para Servlets de Java y tecnologías JSP. (Jakarta Tomcat versión 5.5).

Para modelizar el paradigma de desarrollo MVC y distinguir plenamente las capas utilizaré el Framework Struts. (Jakarta Struts 1.1)


El servicio de Directorios LDAP que se va a utilizar como base de datos del aplicativo será OpenLDAP por ser un desarrollo Open Source. (OpenLDAP versión 2.3.7)

El IDE para desarrollo en Java que he selecciona es la versión 3.0 del conocido software Eclipse.

Para el desarrollo de aplicaciones Tomcat y Struts utilizaré los plugins MyEclipse Enterprise WorkBench 3.8.4 y el plugin de Sysdeo para Tomcat-Eclipse V3

Para acceso a OpenLDAP v2.3.7 utilizaré la API JNDI. Ésta es la API J2EE específica para acceso servicio de Directorio.

También se ha utilizado un visualizador/editor de LDAP. Se hace imprescindible en las primeras fases de la implementación y especialmente durante las pruebas. El producto es LDAP Browser/Editor 2.8.2 beta II, y es una aplicación Open Source.

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

IMPLEMENTACION DEL MODELO MVC

Este proyecto pretende seguir las pautas que permitan un desarrollo estructurado de una aplicación, donde las verificaciones de sesión se centralizan y cada uno de los casos de uso se distingue claramente.

Con la implementación de Struts se fuerza a una arquitectura perfectamente estructurada que divide perfectamente la lógica de negocio (Model), la presentación (View) y el control de flujo (Controller).

El paradigma MVC consiste en dividir las aplicaciones en tres partes:

El controlador, encargado de redirigir o asignar una aplicación a cada petición

El modelo es la aplicación que corresponde a la petición, es la lógica de negocio

La vista es la devolución de los resultados

La ventaja y el éxito de este modelo de 'patrón' radica en la perfecta separación entre las capas de presentación, lógica de negocio y acceso a los datos, fundamental para el desarrollo de arquitecturas consistentes, reutilizables y fácilmente mantenibles. Además nos permite aplicar opciones como el multi-idioma, distintos diseños de presentación, etc. sin alterar la lógica de negocio.

Una de los objetivos de este proyecto es la selección de un framework para aplicaciones web J2EE que implemente el modelo MVC.

El framework de presentación Barracuda (<http://barracuda.enhydra.org>) es un tipo de arquitectura similar a Struts aunque parece dar un paso más y proveer un mecanismo de notificación de eventos.

El proyecto Cocoon (<http://xml.apache.org/cocoon>) es un open source desarrollado en la Apache Software Foundation. La finalidad de Cocoon es ayudar a la separación del estilo, lógica y gestión de funciones para web sites basadas en XML

Maverick MVC Framework (<http://mav.sourceforge.net>) ofrece la posibilidad de renderizar vistas usando directamente JSP, Velocity o XSLT (Extensible Stylesheet Language Transformations)

Jakarta Turbine (<http://jakarta.apache.org/turbine>) es un proyecto open source de Jakarta. Actualmente no existe mucha documentación, sin embargo es muy similar a Struts con algunas diferencias.

Finalmente, Struts Framework (<http://struts.apache.org>) es el seleccionado para este proyecto por ser un patrón basado en tecnologías estándar como Java Servlets, JavaBeans, ResourceBundle, y XML. Struts provee su propio componente Controlador y se integra con otras tecnologías para proporcionar el Modelo y la Vista. Para el modelo, Struts interactúa con tecnologías de acceso a datos como JDBC, EJB y paquetes de terceros como Hibernate, iBATIS, o Object Relational Bridge. Para la

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

Vista, Struts trabaja bien con JSP, incluyendo JSTL y JSF, así como plantillas Velocity y otros sistemas de presentación.

En ámbito de trabajo para el desarrollo de la aplicación consiste en:

- Plataforma Java JDK 5.0 (<http://java.sun.com/j2se/1.5.0>)
- Plataforma Java J2EE 1.4 (<http://java.sun.com/j2ee/1.4>)
- Tomcat 5.5 (<http://tomcat.apache.org>). Se trata del contenedor de servlets usado como implementación oficial de Referencia para tecnologías Java Servlet y JavaServer Pages.
- Eclipse 3.0 (<http://www.eclipse.org>) como plataforma IDE para desarrollo.
- MyEclipse 3.8.4 (www.myeclipseide.com). Software comercial que se integra como un plugin de Eclipse que permite, entre otras características, desarrollar con aplicaciones bajo el patrón Struts Framework.
- Sysdeo Eclipse Tomcat Launcher plugin versión 3.1.0 Beta (<http://www.sysdeo.com/eclipse/tomcatplugin>) permite parar y arrancar Tomcat desde eclipse, crea proyectos war.

STRUTS FRAMEWORK

Diagrama estático

Como podemos ver en el diagrama de clases de Struts, los componentes que debe añadir el desarrollador son los 'userActions' y los 'userActionForms'. Éstos son clases de java que extienden respectivamente a Action y ActionForm. Los JavaServer Pages (JSP's) son el componente de presentación. Por otra parte, la mayoría de las clases de controlador están parametrizadas de forma que el fichero struts-config.xml se convierte en el árbitro que regula y controla las peticiones y accesos.

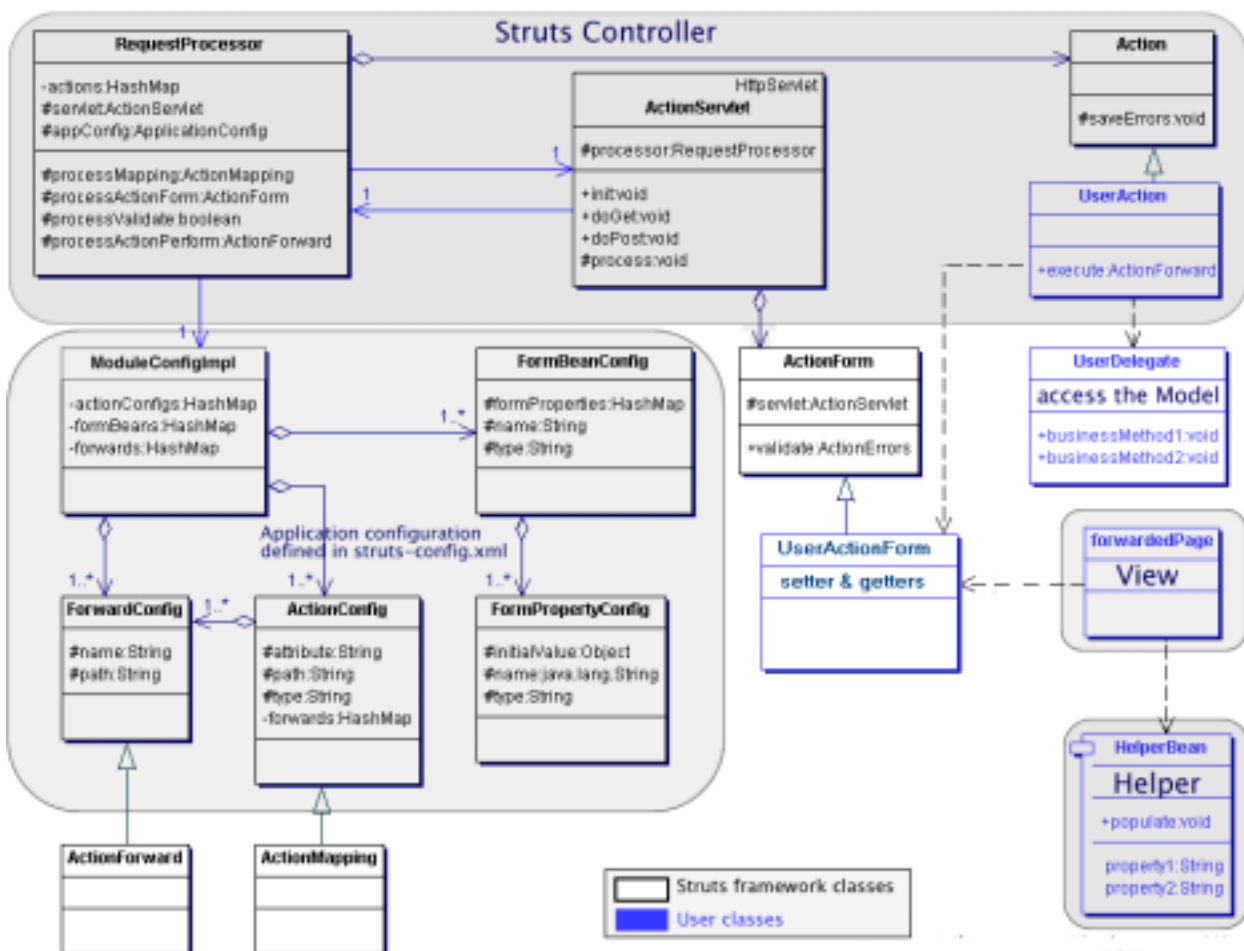
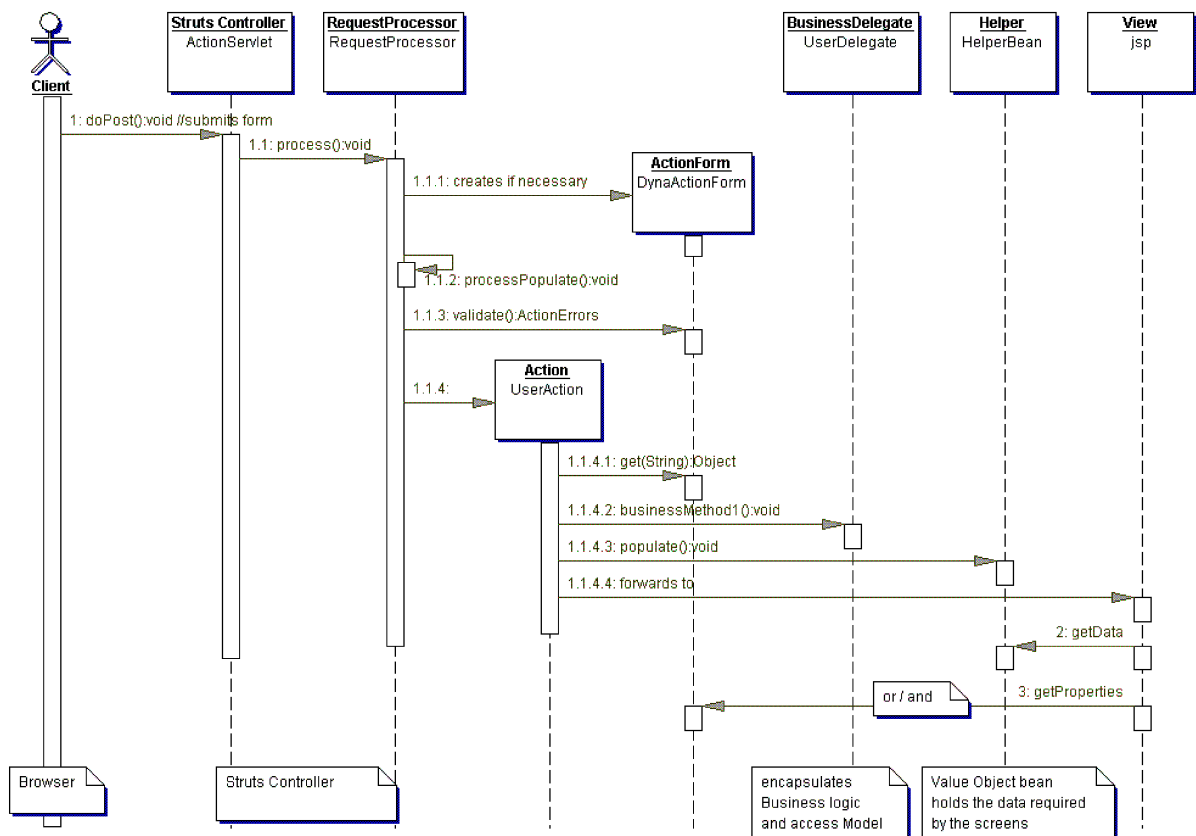


Diagrama de secuencia

- 1 El cliente submite un formulario HTML, se llama al método doPost para que el HTTPServlet *ActionServlet* atienda la petición POST
 - 1.1 El Controlador Struts *ActionServlet* delega a *RequestProcessor* el proceso de la petición y realiza las siguientes acciones
 - 1.1.1 Devuelve el bean *ActionForm* asociado con el mapeo y creación si es preciso
 - 1.1.2 Cumplimenta el *ActionForm* con los campos de entrada del formulario
 - 1.1.3 Valora los valores de los campos de entrada y si la validación encuentra errores, crea los mensajes de error correspondientes.
 - 1.1.4 Toma una instancia de *UserAction* para procesar la petición, llamando al método sobrescrito de *execute* de *UserAction*
 - 1.1.4.1 Recupera los datos del bean *UserActionForm* por medio de los métodos *getProperties*
 - 1.1.4.2 Llamada a servicio de negocio a través de *BusinessDelegate*
 - 1.1.4.3 Rellena los valores de l objeto bean
 - 1.1.4.4 Llama a JSP especificado como destino en *struts-config.xml*
- 2 La página llamada obtiene datos del objeto *HelperBean*
- 3 y / o del bean *ActionForm*



	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

IMPLEMENTACION DEL MODELO DE DATOS (LDAP)

La inmensa mayoría de las aplicaciones de gestión utilizan bases de datos relacionales. El API JDBC ofrece a los desarrolladores de programas un modelo de conectar datos relacionales desde el interior del código Java.

Sin embargo, la característica que hace más singular este proyecto reside en el modelo de datos. Como se ha planteado en la presentación, se va a trabajar con un tipo de base de datos especial llamado LDAP.

LDAP (Lightweight Directory Access Protocol) y los servicios de directorio en general no son más que un tipo especial de bases de datos que proporcionan rápido acceso a sus almacenes de datos. Estas bases de datos almacenan la información de forma jerárquica y están optimizadas para lectura, en contraste con las bases de datos consideradas normalmente que se organizan de forma relacional como es el caso de Oracle, Microsoft SQL Server, MySQL, IBM DB2, CloudScape, etc.

En redes Microsoft Windows 2000 y Windows 2003, el Active Directory es el servicio de directorio que administra, controla y organiza centralmente los recursos de la red. Active Directory hace que la topología física de red y los protocolos pasen desapercibidos, de forma que un usuario de una red puede tener acceso a cualquier recurso sin saber dónde está el recurso o cómo está conectado físicamente a la red. Ejemplos de este tipo de recursos son las estaciones de trabajo (workstations), los servidores y las impresoras. Active Directory ofrece una amplia compatibilidad con las normas y protocolos existentes. En la siguiente tabla podemos se describen tecnologías compatibles con AD y su función:

Tecnología	Descripción
Protocolo de configuración dinámica de host (DHCP)	Admon. Direcciones de red
Protocolo de actualización dinámica (DNS)	Admon. Espacios de nombres de red
Protocolo de tiempo de red simple (SNTP)	Admon. De tiempo distribuido
Protocolo ligero de acceso al directorio (LDAP)	Acceso a directorios
Formato de intercambio de datos LDAP (LDIF)	Sincronización de directorios
Kerberos v5	Autenticación

La aplicación objeto del presente proyecto utiliza la información depositada en el LDAP para inventariar y manipular la información contenida. De forma que supone una herramienta adicional al propio LDAP. Hemos visto que AD de Microsoft mantiene en su LDAP la información necesaria de impresoras y *hosts* suficiente para permitir un control descentralizado de toda la red.

La aplicación que se presenta en este trabajo no utiliza el LDAP de Microsoft, de hecho, está preparada para trabajar independientemente de cualquier implementación mientras sea compatible con la especificación LDAP v3. Lo que realmente la hace más interesante con AD es que los recursos de *hosts* e impresoras quedan automáticamente recogidos por el LDAP en el momento que se instalan en la red y por lo tanto la información del LDAP del Active Directory siempre está actualizada.

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

La mayoría de servidores LDAP disponibles en el mercado están bien equipados con mecanismos que pueden ser usados para comunicar de forma segura, entre ellos figuran:

- IBM Tivoli Directory Server (<http://www-306.ibm.com/software/tivoli/resource-center/security/code-directory-server.jsp>) es un servidor de directorios para empresas para intranets corporativas e Internet.
- Sun Java System Directory Server (http://www.sun.com/software/products/directory_srvr_ee/dir_srvr/index.xml) es el servidor de Sun que recibe la herencia de Netscape LDAP e iPlanet LDAP tras las compras y fusiones de empresas.
- Oracle Internet Directory (<http://www.oracle.com/technology/products/oid/index.html>) es el repositorio para la gestión de identificación de los productos Oracle
- Novel eDirectory/NDS (<http://www.novell.com/products/edirectory>) es el servidor de directorios de Novel.
- OpenLDAP (<http://www.openldap.org>) es una suite de aplicaciones y herramientas de desarrollo 'open source'. El código deriva de la distribución LDAP v3.3. de la Universidad de Michigan y está desarrollado bajo una licencia BSD.

Para el desarrollo del aplicativo utilizaré OpenLDAP como servidor de Directorio de desarrollo open source. Por ello, se migrarán los datos de un servidor Microsoft Active Directory que contiene la información de servidores e impresoras de un dominio y los traspasaré a OpenLDAP.

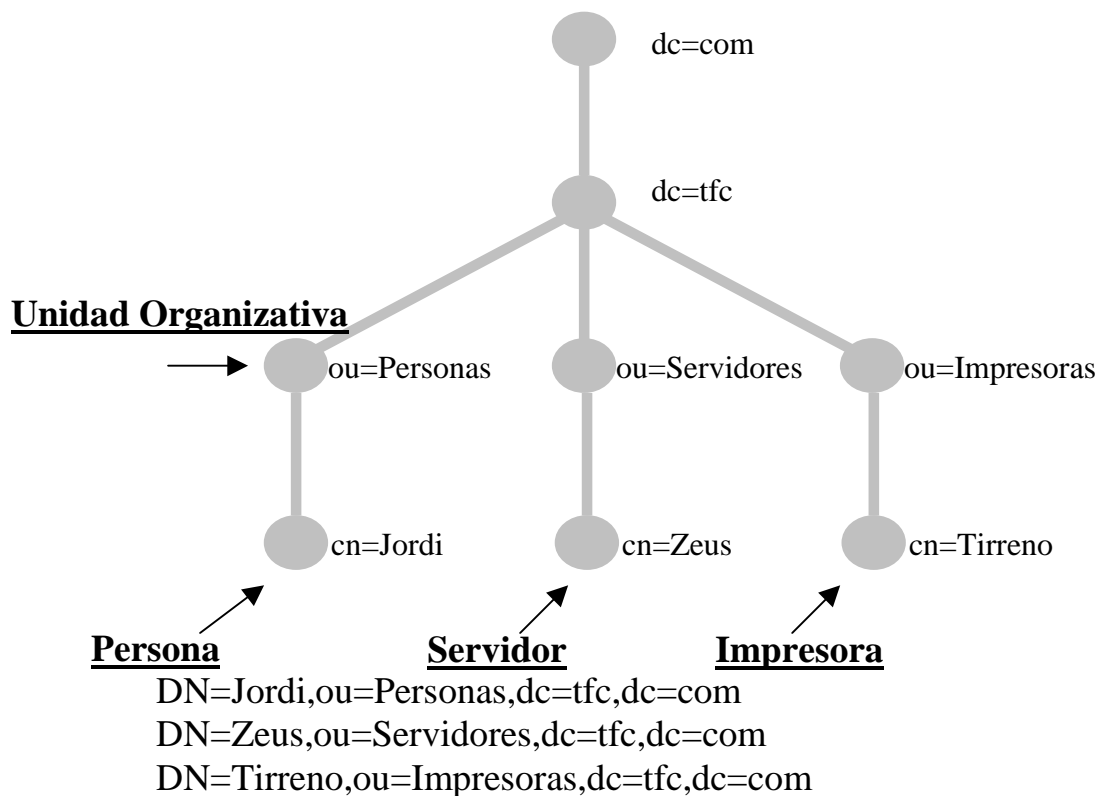
DISEÑO DEL LDAP

EL fichero de configuración de openLDAP es `/usr/local/etc/openldap/slapd.conf`. Este fichero está estructurado en dos partes, una primera con las opciones globales de LDAP y otra con las definiciones de la base de datos (directorio).

Definición de la base de datos:

Database	Bdb	El tipo de base de datos a utilizar
Suffix	"dc=tfc,dc=com"	Sufijo del Distinguished Name (DN) de las consultas que serán pasadas al LDAP.
Rootdn	"cn=Manager,dc=tfc,dc=com"	Distinguished Name (DN) que no está sujeto a los accesos de control ni límites administrativos del LDAP
Rootpw	secret	Contraseña para el rootdn

Una vez definido el tipo de base de datos, el sufijo de todos sus elementos y un administrador, ya se puede abrir el servicio LDAP y empezar a definir su estructura jerárquica. Esta organización nos permite ordenar la información en subdirectorios que contienen la información muy específica.



 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

El modelo de información LDAP está basado en entradas. Una entrada es una colección de atributos que tienen un único y global Nombre Distintivo (DN).

El DN se utiliza para referirse a una entrada sin ambigüedades. Cada atributo de una entrada posee un tipo y uno o más valores.


Los datos del directorio se representan mediante pares de: atributo y su valor, por ejemplo el atributo commonName o cn, se usa para almacenar el nombre de una persona. cn=Jordi. Cada persona que se introduzca en el directorio se define mediante una colección de atributos que hay en la clase de objetos Personas.

Esquema de la Unidad organizativa: Personas

'Personas' tiene el objectClass: organizationalUnit

Atributos de organizationalUnit

Atributo	Carácter	Descripción	Sintaxis
Ou	Obligatorio	Nombre de la Unidad Organizativa	
userPassword	Opcional	Contraseña de usuario	OctetStringMatch
searchGuide	Opcional	Guía de búsqueda	
seeAlso	Opcional	DN relacionado	
businessCategory	Opcional	Categoría laboral	
X121Address	Opcional	Dirección X121	Numérico
registeredAddress	Opcional	Dirección Postal	
destinationIndicator	Opcional	Indicador de destino	CaseIgnoreMatch
preferredDeliveryMethod	Opcional	Método de entrega	
telexNumber	Opcional	Número de telex	
teletexTerminalIdentifier	Opcional	identificador teletex	
telephoneNumber	Opcional	Número teléfono	TelephoneMatch
internationaliSDNNumber	Opcional	Número intern. ISDN	NumericStringMatch
facsimileTelephoneNumber	Opcional	Número de fax	
Street	Opcional	Calle	
postOfficeBox	Opcional	Buzón	
postalCode	Opcional	Código Postal	
postalAddress	Opcional	DirecciónPostal	
physicalDeliveryOfficeName	Opcional	Oficina entrega	
St	Opcional	Estado o provincia	
L	Opcional	Localización	
description	Opcional	Descripción	

 <p>UOC Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

Esquema de la Unidad organizativa: Impresoras

'Impresoras' tiene dos objectClass: organizationalUnit y printerAbstract

Atributos de printerAbstract

Atributo	Carácter	Descripción	Sintaxis
printerName	Obligatorio	Nombre Impresora	
Printer-info	Opcional	Dirección IP	
Printer-make-and-model	Opcional	Marca y modelo	
printer-location	Opcional	Localización	

Esquema de la Unidad organizativa: Servidores

'Servidores' tiene dos objectClass: organizationalUnit y serverAbstract

Atributos de serverAbstract


Atributo	Carácter	Descripción	Sintaxis
gecos	Obligatorio	Nombre Servidor	
ipHostNumber	Opcional	Dirección IP	
nisMapName	Opcional	Marca y modelo	
nisMapEntry	Opcional	Localización	

Carga inicial del LDAP:

El LDAP se cargará inicialmente con la estructura definida en este capítulo. También se añade un usuario administrador.

El siguiente script realiza esta operación

```
#####
##### organizacion para uoc ####
dn: dc=tfc, dc=com
objectclass: top
objectclass: dcObject
objectclass: organization
o: TreballFiCarrera
dc: tfc
#
#
dn: ou=Personas, dc=tfc, dc=com
ou: Personas
objectClass: top
objectClass: organizationalUnit
```

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

```

dn: ou=Servidores, dc=tfc, dc=com
ou: Servidores
objectClass: top
objectClass: organizationalUnit
#
#
dn: ou=Impresoras, dc=tfc, dc=com
ou: Impresoras
objectClass: top
objectClass: organizationalUnit
#
#
dn: cn=Jordi, ou=Personas, dc=tfc, dc=com
ou: Sistemas
o: TreballFiCarrera
cn: sisujbs
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
sn: Blanca
givenName: Jordi
displayName: Jordi Blanca
title: Sr.
mail: jordibs@uoc.edu@mgs.es
initials: J.B.S.
homePostalAddress: Lepanto 344
postalAddress: Entenza 325
l: Barcelona
postalCode: 08025
telephoneNumber: 934953137
homePhone: 934369364
#####

```

El resto del LDAP se migrará del Active Directory a nuestro LDAP. La migración de los datos supondrá crear un fichero LDIF a partir de los datos del LDAP del Active Directory. Se pueden utilizar una serie de scripts para la fácil migración de datos o bien utilizar software libre o comercial ([PADL Software](#)) que realice este proceso.

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

DISEÑO DE CLASES

Paquetes del aplicativo (Packages)

Las clases del aplicativo se han agrupado en paquetes con objeto facilitar la administración de las clases.

Paquete: edu.uoc.tfc.j2ee

Contiene un fichero de constantes de la aplicación y el fichero de properties para Struts.

Paquete: edu.uoc.tfc.j2ee.action

Contiene todas las ActionsClass de la aplicación. Se ha procurado no contengan absolutamente nada de la parte de modelo del aplicativo. Su función es recuperar las variables de sesión y de los formularios correspondientes y con ellos hacer una llamada a una clase de modelo -independiente de la *Action*. Finalmente La *action* indica al controlador qué JSP debe llamar de entre las JSP's posibles para esta action.

Paquete: edu.uoc.tfc.j2ee.form

Este paquete contiene las clases Forms del aplicativo. Estos son los encargados de mapear los valores entrados en los JSP's con clases Java, para que las Actions puedan trabajar con estos valores. Pueden hacer algún tipo de validación bastante básico.

Paquete: edu.uoc.tfc.j2ee.service

Constituyen las clases de modelo de la aplicación. Si la aplicación se hubiera hecho con EJB's, estas clases son las que podrían correr en un servidor aparte y dejar a este servidor exclusivamente con las funciones de presentación y control. Las clases contenidas en este paquete son las únicas que acceden al LDAP.

Paquete: edu.uoc.tfc.j2ee.view

Este paquete contiene las clases que sirven para intercambio de información entre las capas de presentación – control – modelo. Sus métodos se limitan a los setters y getters de los atributos privados que constituyen estas clases

Descripción de las clases

Paquete	Clase	Descripción
edu.uoc.tfc.j2ee	Iconstants.java	Constantes
edu.uoc.tfc.j2ee	ApplicationsReources	Propiedades de Struts
edu.uoc.tfc.j2ee.action	AdminUserAction.java	Administración de usuarios
edu.uoc.tfc.j2ee.action	AltaAction.java	Alta en LDAP
edu.uoc.tfc.j2ee.action	BorrarAction.java	Baja en LDAP
edu.uoc.tfc.j2ee.action	CambiarLDAPAction.java	Seleccionar el LDAP
edu.uoc.tfc.j2ee.action	CambiarPassAction.java	Cambiar contraseña
edu.uoc.tfc.j2ee.action	ConsultaBuscarAction.java	Consulta del LDAP
edu.uoc.tfc.j2ee.action	EditarBajaAction.java	Mostrar datos de la Baja
edu.uoc.tfc.j2ee.action	EditarModAction.java	Mostrar datos de la Edición
edu.uoc.tfc.j2ee.action	LoginAction.java	Conexión
edu.uoc.tfc.j2ee.action	LogoutAction.java	Desconexión
edu.uoc.tfc.j2ee.action	MenuAltaAction.java	Mostrar datos del alta
edu.uoc.tfc.j2ee.action	MenuConsultaAction.java	Mostrar datos consulta
edu.uoc.tfc.j2ee.action	MenuLDAPAction.java	Mostrar datos del LDAP
edu.uoc.tfc.j2ee.action	MenuPassAction.java	Mostrar cambio contraseña
edu.uoc.tfc.j2ee.action	MenuUserAction.java	Mostrar datos de admon
edu.uoc.tfc.j2ee.action	ModificarAction.java	Modificar en LDAP

edu.uoc.tfc.j2ee.form	AdminUserForm.java	Administración de usuarios
edu.uoc.tfc.j2ee.form	AltaForm.java	Alta en LDAP
edu.uoc.tfc.j2ee.form	BorrarForm.java	Baja en LDAP
edu.uoc.tfc.j2ee.form	CambiarLDAPForm.java	Seleccionar el LDAP
edu.uoc.tfc.j2ee.form	CambiarPassForm.java	Cambiar contraseña
edu.uoc.tfc.j2ee.form	ConsultaBuscarForm.java	Consulta del LDAP
edu.uoc.tfc.j2ee.form	EditarBajaForm.java	Mostrar datos de la Baja
edu.uoc.tfc.j2ee.form	EditarModForm.java	Mostrar datos de la Edición
edu.uoc.tfc.j2ee.form	LoginForm.java	Conexión
edu.uoc.tfc.j2ee.form	LogoutForm.java	Desconexión
edu.uoc.tfc.j2ee.form	MenuAltaForm.java	Mostrar datos del alta
edu.uoc.tfc.j2ee.form	MenuConsultaForm.java	Mostrar datos consulta
edu.uoc.tfc.j2ee.form	MenuLDAPForm.java	Mostrar datos del LDAP
edu.uoc.tfc.j2ee.form	MenuPassForm.java	Mostrar cambio contraseña
edu.uoc.tfc.j2ee.form	MenuUserFormjava	Mostrar datos de admon
edu.uoc.tfc.j2ee.form	ModificarForm.java	Modificar en LDAP

edu.uoc.tfc.j2ee.service	CtxImpresora.java	Contexto LDAP de Impresora
edu.uoc.tfc.j2ee.service	CtxServidor.java	Contexto LDAP de Servidor
edu.uoc.tfc.j2ee.service	DataManagerLDAP.java	Gestor datos LDAP
edu.uoc.tfc.j2ee.service	DataModel.java	Registro de la tabla
edu.uoc.tfc.j2ee.service	IAuthentication.java	Interface Seguridad
edu.uoc.tfc.j2ee.service	IDataManager.java	Interface Datos
edu.uoc.tfc.j2ee.service	InvalidLoginException.java	Excepción de Login
edu.uoc.tfc.j2ee.service	LdifImpresora.java	Plantilla de Impresora
edu.uoc.tfc.j2ee.service	LdifServidor.java	Plantilla de Servidor

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

edu.uoc.tfc.j2ee.service -----	SecurityLDAP.java	Gestor de Seguridad
edu.uoc.tfc.j2ee.view	DataViewForm.java	Boundle datos LDAP
edu.uoc.tfc.j2ee.view	UserViewForm.java	Boundle datos usuario

Descripción de páginas

Paquete	Clase
alta.jsp	Alta registros LDAP
consulta.jsp	Presentación de Consulta
consultaBuscar.jsp	Presentación Datos LDAP
editar.jsp	Modificación registros LDAP
editarBaja.jsp	Mostrar datos Baja
editarMod.jsp	Mostrar datos Edición
login.jsp	Conexión
menu.jsp	Menú Principal
MenuLDAP.jsp	Selección de LDAP
MenuPass.jsp	Cambio de Contraseña
MenuUser.jsp	Administración de Usuarios

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

El controlador

Desde Struts, el control de la aplicación se conduce desde un fichero que regula el tráfico de Actions, Formularios y páginas JSP.

Por lo general, las llamadas dentro de la aplicación se hacen a un tipo de componentes de Struts llamado Actions. Antes que la clase 'Action' recoja la llamada, se llama al Formulario relacionado y este hace un mapeo de los campos del JSP con los atributos de la propia clase, e incluso puede llegar a hacer una primera verificación de los datos introducidos.

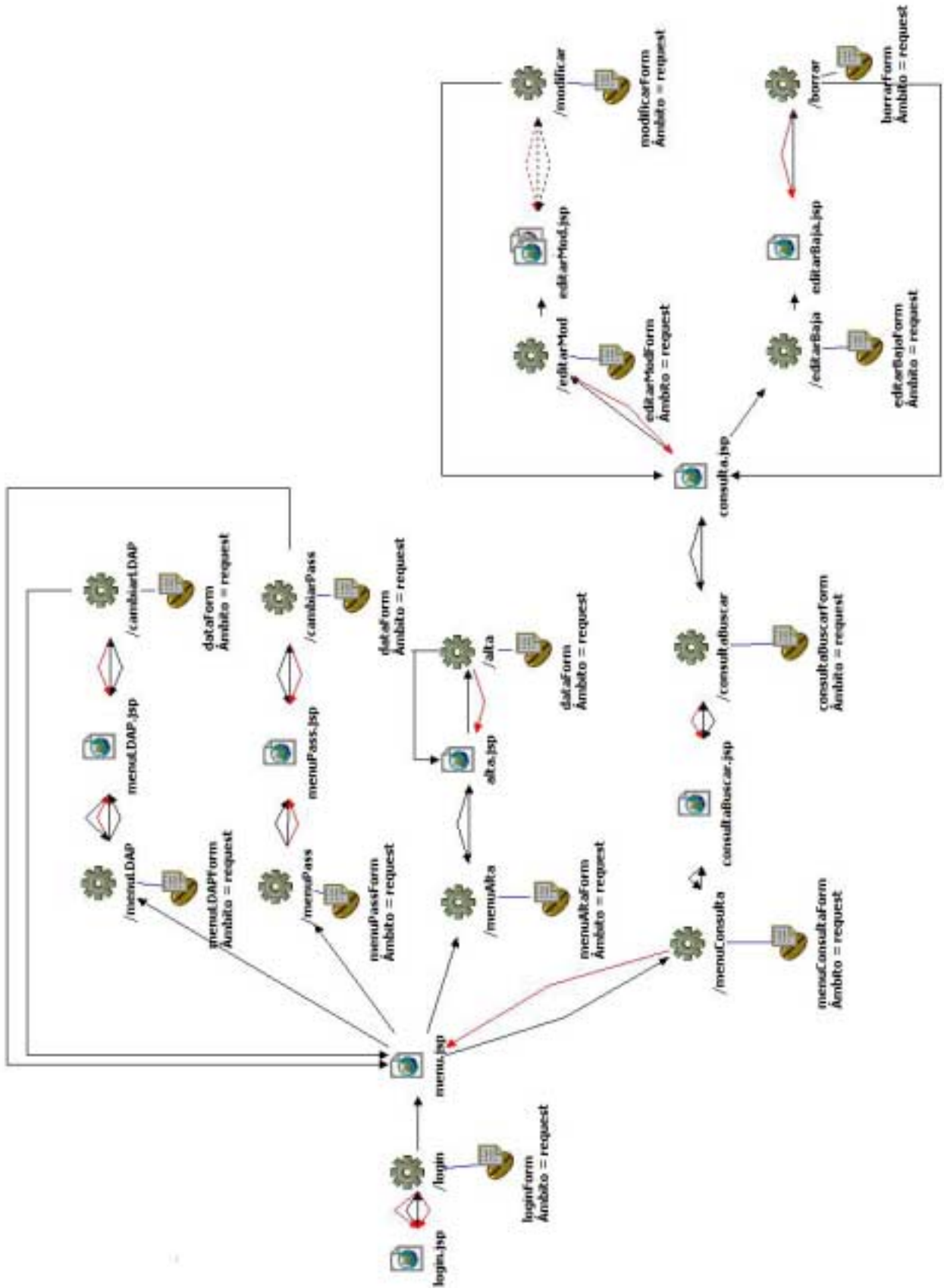
A continuación la clase 'Action' llamada toma el control. La clase Action puede acceder a los datos de sesión, como cualquier componente web J2EE, y a los datos de formulario asociado.

La 'Action' hace la llamada a los procesos de Modelo traspasando las clases definidas de modelo. En nuestro caso estas clases van a ser IDataManager y Iauthentication. La primera va a ser la clase que contendrá los datos que se traspasan aplicación y 'Modelo' para la gestión del LDAP (Consultas, Altas, Modificaciones y Bajas de registros LDAP), mientras que la segunda contiene los datos que traspasa el 'Modelo' a la aplicación referidos al Usuario.

En función del resultado devuelto por el modelo, la 'action' decide hacer realizar un forward a cualquiera de los JSP's.

En la fase de 'Control' del modelo MVC consiste en ligar todos estos componentes: Actions, Forms, JSP's.

A continuación se muestra una representación gráfica de lo que necesita el fichero de Control para gobernar la aplicación. En esta representación no se han representado los conectores entre cualquier todos los JSP's y las acciones /login, /menuPass, /menuLDAP, /menuAlta y /menuConsulta. No se han representado por clarificar el diagrama, sin embargo desde cualquier página siempre podemos acceder por medio de los botones 'Logoff' 'Alta' 'Consulta' 'LDAP' y 'Contraseña' a las acciones comentadas.



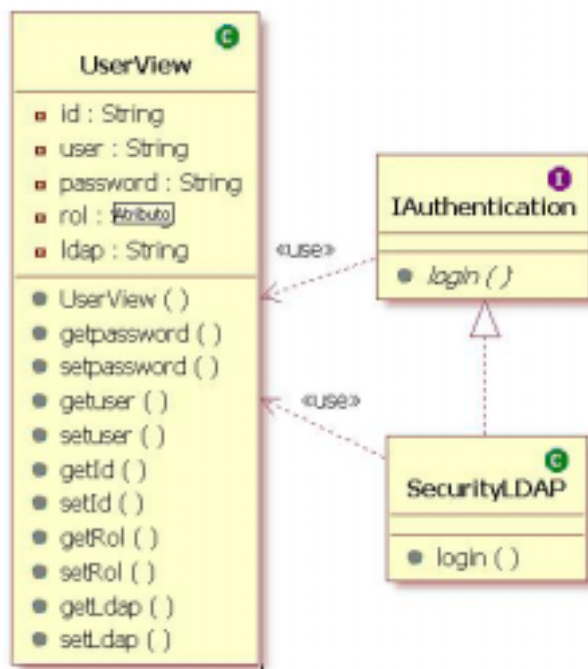
El Modelo

Las clases del 'Modelo' son las que tienen contacto con los datos del negocio.

Con estas clases accederemos a las bases de datos, en nuestro caso accederemos a un directorio LDAP.

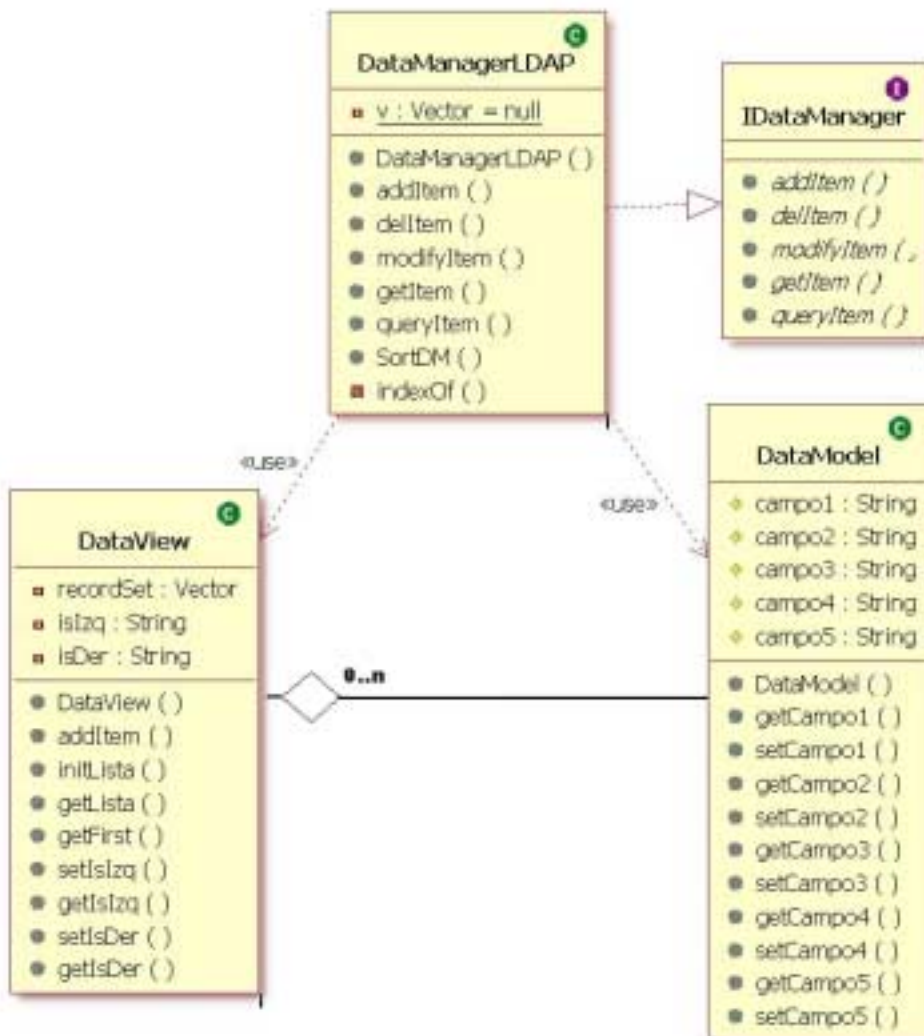
Para la Autenticación se define la clase SecurityLDAP

Esta clase con su método login de firma: UserView login (user, password)
Devuelve un objeto de la clase UserView que se guarda como atributo de una sesión.
UserView tiene los parámetros específicos y necesarios durante la aplicación



Para el acceso a los datos de Servidores e Impresoras dentro del LDAP se utiliza la clase DataManagerLDAP. Los parámetros de los métodos de esta clase manejan objetos de la clase DataModel y DataView.

DataModel tiene una correspondencia con un registro del LDAP y DataView es un Recorset de DataModel's y contiene el resultado de una consulta.



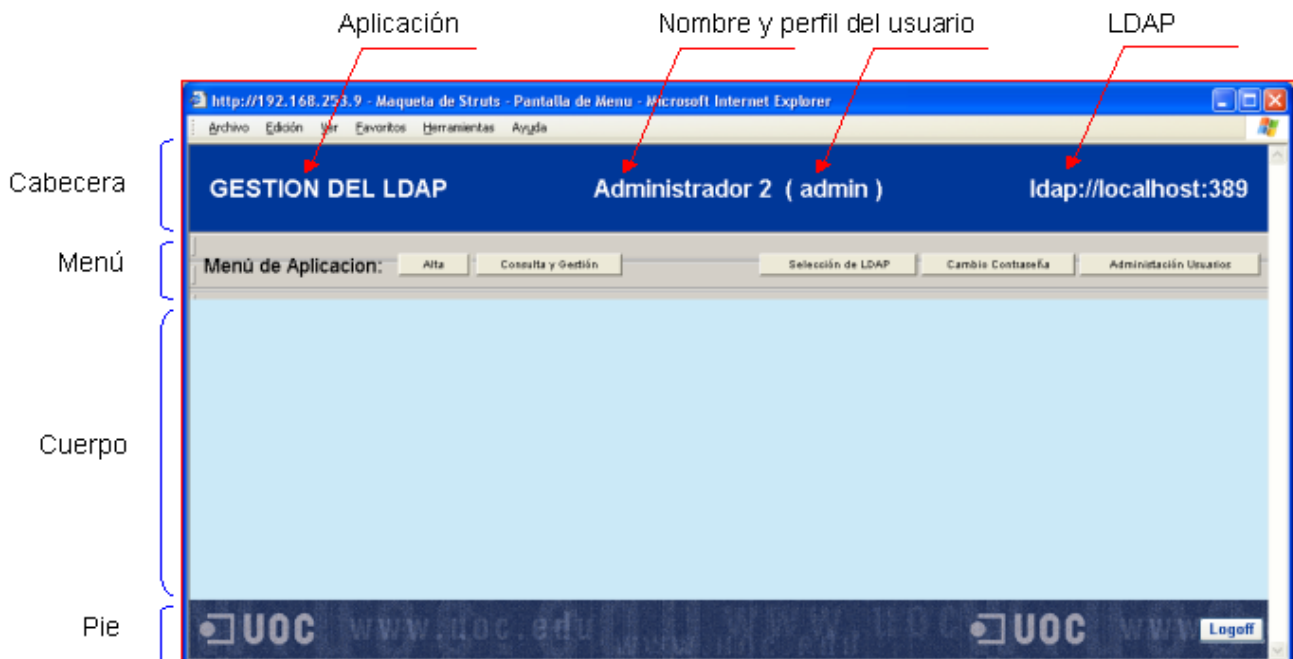
 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	


CLASES DE LA VISTA

La tercera parte del framework del modelo MVC es la vista, es decir, la parte que visualiza el usuario final.

El diseño de todas las páginas es común, y sigue un patrón en todas las pantallas de la aplicación. Se divide en cuatro apartados:

- Cabecera: Destaca el título de la aplicación y contiene información relativa al nombre del usuario y el directorio LDAP al que se está accediendo. Es constante durante la aplicación mientras no se cambio de usuario o se decide cambiar de LDAP.
- Pie de página: Esta parte es fija durante toda la aplicación. Contiene un logo corporativo y un botón 'Logoff' que lleva a la página de identificación para cambiar de usuario. Si se pulsa, se destruyen todos los objetos de sesión que pueda tener la aplicación.
- Una barra de menús que está personalizada en función del tipo de usuario: el administrador que puede alterar el contenido del LDAP; y el usuario genérico que solo puede realizar consultas. Todos los usuarios autenticados pueden cambiar su contraseña y seleccionar uno de entre los directorios LDAP a los que tienen permiso de acceso.
- Área de trabajo o Cuerpo. Zona de trabajo situada en la parte central donde se desarrolla la aplicación.



 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

Relación de páginas:

Página login.jsp

El traspaso de información entre las páginas que ve el usuario final y la aplicación se establece a través de los campos que se visualizan o rellenan en la pantalla y los parámetros de sesión que se han cumplimentado en la fase de modelo.

En la presentación de esta página no se ofrece ningún dato, puesto que el usuario todavía tiene que validarse. Esta página sólo tiene que presentar los campos para rellenar y el campo de error por si se ha producido alguna incidencia en la autenticación del usuario. Struts ya ofrece los mecanismos para repintar la pantalla con el resultado de los errores o excepciones que se hayan producido.

Si no se producen errores esta página siempre lleva a la action 'TFC/action/login'.


Página /TFC/action/login

Accesible por todos los usuarios.

Los datos de usuario y contraseña enviados en la pantalla anterior se mapean con el formulario loginForm. El diseño de la clase es el siguiente:

LoginForm C	
▣	password : String
▣	usuario : String
●	LoginForm ()
●	validate ()
●	reset ()
●	setUsuario ()
●	getUsuario ()
◆	resetFields ()
●	getPassword ()
●	setPassword ()

La action loginAction comprueba la validez de los datos y si es correcto direcciona la salida a menu.jsp. En este punto, el usuario sólo puede pulsar una de las opciones de la barra de menú, dado que el área de trabajo está en blanco.

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

La pantalla tiene una cabecera y un pie de página con información genérica, y además una barra de menú de la aplicación. La barra está personalizada dependiendo del tipo de usuario. Si el usuario es administrador le aparece tres botones adicionales que corresponden a: 'Alta de registros en el LDAP', 'Selección de LDAP' y 'Gestión de usuarios'. El área de trabajo queda totalmente libre porque el usuario todavía no ha realizado ninguna acción de trabajo.

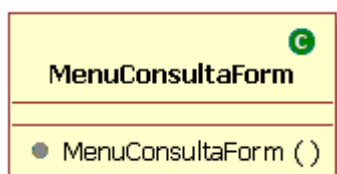
Estas opciones son comunes para todas las páginas de la aplicación. Sólo las voy a desarrollar en ésta, aunque es válida para todas las demás excepto la login.jsp.

Las opciones del jsp son las siguientes: :

- pulsar botón *alta* que se llama a 'TFC/action/alta'
- pulsar botón *consulta* con lo que se llama a TFC/action/menuConsulta
- pulsar botón *LDAP* con lo que se llama a 'TFC/action/menuLDAP'
- pulsar botón *Contraseña* que se llama 'TFC/action/menuPass'
- pulsar botón *Gestión Usuarios* con lo que se llama a 'TFC/action/menuUser'
- pulsar botón *Logout* con lo que se llama a 'TFC/login.jsp'

Página /TFC/action/menuConsulta

El usuario viene de una pantalla que no envía datos, por lo tanto el formulario de mapeo *menuConsultaForm* existe en su mínima expresión:



La action no tiene modelo y redirecciona la salida al JSP *consultaBuscar.jsp* que tiene un campo que solicita un argumento de selección de registros. Al pulsar Aceptar se llama a 'TFC/action/consultaBuscar'

Página /TFC/action/consultaBuscar

Accesible por todos los usuarios, aunque está personalizada.

El perfil administrador tiene en cada fila una que muestra los registros seleccionados una columna para marcar los registros que se desea borrar. Además los identificadores tienen un enlace que lleva a la pantalla de edición para su posterior modificación. El perfil usuario no contempla ni la columna de borrado, ni los enlaces para edición.

Esta pantalla necesita un área de datos para poder representar la tabla con los registros seleccionados. Además necesita un control para paginación y uno más

 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

columna seleccionada para clasificación. También maneja el mismo campo que la pantalla anterior solicitando un argumento de selección de registros. Para manejar las acciones seleccionadas por el usuario la acción correspondiente, tenemos el formulario *ConsultaBuscarForm* que tiene la siguiente información:

ConsultaBuscarForm
▪ campo1 : String
▪ botonIzq : String
▪ botonDer : String
▪ pantalla : String
▪ sort1 : String
▪ sort2 : String
▪ sort3 : String
▪ sort4 : String
▪ sort5 : String
▪ lista : List = null

La acción recupera los datos de la parte de 'modelo' y redirecciona la salida al JSP: *consulta.jsp*.

Esta página coloca en el 'Área de trabajo' el resultado de la selección en el LDAP y se muestra en forma de tabla con cinco filas (parametrizable en la clase de *IConstants*). Esta tabla es paginable, de forma que, si procede aparecen unos botones '<<<' y '>>>' para retroceder o avanzar en la paginación respectivamente.

Además los títulos de las columnas mantienen unos hipervínculos que permiten ver la información clasificada por cualquiera de las columnas.


Las posibles salidas de este jsp son los botones de paginación y los de clasificación de la tabla. Estos vuelven a autollamar a la misma pantalla: */TFC/action/consultaBuscar*.

Si se pulsa algunos de los links que modifican o borran un registro, la aplicación va a la pantalla */TFC/editarMod* o *TFC/editarBaja* respectivamente.

Página */TFC/action/editarMod*

Sólo accesible por usuarios con rol de administrador en la aplicación.

Esta página recibe los datos del registro seleccionado en la pantalla anterior.

 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

La clase asociada a los valores del formulario es editarModForm y tiene la siguiente estructura:

EditarModForm C
<ul style="list-style-type: none"> ▣ campo1 : String ▣ campo2 : String ▣ campo3 : String ▣ campo4 : String ▣ campo5 : String
<ul style="list-style-type: none"> ● EditarModForm () ● validate () ● reset () ● setCampo1 () ● setCampo2 () ● setCampo3 () ● setCampo4 () ● setCampo5 () ● getCampo1 () ● getCampo2 () ● getCampo3 () ● getCampo4 () ● getCampo5 () ◆ resetFields ()


Cuando entra la clase 'action', recupera los datos del registro y redirige la salida a la JSP: *editarMod.jsp*. Esta página presenta los datos del registro para proceder a su modificación. Si se presiona el botón 'Modificar', se llama a la página *TFC/action/modificar*, en caso de pulsar la tecla 'Cancelar' se redirecciona a la página */TFC/action/consultaBuscar*.

Página /TFC/action/modificar

Sólo accesible por usuarios con rol de administrador en la aplicación.

Esta página recibe los datos del registro seleccionado en la pantalla anterior.

La acción pasa el registro a al modelo para su actualización. La salida se redirecciona a la página */TFC/action/consultaBuscar*.

 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

Página /TFC/action/editarBaja

Sólo accesible por usuarios con rol de administrador en la aplicación.

Esta página recibe los datos del registro seleccionado en la pantalla anterior.

La clase asociada a los valores del formulario es *editarBajaForm* y tiene la siguiente estructura:


EditarBajaForm C
<ul style="list-style-type: none"> ▣ campo1 : String ▣ campo2 : String ▣ campo3 : String ▣ campo4 : String ▣ campo5 : String
<ul style="list-style-type: none"> ● EditarBajaForm () ● validate () ● reset () ● setCampo1 () ● setCampo2 () ● setCampo3 () ● setCampo4 () ● setCampo5 () ● getCampo1 () ● getCampo2 () ● getCampo3 () ● getCampo4 () ● getCampo5 () ◆ resetFields ()

Cuando entra la clase 'action', recupera los datos del registro y redirige la salida a la JSP: *editarBaja.jsp*. Esta página presenta los datos del registro para proceder a su modificación. Si se presiona el botón 'Borrar' se llama a la página *TFC/action/borrar*, en caso de pulsar la tecla cancelar se redirecciona a la página */TFC/action/consultaBuscar*.

Página /TFC/action/borrar

Sólo accesible por usuarios con rol de administrador en la aplicación.

Esta página recibe los datos del registro seleccionado en la pantalla anterior.

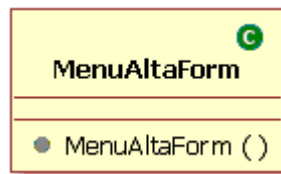
	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

La acción pasa el registro a al modelo para su borrado. La salida se redirecciona a la página */TFC/action/consultaBuscar*.

Página */TFC/action/menuAlta*

Sólo accesible por usuarios con rol de administrador en la aplicación.

El usuario viene de una pantalla que no envía datos, por lo tanto el formulario de mapeo *menuAltaForm* existe en su mínima expresión:



La action no tiene modelo y redirecciona la salida al JSP *alta.jsp* que solicita los datos para añadir el nuevo registro. Al pulsar 'Añadir' se llama a la *'TFC/action/alta'*

Página */TFC/action/alta*

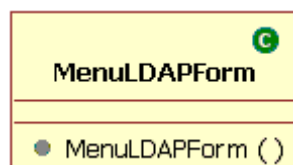
Sólo accesible por usuarios con rol de administrador en la aplicación.


Esta página recibe los datos del registro integrado en la pantalla anterior.

La acción pasa el registro a al modelo para su inserción. La salida se redirecciona a la página */TFC/menuAlta* para permitir más altas de registros.

Página */TFC/action/menuLDAP*

El usuario viene de una pantalla que no envía datos, por lo tanto el formulario de mapeo *menuLDAPForm* existe en su mínima expresión:



	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

La acción no tiene modelo y redirecciona la salida al JSP *menuLDAP.jsp* que solicita el valor del nuevo LDAP. Al pulsar 'Aceptar' se llama a la '*TFC/action/cambiarLDAP*'

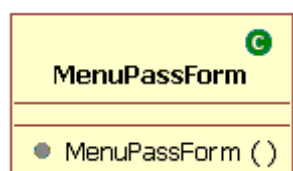
Página */TFC/action/cambiarLDAP*

Esta página recibe el dato del LDAP seleccionado.

La acción modifica el nuevo valor con el atributo de sesión correspondiente. La salida se redirecciona a la página */TFC/login* que presenta el 'Área de trabajo' en blanco.

Página */TFC/action/menuPass*

El usuario viene de una pantalla que no envía datos, por lo tanto el formulario de mapeo *menuPassForm* existe en su mínima expresión:



La acción no tiene modelo y redirecciona la salida al JSP *menuPass.jsp* que solicita el valor de la nueva contraseña. Al pulsar 'Aceptar' se llama a la '*TFC/action/cambiarPass*'

Página */TFC/action/cambiarPass*

Esta página recibe el dato de la nueva contraseña.

La acción pasa el nuevo valor al modelo para su actualización. La salida se redirecciona a la página */TFC/menuAlta* para permitir más altas de registros.

La salida se redirecciona a la página */TFC/login* que presenta el 'Área de trabajo' en blanco.

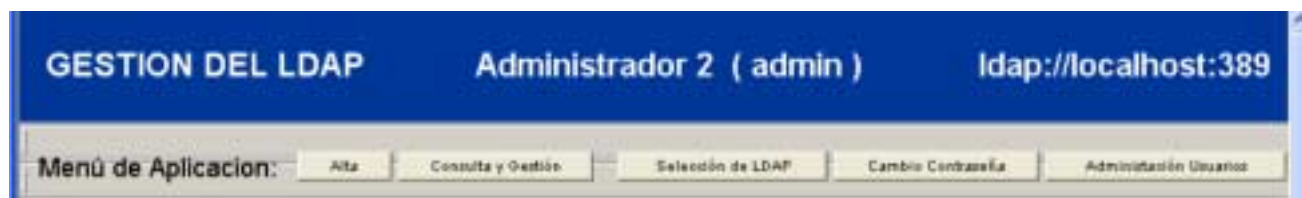
	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

FUNCIONALIDADES EN LA APLICACION

Durante el desarrollo de la aplicación se han realizado algunas ampliaciones al prototipo propuesto en el análisis:

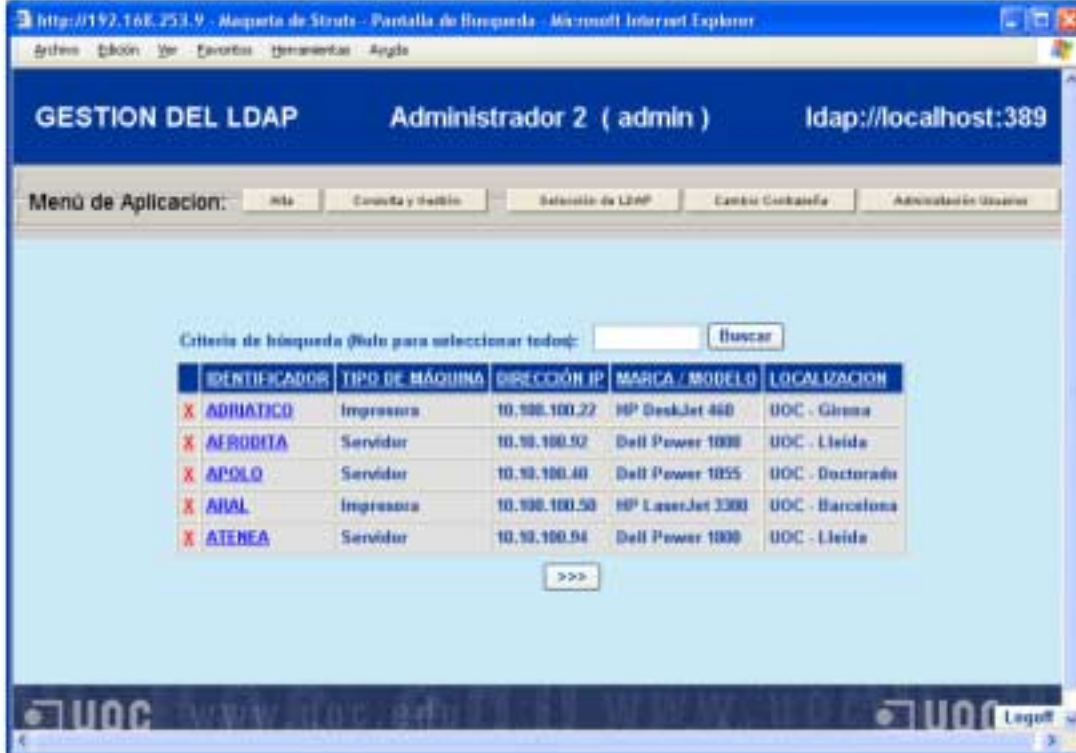
1. Se distinguen dos perfiles de usuario en la aplicación, el perfil de administrador y el de consultor LDAP. Mientras el primero puede realizar las operaciones de alta, baja, modificación y consulta; el perfil de consultor sólo tiene acceso a la consulta.
2. Se permite a cualquier usuario que pueda entre en la aplicación, modificar su propia contraseña.
3. Las acciones de alta, baja y modificación tienen siempre un botón para cancelar la operación, y el botón de aceptación coincide con el verbo de la acción que se realiza: agregar, borrar, modificar.
4. El número de filas del listado de servidores e impresoras está limitado por un parámetro y se permite la paginación. Además, pinchando sobre la cabecera de las columnas permite ordenar el listado por la columna correspondiente.
5. Cuando se entra en la aplicación se asigna un LDAP por defecto al usuario. Ahora existe una opción en el menú de aplicación que permite en cualquier momento cambiar de LDAP, seleccionándolo de una lista de directorios posibles.
6. Un usuario con rol de administrador puede gestionar los usuarios del LDAP y asignar permisos como administrador o usuario o bien quitar permisos para poder acceder a la aplicación

OPCIONES EN LA BARRA: MENÚ DE APLICACIÓN

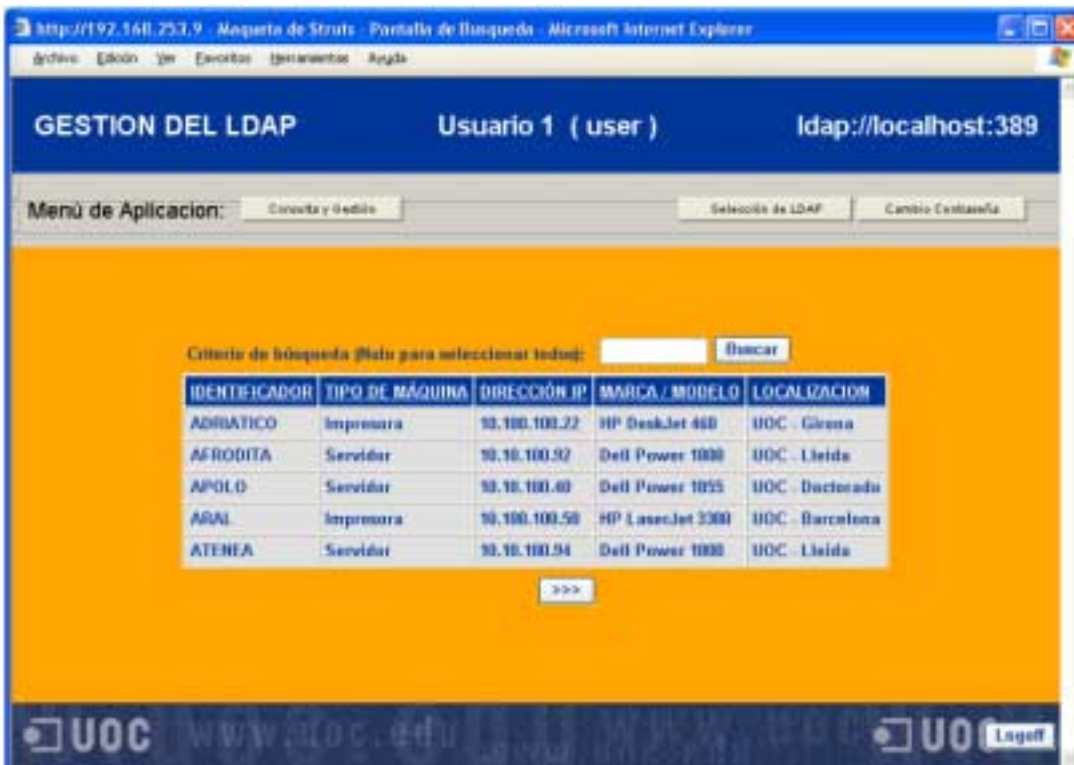


La aplicación soporta dos tipos de roles: uno administrador que puede gestionar totalmente el LDAP aparte de tener permisos para otras utilidades del aplicativo y otro usuario que realiza operaciones de consulta. Visualmente se han dado dos tonalidades distintas dependiendo del rol: la azulada para los administradores y otra anaranjada para los usuarios con rol que sólo permite hacer consultas.

DOS PERFILES DE USUARIO: 1: ADMINISTRADOR



DOS PERFILES DE USUARIO: 2 USUARIO






Las características de usabilidad que ofrecen las pantallas de la aplicación son las siguientes:

Mantiene una cabecera, la barra de menú general y el pie constante durante toda la aplicación. Los botones situados en estas áreas siempre están activos, ello permite al usuario concentrar la atención o bien en el cuerpo de la pantalla, o bien en la periferia. En este último caso todos los elementos permanecen inalterables sea cual sea la pantalla en que nos encontramos. Por otra parte, esta parte fija del aplicativo pretende llevar el máximo de información al usuario.

La pantalla de consulta, que es la que se representa en la figura anterior, tiene varios controles que permiten aumentar la productividad de sus destinatarios. Se ofrece un campo de filtro que permite acotar la consulta de la base de datos. En esta versión, el criterio es muy sencillo: si se deja en blanco el campo de filtro se listan todos los registros del LDAP. Si se pone una cadena de caracteres en este campo, aparecerán aquellas filas cuyo identificador coincida con las letras desde el principio. Es decir, si en el campo de filtro ponemos 'A', como resultado aparecerán aquellas filas que el identificador empiece por 'A' (no se diferencian mayúsculas i minúsculas). En próximas versiones se puede ampliar esta capacidad para permitir distintos tipos de comodines para montar la plantilla de búsqueda.

Las filas seleccionadas se pueden clasificar alfabéticamente por cualquiera de sus columnas. Las cabeceras de las columnas comienzan por la descripción de las mismas y tienen un hipervínculo, que si es pulsado, vuelve a mostrarse la misma página pero con las filas clasificadas según el orden de la columna. Por ejemplo, si pulsamos en el

 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

texto Localización, se vuelven a mostrar las filas ordenadas por orden de Localización (del servidor o la impresora).

Los usuarios que tienen el rol de administrador, pueden ver como todas las celdas de la columna 'identificador' y la columna situada en primera posición con una 'X' tienen un hipervínculo. Este hipervínculo lleva a la edición o al borrado de la fila.

Las filas de esta pantalla de consulta son paginables. Eso quiere decir que, si se determina por un parámetro de la aplicación que las tablas visualicen 5 filas, en caso de que la consulta tenga como resultado más de cinco, sólo mostrará las cinco primeras y aparecen unos botones de paginación que permiten avanzar en la tabla a los cinco siguientes. En esta pantalla ya aparecen dos botones (si procede) uno para avanzar a los cinco siguientes y otro botón para retroceder a los cinco anteriores.

ORDENACION POR COLUMNA 'IDENTIFICADOR' O CUALQUIER OTRA COLUMNA


	IDENTIFICADOR	TIPO DE MÁQUINA	DIRECCIÓN IP	MARCA / MODELO	LOCALIZACION
X	HERACLES	Servidor	10.10.12.94	Dell Power 1800	UOC - Lleida
X	HERMES	Servidor	10.10.100.43	Dell Power 1855	UOC - Doctorado
X	JONICO	Impresora	10.100.100.66	HP DeskJet 460	UOC - Girona
X	KRONOS	Servidor	10.10.45.67	Dell Power 1855	UOC - Doctorado
X	MEDITERRANEO	Impresora	10.100.100.150	HP LaserJet 3380	UOC - Barcelona

PAGINACION HACIA DELANTE Y ATRÁS

Criterio de búsqueda (Nulo para seleccionar todos):

	IDENTIFICADOR	TIPO DE MÁQUINA	DIRECCIÓN IP	MARCA / MODELO	LOCALIZACION
X	HERACLES	Servidor	10.10.12.94	Dell Power 1800	UOC - Lleida
X	HERMES	Servidor	10.10.100.43	Dell Power 1855	UOC - Doctorado
X	JONICO	Impresora	10.100.100.66	HP DeskJet 460	UOC - Girona
X	KRONOS	Servidor	10.10.45.67	Dell Power 1855	UOC - Doctorado
X	MEDITERRANEO	Impresora	10.100.100.150	HP LaserJet 3380	UOC - Barcelona

<<< >>>

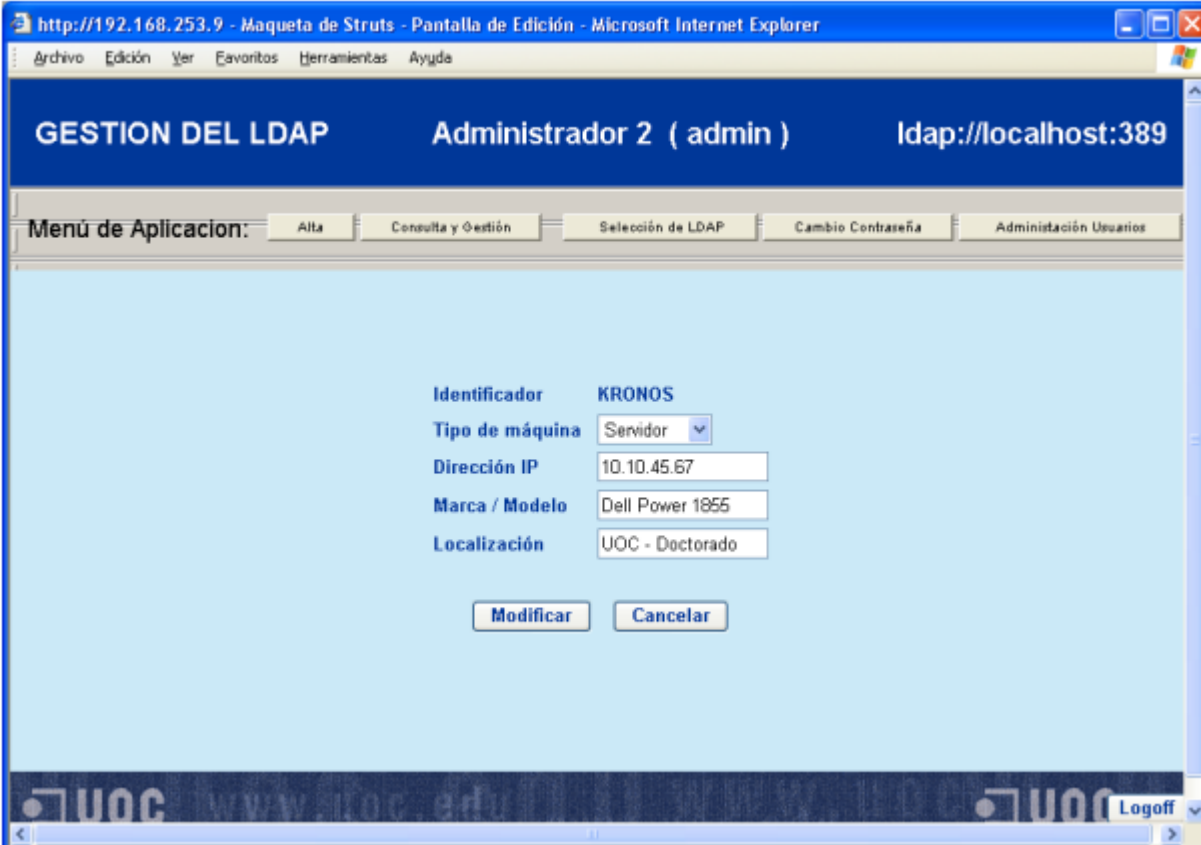
 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

En la figura siguiente se puede observar cómo al usuario administrador le aparecen los campos de identificador con un enlace así como la primera columna que contiene 'X'. Esto habilita al rol administrador a poder ejecutar modificaciones y bajas. Los usuarios que no tienen el rol administrador no tienen esa posibilidad. Asimismo, la barra de menú aparece para los administradores con la opción 'alta', cosa que no puede ver el resto de usuarios.

IDENTIFICADOR	TIPO DE MÁQUINA
ADRIATICO	Impresora
AFRODITA	Servidor
APOLO	Servidor
ARAL	Impresora
ATENEA	Servidor

	IDENTIFICADOR	TIPO DE MÁQUINA
X	HERACLES	Servidor
X	HERMES	Servidor
X	JONICO	Impresora
X	KRONOS	Servidor
X	MEDITERRANEO	Impresora

Cuando se solicita al usuario el campo 'Tipo de máquina', aparece un desplegable para seleccionar una de las dos entradas posibles: Impresora o Servidor.




The screenshot shows a web browser window with the URL `http://192.168.253.9 - Maqueta de Struts - Pantalla de Edición - Microsoft Internet Explorer`. The page title is "GESTION DEL LDAP" and the user is logged in as "Administrador 2 (admin)" with the session ID "ldap://localhost:389".

The application menu includes: Alta, Consulta y Gestión, Selección de LDAP, Cambio Contraseña, and Administración Usuarios.

The form displays the following fields:

- Identificador: KRONOS
- Tipo de máquina: Servidor (dropdown menu)
- Dirección IP: 10.10.45.67
- Marca / Modelo: Dell Power 1855
- Localización: UOC - Doctorado

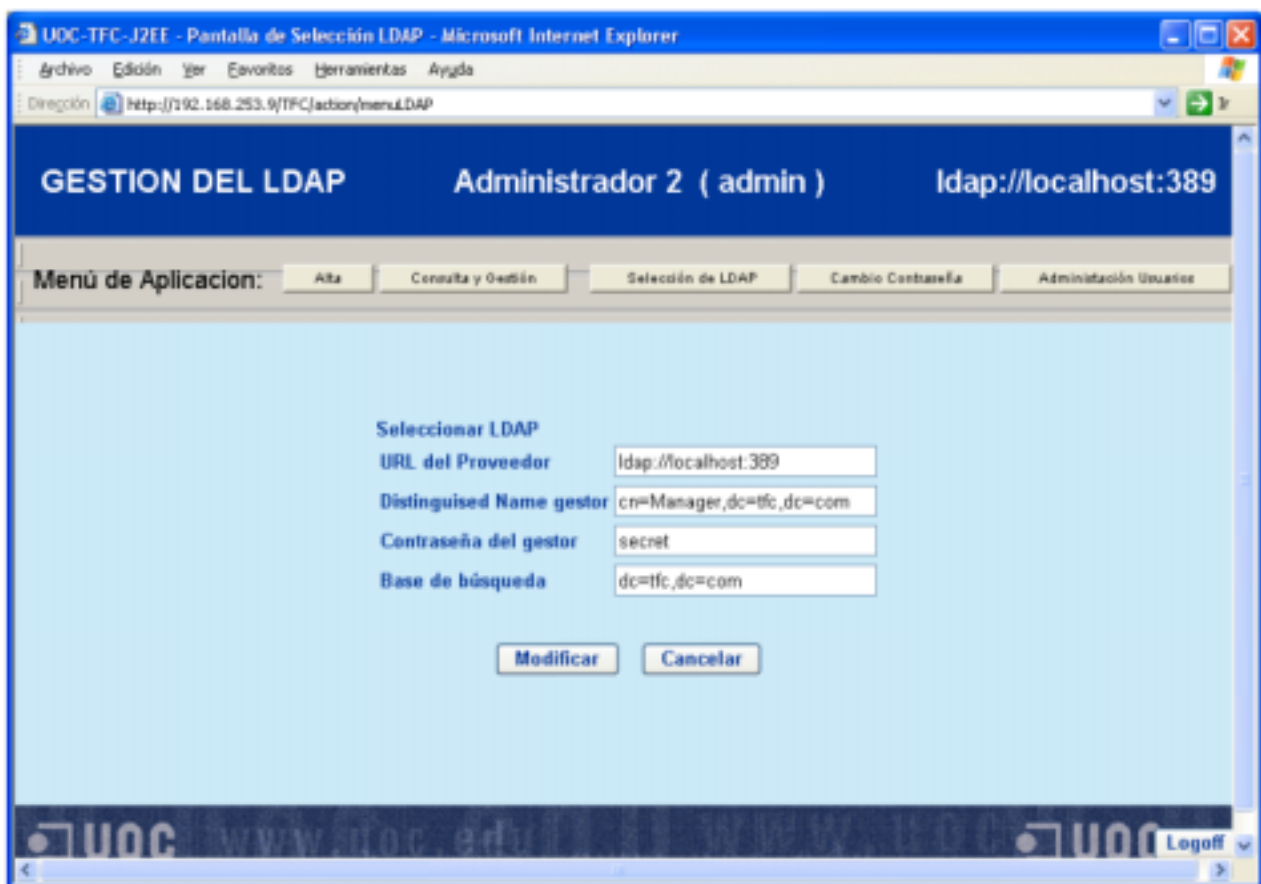
Buttons for "Modificar" and "Cancelar" are visible at the bottom of the form.

 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

LDAP SELECCIONABLE

La aplicación permite que un usuario pueda cambiar el LDAP. Para ello deben conocerse el nombre de la URL del servidor y el puerto por el que escucha. Además debe entrarse el Distinguished Name con autoridad para hacer consultas y modificaciones en el LDAP, así como su contraseña.

El esquema de este LDAP debe coincidir con el que se ha mostrado anteriormente. El cambio de LDAP afecta a los listados de Servidores e Impresoras. La información referida a personas (nombre, permisos, contraseñas, etc) siempre tira del mismo servidor LDAP inicial.



UDC-TFC-J2EE - Pantalla de Selección LDAP - Microsoft Internet Explorer

http://192.168.253.9/TFC/acton/menuLDAP

GESTION DEL LDAP **Administrador 2 (admin)** **ldap://localhost:389**

Menú de Aplicación: Alta Consulta y Gestión **Selección de LDAP** Cambio Contraseña Administración Usuarios

Seleccionar LDAP


URL del Proveedor:

Distinguished Name gestor:

Contraseña del gestor:

Base de búsqueda:


UOC www.uoc.edu Logoff

 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

CAMBIO DE CONTRASEÑA

Desde el aplicativo se permite que el usuario pueda cambiar su contraseña:



 <p>Universitat Oberta de Catalunya www.uoc.edu</p>	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

GESTION DE USUARIOS.

Un administrador de esta aplicación, puede dar o quitar permisos de entrada a la misma al resto de usuarios del LDAP. Lo que se ha determinado que no puede hacer desde esta aplicación es dar altas o bajas de usuarios.



	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

CONCLUSIONES

Personalmente, creo que se han cumplido los objetivos propuestos inicialmente en el ya lejano 'Plan de Trabajo'

Los hitos se han cumplido escrupulosamente y ello ha permitido llegar al día de hoy con la aplicación totalmente operativa y los documentos que la acompañan finalizados. Creo que a lo largo de la carrera no se ha dado la suficiente importancia a una correcta planificación. Cuando los plazos de entrega son largos en el tiempo, y por largos me refiero a meses como es el caso de este trabajo de fin de carrera en el que disponemos de 4 meses para su realización, es imprescindible marcar unos puntos intermedios que permitan hacer una autocrítica de la evolución del plan de trabajo. Esto se eleva exponencialmente cuando en la elaboración de un proyecto intervienen grupos de personas. En este último caso se ve claramente la funcionalidad del 'Jefe de proyecto', el cual tiene como una de sus funciones el desarrollo y seguimiento de un plan de trabajo.

Por otra parte, este trabajo me ha supuesto una ventana para ver como se puede organizar un equipo informática. El paradigma MVC, aparte de la fuerza teórica y aportación tecnológica que ofrece al desarrollo de aplicaciones, aporta una división del trabajo que permite que diversos equipos puedan trabajar en facetas y técnicas totalmente distintas. De manera que, pueden existir equipos de desarrolladores que conozcan perfectamente la línea de negocio de la empresa sin la necesidad de tener conocimientos de tecnologías web. Otro grupo por el contrario, sería el encargado de manejar la navegación de pantallas y el control de la aplicación, éstos no requieren tener un conocimiento detallado del modelo de negocio, pero sí deben conocer perfectamente los pormenores del análisis de la aplicación. Finalmente, la parte de presentación podría estar llegando a estar 'externalizada' y otro equipo se podría encargar del desarrollo de la parte de presentación o visualización. La presentación podría estar en distintos formatos: formato web, formatos para PDA's, formatos para webservices, etc. Este último equipo podría estar compuesto por expertos en JavaScript que sin embargo y HTML dinámico necesitan tener extensos conocimientos de Java.


Resumiendo, con este trabajo he descubierto una nueva dimensión en el desarrollo de aplicaciones informáticas que es la organización dentro de un equipo informática y las técnicas y herramientas disponibles.

Me ha sorprendido, la facilidad con que algo tan distante como puede ser un Servidor de directorios LDAP, y cuando digo distante me refiero a nunca visto en el resto de asignaturas, se ha podido integrar en mi aplicación y la facilidad con que se podría

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

reconvertir el aplicativo para que la fuente de datos fuera un servidor convencional de base de datos como MySQL. Apenas supondría añadir un par de clases del modelo de negocio que heredarían de las Interfaces IDataManager e Iauthentication. ¡Viva la reutilización!

Debo reconocer que la elaboración de este trabajo ha sido un auténtico paso por el desierto, dedicando cientos de horas a no se sabe qué, pero finalmente ha salido a la luz un trabajo por el que siento un especial orgullo. Llamadle vanidad.

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

LINEAS DE FUTURO DESARROLLO

Dada la limitación que tiene en el tiempo el trabajo de fin de carrera y aunque los objetivos personales que me había propuesto al inicio de curso creo se han cumplido, e incluso se han enriquecido con las aportaciones propuestas Jordi Ceballos, creo que hay algunos aspectos que me gustaría ampliar del presente proyecto.

El primero sería que la aplicación estuviera protegida por el propio Tomcat por medio de la seguridad J2EE con los parámetros suministrados en el fichero web.xml tanto en lo que se refiere a la autenticación de usuarios como a la autorización a los recursos a los que se debe acceder. Con ello la aplicación quedaría descargada de esta funcionalidad y sólo le quedaría la personalización de las páginas según el rol que desempeña el usuario.

También debe reforzarse la seguridad en cuanto a la política de contraseñas para evitar que los usuarios utilicen claves triviales o repetidas en los últimos ciclos.

También ha quedado pendiente mejorar la presentación del aplicativo. Sin embargo, en este sentido soy de la opinión que los auténticos profesionales de esta técnica son los publicistas y no los informáticos. Y ellos son los que deben dar el 'toque' corporativo y presencial a los aplicativos. El trabajo del informático es utilizar estos recursos recogidos a modo de plantillas y ficheros CSS y aplicarlos en los desarrollos informáticos.

Finalmente, y aunque expresamente queda fuera de este proyecto por ser requerimiento del trabajo de fin de carrera utilizar software Open Source, queda adaptar el aplicativo para que trabaje con el esquema de 'Active Directory' que es el entorno donde queda plenamente justificado el presente proyecto. En cualquier caso, las modificaciones necesarias para esta adaptación serían mínimas.

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

BIBLIOGRAFIA

APACHE-TOMCAT

Apache Tomcat 5
Vivek Chopra
Wiley Publishing, Inc

Apache Jakarta-Tomcat
James Goodwill
Apress

Servlets and JavaServer Pages
Jayson Falkner
Addison-Wesley

Tomcat: The Definitive Guide
Jason Britain
O'Reilly

STRUTS

The Struts Framework, Practical guide for Java Programmers
Sue Spielman
Morgan Kaufman Publishers

Struts in Action, Building web applications with the leading Java framework
Ted Husted
Manning

Jakarta Struts
Chuck Cavaness
O'Reilly

Mastering Jakarta Struts
James Goddwill
Wiley Publishing, Inc

Jakarta Struts for Dummies
Mike Robinson and Ellen Finkelstein
Wiley Publishing, Inc

	TREBALL FI DE CARRERA - J2EE	
	Semestre set. 05 – gen. 06	MEMORIA
	F. Jorge BLANCA SÁNCHEZ	

OpenLDAP

El website OpenLDAP Project - <http://www.openldap.org>

The ABC's of LDAP: How to Install, Run and Administer LDAP Services
Reinhard Voglmaier
Auerbach Publications

LDAP System Administration
Gerald Cartes
O'Reilly

Understanding and Deploying LDAP Directory Services, Second Edition
Timothy A. Howes
Addison Wesley

Active Directory

Planning Implementing and Maintaining a Microsoft Server 2003 Active Directory
Infrastructure
Bill Spealman
Microsoft Press 2004

JNDI

Java Naming and Directory Interface. Application Programming Interface (JNDI API)
Sun Microsystems, Inc

Java Data Access – JDBC, JNDI and JAXP

MyEclipse

Tips and tricks for eclipse and the IDE MyEclipse
Sacha Wolski
<http://www.laliluna.de>