ATICA 2010

II Jornadas Nacionales sobre Aplicación de Tecnologías de la Información y Comunicaciones Avanzadas

25 y 26 de Enero de 2010

II Jornadas Nacionales sobre Aplicación de las Tecnologías de la Información y Comunicaciones Avanzadas (ATICA 2010)

Organizadas por:

Actas de las
II Jornadas Nacionales sobre Aplicación de las Tecnologías de la Información y Comunicaciones Avanzadas (ATICA 2010)

Escuela Técnica Superior de Ingeniería Informática
Universidad de Alcalá
Alcalá de Henares (Madrid)
25 y 26 de Enero de 2010

Editores de la presente edición:

Pedro Antonio de Alarcón Marín
Roberto Barchino Plata
Luis Bengochea Martínez
José Antonio Gutiérrez de Mesa
José María Gutiérrez Martínez
José Ramón Hilera González
José Javier Martínez Herraiz
Salvador Otón Tortosa

Editorial: Servicio de Publicaciones - Universidad de Alcalá

Universidad de Alcalá
SERVICIO DE PUBLICACIONES

El contenido de este libro no podrá ser reproducido,
ni total ni parcialmente, sin el previo permiso escrito del editor.
Todos los derechos reservados

@ Universidad de Alcalá
Servicio de Publicaciones
Plaza de San Diego, s/n
28801 Alcalá de Henares
www.uah.es

ISBN: 978-84-8138-856-5

Depósito Legal: M-2363-2010

Impresión y encuadernación: Imprenta UAH
Impreso en España

Presidencia de honor

Virgilio Zapatero Gómez (Rector de la Universidad de Alcalá)
Octavio Granado Martínez (Secretario de Estado de la Seguridad Social)

Presidencia institucional

Michel Heykoop Fung-a-You (Vicerrector de la Universidad de Alcalá)
Eladio Quintanilla Rojo (Gerente de Informática de la Seguridad Social)

Presidencia de las Jornadas

José Ramón Hilera González (Universidad de Alcalá)
Francisco Javier Santamaría Zapata (Seguridad Social)

Comité de Organización

Presidentes:

Luis Bengochea Martinez (Universidad de Alcalá)
Pedro Antonio de Alarcón Marín (Seguridad Social)

Miembros:

Roberto Barchino Plata (Universidad de Alcalá)
José Antonio Gutiérrez de Mesa (Universidad de Alcalá)
José María Gutiérrez Martínez (Universidad de Alcalá)
José Ramón Hilera González (Universidad de Alcalá)
José Javier Martínez Herraiz (Universidad de Alcalá)
Salvador Otón Tortosa (Universidad de Alcalá)
Juan Antonio Rodrigo Yanes (Universidad de Alcalá)
Blanca Menéndez Olías (Fundación General de la Univ. de Alcalá)
Ana María Privado Rivera (Fundación General de la Univ. de Alcalá)
Olga Carazo López (Seguridad Social)
Mónica Rodríguez Rodríguez (Seguridad Social)
Mª Pilar Ordoño Álvarez (Seguridad Social)
Mª José Martínez Sánchez (Seguridad Social)

Comité Científico

Presidentes:

José Antonio Gutiérrez de Mesa (Universidad de Alcalá)
Francisco Delgado Azuara (Seguridad Social)

Miembros

Alfonso López Baca (Universidad de Alcalá)
Andrés Hermoso Arnaez (Seguridad Social)
Ángel Fernández Álvarez (Universidad de Alcalá)
Angel Francés (Universidad de Zaragoza)
Angel José Perez Izquierdo (Seguridad Social)
Antonio Moratilla Ocaña (Universidad de Alcalá)
Apolonia Martínez Nadal (Universidad Islas Baleares)
Carlos Corral Mata (Seguridad Social)
Carmen Pagés Arévalo (Universidad de Alcalá)
Daniel Rodríguez García (Universidad de Alcalá)
David Castro Esteban (Universidad de Alcalá)
Eladio Domínguez (Universidad de Zaragoza)
Elena Davara Fernández de Marcos (Davara & Davara Asesores Jurídicos)
Eugenio Bezares Ruiz (Seguridad Social)
Francisco Javier Bueno Guillén (Univ. de Alcalá)
Gertrudis López López (Univ. Central de Venezuela)
Javier Alonso García (Seguridad Social)
Javier De Pedro Carracedo (Universidad de Alcalá)
Javier Fernández Fernández (Seguridad Social)
José Antonio de Frutos Redondo (Univ. de Alcalá)
José Antonio Pámies Guerrero (Univ.de Alcalá)
José Carlos Holgado (Universidad de Alcalá)
José Ignacio Pérez Sanz (Universidad de Alcalá)
José Javier Martínez Herráiz (Universidad de Alcalá)
José Luis Cuadrado Garcia (Universidad de Alcalá)
José María Gutiérrez Martínez (U. de Alcalá)
José Ramón Hilera González (Universidad de Alcalá)
José Raul Durán Diaz (Universidad de Alcalá)
José Raúl Fernández del Castillo (Universidad de Alcalá)
Juan Antonio Rodrigo Yanes (Universidad de Alcalá)
Juan José Cuadrado Gallego (Uerdad de Alcalá)
Llorenç Huget Rotger (Universidad Islas Baleares)
Luis Bengochea Martínez (Universidad de Alcalá)
Luis Fernández Sanz(Universidad de Alcalá)
Luis Usero Aragones (Universidad de Alcalá)
M.José Domínguez Alda (Universidad de Alcalá)

Mª Antonia Zapata (Universidad de Zaragoza)
Mª Concepción Antón García (Seguridad Social)
Mª Jesús Lapeña (Universidad de Zaragoza)
Manuel Pérez Santander (Universidad de Alcalá)
Mario Triguero Garrido (Universidad de Alcalá)
Miguel Angel Davara (Davara & Davara Asesores Jurídicos)
Miguel Angel Navarro Huerga (Univ.de Alcalá)
Oscar Rebollo Martínez (Seguridad Social)
Pedro Valcarcel Lucas (Seguridad Social)
Rafael Cambralla Diana (Universidad de Alcalá)
Rafael Rico López (Universidad de Alcalá)
Raúl V. Ram´rez Velarde (Universidad de Alcalá)
Roberto Barchino Plata (Universidad de Alcalá)
Salvador Gómez Pedraz (Universidad Carlos III)
Salvador Otón Tortosa (Universidad de Alcalá)
Sonia Miraut Martín (Seguridad Social)
Teresa Díez Folledo(Universidad de Alcalá)

Prólogo

Como fruto de la colaboración entre la Universidad de Alcalá y la Secretaría de Estado de la Seguridad Social, surgen estas **"II Jornadas Nacionales sobre Aplicación de Tecnologías de la Información y Comunicaciones Avanzadas (ATICA 2010)",** con el objetivo de presentar y poner en común trabajos y experiencias en el ámbito de la aplicación de las Tecnologías de la Información y las Comunicaciones (TICs), que puedan ser de utilidad a los asistentes, aportando ideas y soluciones a problemas reales relacionados con diferentes aspectos de la utilización de estas tecnologías en entornos de gestión, en general, y de gestión de la Seguridad Social, en particular.

Las áreas de interés de las Jornadas incluyen las siguientes, aunque no están limitadas a ellas: la Ingeniería del Software, la Ingeniería Web, las comunicaciones y redes de ordenadores, la administración de sistemas informáticos, y la aplicación de las TICs en el ámbito de la Seguridad Social.

En relación con el área de la Ingeniería del Software, en este libro de actas se recoge la descripción de trabajos de desarrollo de aplicaciones reales en los que se han utilizando herramientas, métodos y tecnologías para la automatización de las actividades de análisis, diseño, construcción, implementación, pruebas e implantación de los productos software presentados por sus autores. Además de aplicaciones de gestión para su ejecución en un entorno de escritorio, también pueden encontrarse en el libro complejas aplicaciones Web basadas en las tecnologías más avanzadas, que nada tienen que ver con las clásicas páginas Web, y para cuyo desarrollo los autores han seguido un enfoque de ingeniería, aplicando la metodología de Ingeniería Web más adecuada en cada caso.

Otros trabajos están relacionados con las comunicaciones y la construcción, gestión, configuración y verificación de redes de ordenadores; así como con la administración de sistemas informáticos, tanto en lo que respecta a la instalación, configuración y mantenimiento de sistemas operativos, como de gestores de bases de datos.

Aunque los trabajos incluidos en este libro de actas pueden ser de aplicación en cualquier ámbito, la mayor parte de ellos pueden orientarse a su aplicación al Sistema de la Seguridad Social, colaboradora en la organización de estas jornadas. Además, el objetivo de las propuestas y estudios realizados por los diferentes autores es que sus trabajos puedan servir de referencia y aportar nuevas ideas en relación con la aplicación de las tecnologías de la información y las comunicaciones en el ámbito de la Seguridad Social. Para ello, en algunos casos se ha recurrido a plantear supuestos utilizando información y escenarios (organizaciones, unidades o departamentos de informática) ficticios, pero suficientemente concretos en el marco de sus competencias, por lo que es posible que el lector pueda asociarlos, por pura coincidencia, a otros conocidos.

Estas jornadas han sido una realidad gracias a la estrecha colaboración entre la Universidad de Alcalá y la Secretaría de Estado de la Seguridad Social, que se ha materializado a través de la Gerencia de Informática de la Seguridad Social. Y que tuvo su origen en el convenio marco de colaboración suscrito por el Rector de la Universidad y el Secretario de Estado de la Seguridad Social el 5 de diciembre de 2006; y que, con toda seguridad, permitirá la realización de nuevas actividades conjuntas tan enriquecedoras como ésta para los profesionales de ambas entidades.

Los presidentes de las jornadas ATICA 2010.

José Ramón Hilera González
Departamento de Ciencias de la Computación
Universidad de Alcalá

Francisco Javier Santamaría Zapata
Gerencia de Informática de la Seguridad Social
Secretaría de Estado de la Seguridad Social

Índice de Contenidos

Prólogo
José Ramón Hilera González; Francisco Javier Santamaría Zapata 9

Ponencias

Diseño de un componente para la normalización de las comunicaciones del Instituto Nacional de la Seguridad Social (INSS) al exterior
Fernando Alonso Fernández
Rosario Vuelta Larrea 15

Sistema de Gestión de Agrupaciones Musicales
Antonio Asensio Portilla
Enrique Fort Roig 21

Sistema de Gestión Bancaria (SGB)
Jesús Mª Ballesteros de Diego 29

Arquitectura de acceso seguro a red basada en servidor VPN
Fernando Casas Grande
Mª Dolores García Ureña Ruiz de Salazar González 35

Aplicación informática para la gestión de los usuarios de un Sistema de Envío de Documentos
Alberto Corral Garrido 43

Desarrollo de una Herramienta para la Migración de Bases de Datos de Microsoft Access a Oracle.
Antonio Echevarrías Escuder 51

Proyecto de Gestión de Ficheros Adabas
Ivan Eguilleor Villena 59

Sistema para la gestión de autorizaciones de aplicaciones a usuarios en la intranet provincial de Huesca. Aplicación AUTAPLI
Mª José Foncillas Sanz 69

Sistema de Gestión de Vacaciones
Mª del Carmen García Alcaraz 71

Sistema de Gestión de Equipos Informáticos Localizador de Elementos en Oficinas (LEO)
Aurelio Gonzalo de Francisco 83

Desarrollo Aplicación Web. Net “MIALMA”
José Carlos Hernández Roldán 93

Gestión de Líneas
Héctor López García 101

CitaMed-Aplicación web para el control de las citas previas con el Servicio Médico de Empresa
Carolina Loza Eguiluz 103

Sistema de Gestión de Confidencialidad
Ana Moreno Gracia
Antonio Fernández Labrador
Susana López Reche 111

Acceso Nacional a SARTIDO
Juan Carlos Ogueta Sáenz 117

Sistema de Gestión de Permisos de Sartido. Aplicación PERSAR
Maria Luisa Osuna Mora
Coro García Arenaza 119

Gestión del Conocimiento Calidad
Antonio Jesús Pérez Reina 121

bd3p. Base de Datos Documental para Direcciones Provinciales
Ángel Punzano Martínez 131

Gestión de los documentos archivados
Esteban Sebastián Marco 139

Aplicación Web (Java EE) para la gestión integral de un almacén de artículos de una Entidad
Luís Miguel Soria Duarte 149

Diseño de una base de datos para gestionar el departamento de salud laboral y seguridad e higiene en el trabajo (GETSALA)
María Antonia Trejo Rodríguez 159

Sistema de Gestión Inmobiliaria: "Inmobiliaria Alcalá"
Juan Villa Martínez
Jesús Resina Hernández 169

Notas

Ponencias

Diseño de un componente para la normalización de las comunicaciones del Instituto Nacional de la Seguridad Social (INSS) al exterior

Fernando Alonso Fernández
Rosario Vuelta Larrea

Dirección Provincial del INSS de Barcelona

Resumen: Diseño e implementación de un componente ActiveX para la normalización de los oficios y resoluciones de acuerdo con el Manual de Imagen Institucional de la Administración General del Estado, aprobado por Resolución de 9 de marzo de 2005 y modificado por la Resolución de 2 de abril de 2007.

1. Introducción

Una de las necesidades que tiene cualquier empresa, en nuestro caso, el Gobierno de España, y más concretamente, el INSS, es el de la normalización de las comunicaciones con el exterior.

El Manual de Imagen Institucional de la Administración General del Estado (AGE) dio las pautas necesarias para potenciar una imagen única y evitar la confusión y falta de identificación del organismo emisor de las comunicaciones.

Como desarrolladores de aplicaciones de gestión y dependientes del INSS, con anterioridad a este proyecto nos veíamos obligados a crear una imagen parecida al modelo oficial imperante en cada momento, que debía ser adaptado a las características propias de la Dirección Provincial a la que pertenecemos (imagen, idioma, diversas agencias –CAISS-).

A partir de la entrada en funcionamiento del citado Manual, hubo modificaciones, las cuales tuvimos que recoger en cada una de nuestras numerosas y diversas aplicaciones.

La formación que hemos recibido a través del Proyecto Ática nos animó a crear este componente ActiveX para facilitar a los desarrolladores la normalización de las comunicaciones al exterior.

2. Descripción

El sistema EscritosFR de emisión de escritos se compone de un archivo ocx ActiveX para incorporar en los desarrollos y aplicaciones que generan escritos.

Para el desarrollo de este sistema hemos usado las herramientas informáticas utilizadas habitualmente y contempla las normas de Imagen Institucional para la generación de escritos.

Para trabajar con este componente, el fichero OCX ActiveX EscritosFR.ocx debe estar copiado en el pc (c:\windows\system32) y además, en tiempo de diseño, se debe incluir la referencia en Visual Studio. Para ello hay que entrar en la opción Componentes del menú Proyecto y marcar el ocx EscritosFR.

Cada nueva versión del ocx se deberá copiar en c:\windows\system32.

3. Fases

Aquí se describen las herramientas de que dispone el desarrollador para utilizar el componente.

La forma de trabajar este componente tiene dos funciones o fases principales:

- Toma de valores en los diferentes atributos
- Ejecución de los métodos de impresión

Una vez referenciado el componente en el programa (Proyecto/Componentes/EscritosFR.ocx),

3.1 Código a implementar (Toma de valores)

Con el atributo referido a la provincia (zBprovincia) que emite los escritos, se cargan los datos con relación al idioma, bilingüismo y demás particularidades de esa provincia.

Los datos del código de barras SARTIDO (zDcodigobarras), el código de acuse de recibo (zGcodigoacuse), el registro de salida SICRES (zFsicres1, zFsicres2, zFsicres3) y el correo electrónico (zDcorreoelectronico), no son imprescindibles para el componente, ya que se ha visto en la práctica que no todas las Direcciones Provinciales los incorporan en sus comunicaciones, por lo tanto quedan a criterio del desarrollador.

El tipo de documento (zEtipodocumento) nos permite distinguir entre una comunicación con formato de oficio o resolución, tanto dirigido a un cliente como a un organismo o empresa.

El resto de los atributos del componente serán asignados por el desarrollador en función de la comunicación que desee emitir.

```
With EscritosFR
  .zBprovincia = 8
  .zCdireccion = "Sant Antoni M. Claret, 5-11"
  .zCtelefono = "93 284 93 58"
  .zCfax = "93 000 00 00"
  .zCzippoblacion = "08037 Barcelona"
  .zDcodigobarras = "08;0555555;4989849;849494;94949"
  .zDcorreoelectronico = "seguridadsocial@seg-social.es"
  .zEtipodocumento = 1
  .zEseccion = "Jubilaciones VI"
  .zEiniciales = "XX/YY"
  .zEreferencia1 = "0-2015/000000-00"
  .zEreferencia2 = "DNI 00.000.000-T"
  .zFsicres1 = "INSS BCN"
  .zFsicres2 = "INSS BAR-9999"
  .zFsicres3 = "99999999999999 12/12/2012 12:12:12"
```

```
  .zGcodigosicres = "99.9"
  .zGdestinatario1 = "Nombre Apellido1 Apellido2"
  .zGdestinatario2 = "C./ Del Rorsario Verde Azulado, 34, 3º 3ª"
  .zGdestinatario3 = "08099 Barcelona"
  .zGcodigoacuse = "08;2666464;466464;646465464;465465"
End with
```

En este momento, el componente está cargado con todos los valores necesarios para ejecutar los métodos de impresión.

3.2 Código a implementar (Ejecución de los métodos de impresión)

Las comunicaciones al exterior pueden constar de una o varias páginas. Los métodos para obtener la copia impresa son estos tres:

Con el método **ImprimirPrimeraPagina** creamos la imagen con los datos obtenidos. Esta imagen la enviamos a la impresora y queda en espera de los datos que pueda o no querer enviar el desarrollador.

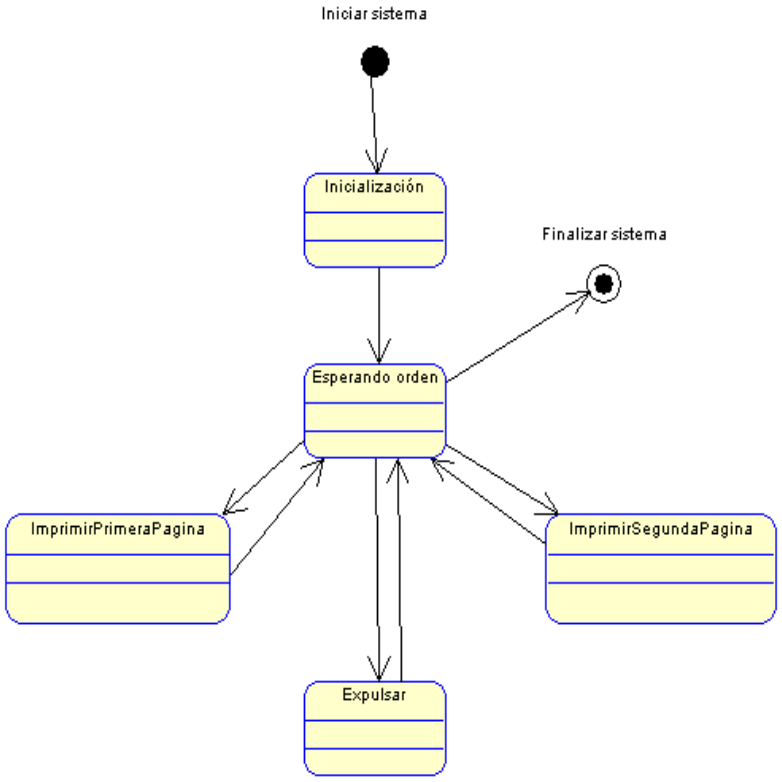
Una vez completada la imagen de la primera página, se deberá ejecutar el método **Expulsar** para obtener físicamente la copia impresa de la imagen enviada a la impresora.

```
With EscritosFR
  .ImprimirPrimeraPagina
  .Expulsar
End With
```

En el caso de querer imprimir una segunda página, el atributo **zEreferencias2pagina** deberá estar activado y se deberá ejecutar el método **ImprimirSegundaPagina** y proceder como en la primera página.

```
With EscritosFR
  .zEreferencias2pagina = True
  .ImprimirSegundaPagina
  .Expulsar
End With
```

3.3 Diagrama de Estado



4. Valoración personal final

Creemos que es una muy buena iniciativa y el esfuerzo del desarrollo de este sistema se verá en breve amortizado con la puesta en marcha por parte de todas las las dependencias del INSS en España.

5. Conclusiones y trabajo futuro

El sistema utilizado para el desarrollo de este componente sirve para implementarlo en cualquier otra Entidad u Organismo gestor de la AGE con muy pequeñas variaciones, que se podría llevar a cabo a requerimiento de cada Entidad.

La labor de distribución de este componente a todas las dependencias del INSS es la principal acción para que este sistema cobre trascendencia.

6. Referencias

- Intranet:

  http://imagen.map.es

- Normativa sobre Imagen Institucional:

REAL DECRETO 1465/1999, de 17 de septiembre, por el que se establecen criterios de imagen institucional y se regula la producción documental y el material impreso de la Administración General del Estado (BOE. 25.09.99)

ORDEN de 27 de septiembre de 1999 por la que se aprueba el Manual de Imagen Institucional de la Administración General del Estado y se dictan normas de desarrollo del Real Decreto 1465/1999, por el que se establecen criterios de imagen institucional y se regula la producción documental y el material impreso de la Administración General del Estado. (BOE 28.09.99)

Referencia del Consejo de Ministros de 17 de septiembre de 1999

Resolución de 2 de abril de 2007, de la Secretaría General para la Administración Pública, por la que se modifica el Manual de Imagen Institucional de la Administración General del Estado y la Guía para la edición y publicación de páginas web en la Administración General del Estado aprobada por Resolución de 9 de marzo de 2005 de la Secretaría General para la Administración Pública (BOE 16.04.07)

Sistema de Gestión de Agrupaciones Musicales

Antonio Asensio Portilla, Tesorería General de la S.S. Dirección Provincial Valencia.
Enrique Fort Roig, Unidad Provincial de Informática de Valencia.

Resumen. El trabajo administrativo y burocrático en una Sociedad Musical es mayoritariamente voluntario, por parte de las personas que la integran, por ello, los dirigentes de la Sociedad deciden facilitar a los miembros de la misma estas tareas. La solución pasa por aprovechar el uso de las nuevas tecnologías Web y de esta forma automatizar aquellos trámites, sencillos a primera vista, pero incómodos en el quehacer diario.
La entrada en el mundo de Internet, conlleva ventajas y obligaciones. Ventajas, para la Sociedad el hecho ofrecer estos servicios a través de internet, aumenta el valor añadido de la misma, para sus miembros, sencillez y comodidad en los trámites, para todos, disponer de una información actualizada y fiable.
Y si hablamos de obligaciones, la seguridad es la piedra angular de una aplicación Web. La calidad del servicio, pieza fundamental. Si el servicio no es bueno, el esfuerzo es vano. El interfaz, grato e intuitivo, los usuarios no son expertos y hay que pensar a quien va dirigido. Y por último, y no menos importante, el respeto a la legalidad en el uso de la información.
Decididos a afrontar el riesgo, se estudia y posteriormente se pone en marcha el Sistema de Gestión de Agrupaciones Musicales (SGAM).

1. Introducción

La Sociedad Musical es un conjunto de personas cuya finalidad es potenciar y dar a conocer la música. Para ello dispone de un centro de formación y diversas agrupaciones musicales especializadas y compuestas por miembros de la Sociedad.
Se pretende crear un sistema informatizado para manejar la información sobre las personas que integran la Sociedad Musical, los diversos grupos que la componen, así como los miembros de los diferentes grupos.
El sistema estará supervisado por un administrador que tendrá acceso a toda la información. Será el encargado de dar de alta a las personas, la modificación de sus datos, la creación de grupos y el control de los accesos al sistema, así como la competencia exclusiva sobre la eliminación de datos sobre sus miembros de acuerdo con la Ley de Protección de Datos.
La sociedad esta distribuida en grupos artísticos compuestos por miembros de la sociedad, teniendo cada grupo un responsable que será miembro del grupo.
Serán tareas del responsable de grupo el alta y mantenimiento de los miembros de su grupo. Anotará la baja cuando un miembro deje de pertenecer al mismo. No se eliminarán miembros ya que se pretende mantener un registro histórico.

El propio responsable de grupo asignará a la persona que lo sustituya en dicha función.
Los miembros de la Sociedad podrán consultar y actualizar sus datos personales, excluido el NIF y teléfonos de contacto.
Estará disponible para todos los miembros la información sobre los grupos existentes en la Sociedad y el detalle de los miembros que componen cada grupo.
La Sociedad desea que la información sea accesible a través de una página web, donde los integrantes de la misma tengan acceso a la información existente, manteniendo los controles de seguridad necesarios.
Para ello el acceso al sistema será mediante la identificación del usuario, a su vez dado de alta en el sistema y una contraseña.

2. Especificación funcional del sistema.

Por lo expuesto anteriormente, se plantea la aplicación en base a varios perfiles de usuario, condicionando la modificación o la eliminación de los datos al perfil que acceda al sistema.

Funciones básicas del sistema:

- Control de acceso al sistema según perfil del usuario.
- Gestión de las personas.
- Gestión de los grupos.

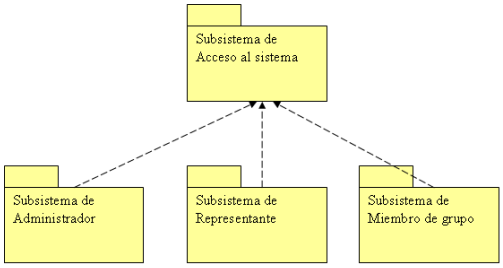
Perfiles de usuarios y opciones específicas:

- ADMINISTRADOR (perfil máximo):
  - Mantenimiento y control del acceso al sistema.
  - Alta, consulta, modificación y eliminación de personas.
  - Alta, consulta, modificación y eliminación de grupos.
  - Alta, consulta, modificación y eliminación de miembros.
  - Asignación de nuevo administrador.
  - Resto de accesos de perfiles inferiores.
- RESPONSABLE DE GRUPO (perfil medio):
  - Alta, consulta y modificación de miembros.
  - Cambio de responsable del grupo.
  - Resto de accesos de perfiles inferiores.
- MIEMBROS DE LOS GRUPOS (perfil bajo):
  - Consulta y modificación de sus datos personales.
  - Alta, consulta, modificación y eliminación de sus teléfonos de contacto.
  - Consulta de los grupos y sus miembros.

Se ha confeccionado un modelo de ficha para la recogida de requisitos, divididos en 4 categorías: requisitos funcionales, de seguridad , de datos y de interfaz.

3. Desglose en subsistemas.

El Sistema se descompone en cuatro subsistemas.



Un subsistema de "**acceso al sistema**" encargado de controlar si el usuario tiene permiso para entrar en el sistema y en su caso le asigna el perfil correspondiente.

Un subsistema de "**administrador**", ligado al perfil del mismo nombre. Tendrá como objetivo el control de usuarios, la gestión de personas, la gestión de grupos y la gestión de datos de miembros.

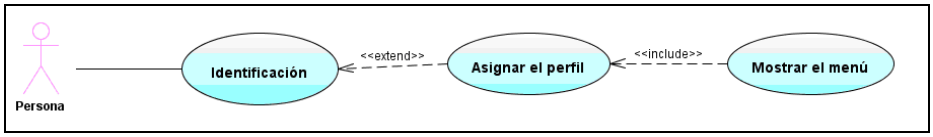
Un subsistema de "**representante de grupo**", también ligado al perfil del mismo nombre y que será el encargado de administrar el grupo.

Un subsistema de "**miembro de grupo**". Mostrará la información sobre los datos personales del usuario, grupos existentes y los miembros que lo componen.

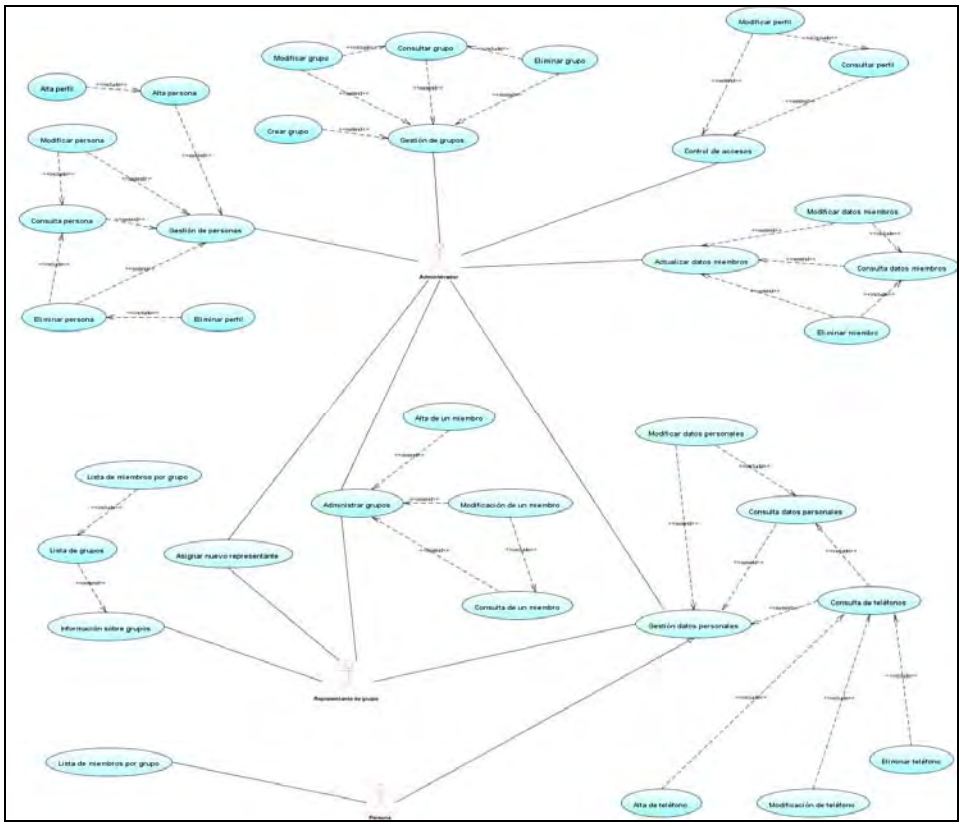
4. Modelo de comportamiento

Se diferencian dos casos de uso:

El primero es el encargado de comprobar el usuario y asignar el perfil correspondiente.



El segundo caso de uso muestra el comportamiento general de la aplicación después de asignar el perfil.



5. Requerimientos técnicos.

Para el desarrollo del sistema descrito en el presente documento se utilizarán tecnologías Java mediante el desarrollo de una aplicación Web en la que el usuario utilice un navegador de Internet para interactuar con la aplicación bajo Windows XP.
El IDE utilizado para el desarrollo ha sido Netbeans 6.5 de Sun Microsystem.
El gestor de base de datos utilizado será el integrado en el IDE Netbeans 6.5 la base de datos Derby de Apache.
Se ha utilizado el patrón MVC (Modelo Vista Controlador) .
Código HTML con controles en JavaScript y CSS embebidos en archivos JSP, SERVLET's con tecnología EJB y entorno de Gestor de Base de Datos para almacenar la información.

6. Diseño de la arquitectura del sistema.

El sistema contará con un servidor y ordenadores PC que actuarán como clientes. Los PCs serán utilizados por los diferentes usuarios de la aplicación para ejecutar la misma, cada uno de ellos para poder funcionar deberá conectarse al servidor, el cual se encargará de controlar todas las operaciones. En el servidor residirá el gestor de base de datos que contiene toda la información de la aplicación, también se instalaran los componentes que añaden la lógica necesaria para el control de las conexiones de los equipos clientes y los componentes para realizar las operaciones solicitadas por los clientes.

7. Descripción del diseño del sistema.

El acceso al sistema se efectúa mediante la identificación del usuario que desea conectarse. Para ello el sistema le muestra la pantalla de inicio y posteriormente la pantalla de validación. Realizada la autenticación del usuario, se le ofrece un menú con las opciones disponibles según el perfil de usuario.

Para realizar este proceso la tabla “menuopciones”, identifica el rol del usuario y las aplicaciones disponibles según el mismo. A partir de aquí se construye el menú
Este sistema tiene la ventaja que permite cambiar los roles de las opciones e insertar nueva opciones sin cambiar el código. Se mantiene el menú está descrito en el punto 2.

8. Diseño de la base de datos.

La base de datos se compone de las siguientes tablas:

TABLA	**Descripción**
MENUOPCIONES	Contiene el rol, el texto de menú, y la aplicación que lanza.
PERSONAS	Contiene los datos de las personas de la Sociedad
GRUPOS	Identificación de los grupos, persona responsable.
TELÉFONOS	Contiene los teléfonos de las personas (se decide crear una tabla de teléfonos, asociada a personas, para evitar limitar su información)
CONTROLACCESOS	Contiene la identificación del usuario para el acceso al sistema.
MIEMBROS	Contiene los miembros que integran los grupos.

Detalle de la tabla MENUOPCIONES mencionada en el punto 7.

Num	OPCION DEL MENU	PERFIL			APLICACION
1	Gestión de miembros de la sociedad	1			nuevoMiembro.jsp
2	Crear nuevo grupo	1			nuevoGrupo.jsp
3	Actualizar datos de los miembros de la agrupación	1			ServletVerDatosMiembros
4	Modificar responsable de la agrupación	1	2		ServletCambioRepGrupos
5	Incluir nuevo miembro a la agrupación	1	2		ServletBuscaGrupo
6	Consulta/actualización de miembros	1	2		ServletBuscaGrupoAct
7	Consulta/actualización de datos personales	1	2	3	ServletConsultaPersona
8	Consultar agrupaciones y miembros de cada agrupación	1	2	3	ServletConsultaGrupos

Diagrama Entidad Relación

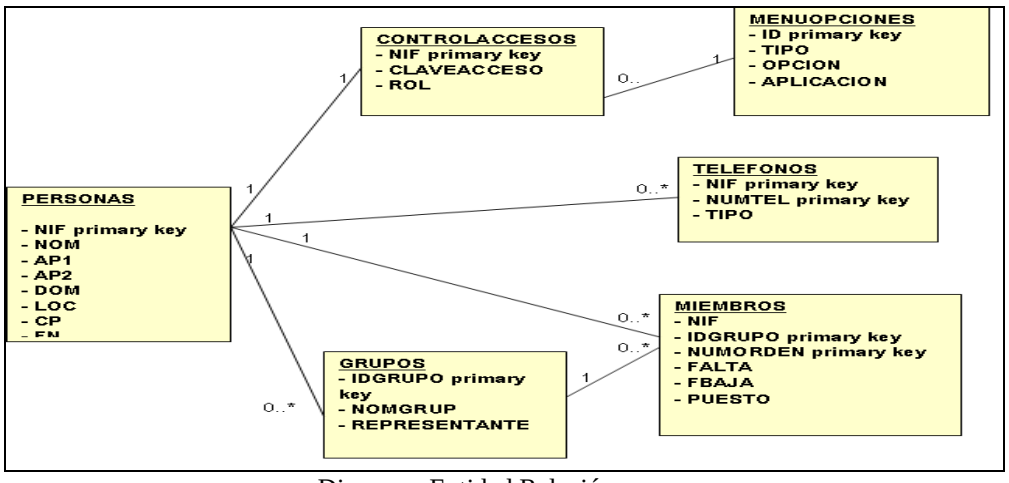
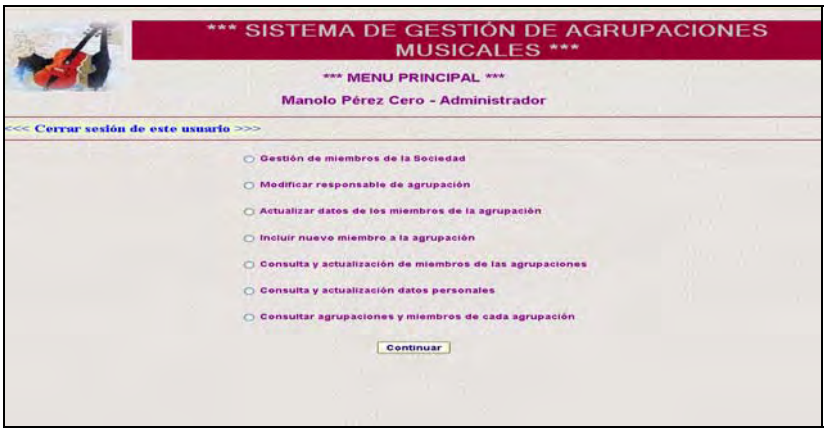


Diagrama Entidad Relación

9. Manual de usuario.

Se confecciona el manual de usuario que describe el uso de la herramienta, analizando los menús, pantallas y descripción de los campos. Dicho manual contempla los tres roles establecidos en la aplicación.



Pantalla perfil de Administrador.

Pantalla de introducción de datos personales

Pantalla de cambio de representante.

Esta pantalla tiene la peculiaridad de que cuando se efectúa el cambio de representante, realiza un cambio de perfiles, pasando el anterior representante a perfil bajo y el nuevo a perfil medio. Con el cambio de representante cierra la sesión para que el acceso sea por el perfil correcto. Cuando el cambio lo realiza el Administrador cambia los perfiles pero no cierra la sesión ya que dispone del perfil alto.

10. Pruebas.

El sistema se somete a un plan de pruebas realizado a la terminación de la codificación de cada subsistema para verificar si cumple las especificaciones citadas.
Finalizada la codificación de la aplicación, con las tablas de la base de datos vacías y en el orden previamente establecido, se han realizado pruebas para identificar posibles errores de acceso, confección correcta del menú según el rol, cambios de rol en los usuarios, controles en los datos introducidos, verificación de los mensajes de error al usuario e identificación del acceso no permitido fuera de sesión.

11. Conclusiones.

Concluido el proyecto y realizadas las pruebas, se han cumplido los objetivos iniciales previstos. El proyecto se ha pensado para incrementar los servicios y ofrecer nuevas expectativas.
La realización de este proyecto ha supuesto una nueva experiencia y un reto debido a la novedad en el uso de esta tecnología para realizar aplicaciones.

12. Referencias.

Documentación entregada en Plan Atica
Programación y código fuente. http://sbcodigo.com/java/
Java en castellano. http://www.programacion.com/java/
HTML en castellano. http://www.programacion.com/html/
Sun Microsystems. http://es.sun.com/
Programación Fácil. http://www.programacionfacil.com/java_jsp/start
Java.com: El centro de la tecnologia Java. http://www.java.com/es/
JavaHispano: Comunidad de Java. http://www.javahispano.org/
World Wide Web Consortium (W3C). http://www.w3.org/
Desarrollo Web. http://www.desarrolloweb.com/
El código: Javascript. http://www.elcodigo.net/
WebEstilo. http://www.webestilo.com/
Netbeans. http://www.netbeans.org/index.html

Sistema de Gestión Bancaria (SGB)

Jesús Mª Ballesteros de Diego

Centro Informático del I.N.S.S.
Gerencia de Informática de la Seguridad Social,

1. Introducción

De forma general, el SGB (Sistema de Gestión Bancaria) permitirá gestionar una serie de cuentas que posee un banco, en la que se podrán realizar diversas operaciones financieras. El empleado se encargará de gestionar la información de las distintas cuentas que posee el banco.
El Empleado está a cargo de crear/introducir la información de las cuentas que se gestionan en el banco.
Todas las operaciones se realizaran por la Intranet mediante un navegador.

2. Descripción

El sistema debe permitir las siguientes funciones a un usuario Empleado:

- Creación / modificación de los datos de un cliente.
- Creación / modificación de los datos de una cuenta.
- Listado de cuentas.
- Listado de clientes.

3. Datos que caracterizan el sistema

Cliente
El empleado del banco rellenará los siguientes datos de un cliente.

- DNI del cliente.
- Nombre del cliente.
- Dirección del cliente.
- Teléfono Contacto.

Cuentas
El sistema bancario posee una serie de cuentas que pertenecen a los clientes. Los datos a guardar para cada cuenta son los siguientes:

- Número de cuenta: Es un identificador único asignado por la aplicación cuando se da de alta una cuenta. Se asignara automáticamente.

- Fecha de alta: La fecha en que se da de alta la cuenta en la aplicación.
- Cliente: DNI del cliente al que pertenece la cuenta.
- Saldo: En euros.
- Activa: Dato que indica que la cuenta no ha sido cancelada.
- Fecha de baja: Fecha en que ha sido cancelada una cuenta. Si la cuenta no está activa.
- Observaciones: Texto libre para indicar observaciones de la cuenta.

4. Requisitos.

Al empleado se mostrara un menú donde podrá elegir entre las distintas tareas que puede realizar.

El empleado podrá dar de alta a clientes.

El empleado podrá modificar los datos de las cuenta de los clientes.

Se podrá listar los datos de los clientes junto con su saldo total

Se podrá listar los datos de todas las cuentas.

Para incluir un cliente, se introducirán los datos relativos al cliente, que son: D.N.I. Del cliente, su nombre, su dirección y su teléfono.

Cuando se da de alta a un cliente, se le genera automáticamente una nueva cuenta con una fecha de alta del día en que se efectúa y un saldo inicial de cero. El número de la cuenta lo generará el sistema automáticamente.

Un mismo cliente podrá tener varias cuentas.

No se podrá dar de baja a clientes, se considera que si un cliente ha tenido cuentas en el sistema, estas estarán desactivadas, pero los datos se mantendrán por si alguna vez quiere volver a activar una cuenta.

Se podrá dar de baja una cuenta (desactivar).

Cuando se desactive una cuenta, se guardará la fecha en la que se hace.

Se podrá volver a activar una cuenta previamente dada de baja (activar).

En el caso anterior, la fecha de alta continuará siendo la misma en la que se dio de alta la cuenta, y la fecha de baja se borrará.

Se podrá ingresar dinero.

Se podrá sacar dinero.

Se podrán guardar observaciones sobre las cuentas, así como modificar dichas observaciones.

Si se crea una nueva cuenta para un cliente, esta se creará de la misma forma que cuando crea la cuenta inicial del cliente.

No se podrá tener una cuenta si no se es cliente.

Cuando una cuenta se de baja (desactive) esta se queda con saldo 0. Se supone que se ha sacado el saldo restante.

Cuando una cuenta se vuelve a activar, su saldo inicial es 0.

No se podrá sacar más efectivo que el total del saldo de la cuenta.

5. Análisis.

La aplicación está dividida en tres capas que son presentación, negocio y acceso a datos.

De esta forma las posteriores modificaciones en los requisitos están más definidas en cada una de sus capas.

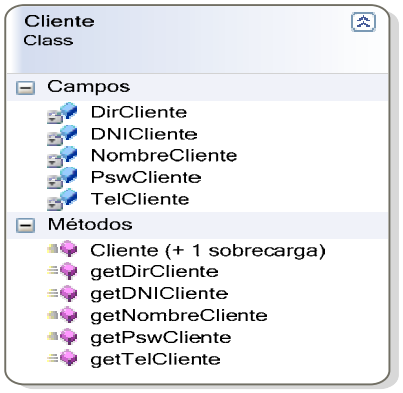
La parte de presentación está realizada mediante paginas web .aspx que integran código HTML con C#.

La capa de negocio esta compuesta por las clases que sirven de base para mediar entre los datos de la presentación y los datos que se deben mantener guardados en la base de datos. Están programadas en C#.

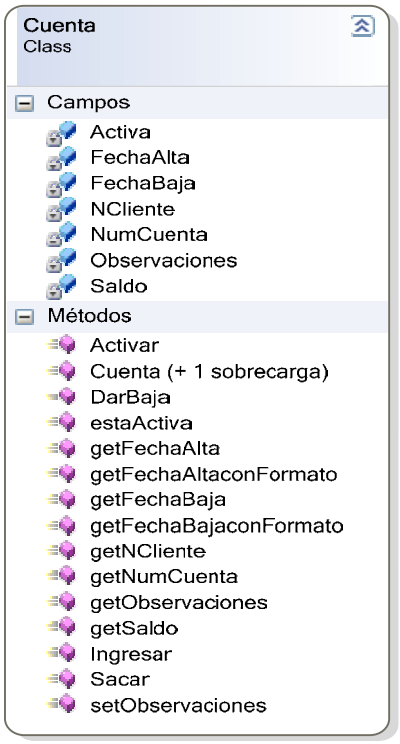
La capa de acceso a datos contiene las clases necesarias para acceder a los datos de forma que cualquier modificación en la ubicación de la base de datos, o las modificaciones en la misma, supongan un mínimo coste de modificaciones al resto de capas. Están programadas en C#.

6. Diseño.

– Cliente:

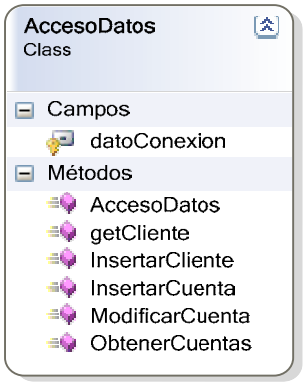


– Cuenta:



.

– AccesoDatos.



–

7. Diseño de la Base de Datos.

La Base de datos para conservar los datos necesarios para el sistema de gestion bancaria utilizará dos tablas:

– La tabla de Clientes.

- DNI del cliente. Guardará número de DNI y su letra de control. Este dato será de 10 posiciones alfanuneméricas; hasta 9 números y una letra al final.
- Nombre del cliente. Se debe guardar el nombre del cliente, para ello se reservaran 50 caracteres alfanuméricos.
- Dirección. 50 posiciones alfanumericas donde se podrá guadar la direccion del cliente.
- Teléfono. Puesto que no se va a operar con este dato, se guardaran 9 posiciones alfanuméricas.
- Clave del cliente. Aunque en este proceso no se utilize, se guardará esta información para un posible uso (se inicializará vacio). Se reservará 10 caracteres para su uso posterior.

– La tabla de Cuentas.

- Numero de la cuenta. Se genera automaticamente de forma que no es necesario introducirlo en el sistema. Emezará desde 1 y tendrá capacidad para almacenar hasta 999.999.999.999.999.999 cuentas bancarias. 18 posiciones numéricas.

Es clave de acceso única. No habrá ningun número de cuenta repetido.

- Identificador del cliente. Se corresponde con un DNI de un cliente. Es clave de acceso para las cuentas de un mismo cliente. Un cliente podrá ser titular de varias cuentas, por lo que el identificador puede estar repetido en la tabla de cuentas. El cliente deberá estar en la tabla de clientes.
- Fecha de alta. Se genera automáticamente cuando se crea una nueva cuenta. Se inicializa con la fecha del sistema y no se porá modificar. Permanecera invariante. Se puede dar el caso de que una cuenta se desactive (dar de baja) y despues se vuelva a activar (Activar); en este caso se conservará la fecha en la que se dió de alta por primera vez.

Son 8 posiciones alfanuméricas con formato aaaammdd. (año(4)mes(2)dia(2)).

- Activa. Este campo indica si en la cuenta se pueden hacer operaciones con ella, o por contrario permanece invariante hasta que se active.

Este campo contendrá los valores 'SI' o 'NO'.

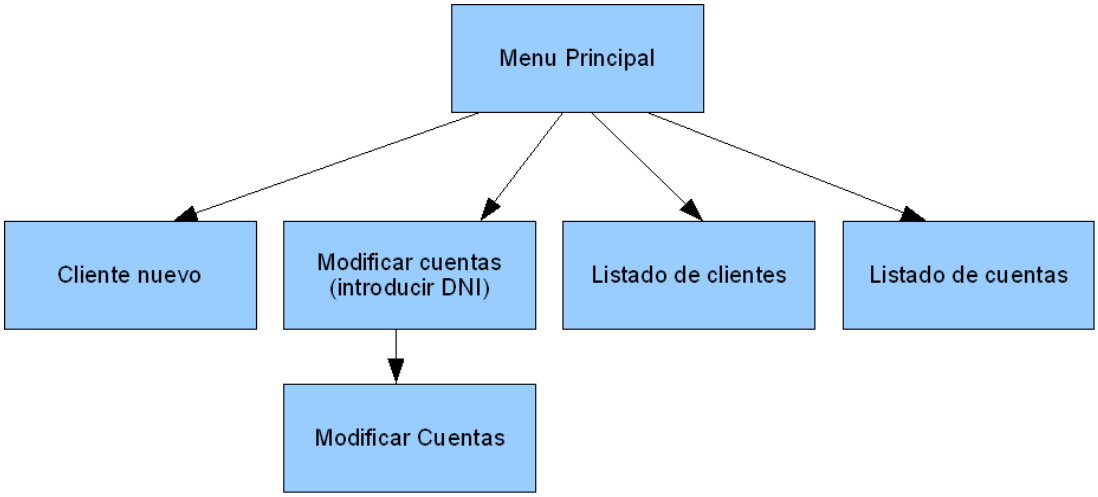
- Fecha de baja. Aquí se indica en que fecha esta cuenta se ha dado de baja. En el momento en el que se desactive la cuenta, contendra la fecha en la que se dio de baja en el sistema. En wel caso de que la cuenta se vuelva a activar, este campo se quedará vacio.

Son 8 posiciones alfanumericas con formato aaaammdd. (año(4)mes(2)dia(2)).

- Saldo. Contine la disponibilidad de dinero de la cuenta de un client. Su formato es de 18 numeros, mas 2 decimales.
- Observaciones. Puede almacenar hasta 250 caracteres de texto libre que un usuario empleado pueda describir para informacion a otros empleados.

8. Estructura de las pantallas de la aplicación.

La estructura de las pantallas es la siguiente:


Menu Principal
Cliente nuevo
Modificar cuentas (introducir DNI)
Listado de clientes
Listado de cuentas
Modificar Cuentas

Arquitectura de acceso seguro a red basada en servidor VPN

Fernando Casas Grande y Mª Dolores García Ureña Ruiz de Salazar González

Unidad Provincial de Informática Barcelona INSS
Gerencia de Informática de la Seguridad Social,

Resumen. En este trabajo se presenta el diseño de un esquema de seguridad para una empresa. El esquema cumple las siguientes características: Disponer de una red interna de ordenadores segura, flexible y optimizada. Permitir el acceso al servidor WebMail que está en una red intermedia (DMZ) y a los datos almacenados en la red interna a los trabajadores externos a través de un túnel en Internet utilizando la tecnología VPN. Esta comunicación se realizará de forma segura. Permitir el acceso a los a los servicios web del servidor WebMail que ofrece la organización a los clientes y a los proveedores.

1. Introducción

El objetivo de una red local virtual es la segmentación lógica de las redes. Así es posible controlar, o incluso impedir todo diálogo entre equipos interconectados sobre un mismo switch.

Una VLAN (Red de Área Local Virtual) es una red conmutada que está segmentada de forma lógica en lugar de hacerlo de forma física o geográfica. Efectivamente, la comunicación entre los diferentes equipos en una red de área local está regida por la arquitectura física. Gracias a las redes virtuales (VLAN), es posible liberarse de las limitaciones de la arquitectura física (limitaciones geográficas, limitaciones de dirección, etc.), ya que se define una segmentación lógica basada en el agrupamiento de equipos según determinados criterios (direcciones MAC, números de puertos, protocolo, etc.).

La VLAN permite definir una nueva red por encima de la red física y, por lo tanto, ofrece las siguientes ventajas:

- Mayor flexibilidad en la administración y en los cambios de la red, ya que la arquitectura puede cambiarse usando los parámetros de los conmutadores.
- Aumento de la seguridad, ya que la información se encapsula en un nivel adicional y posiblemente se analiza.
- Disminución en la transmisión de tráfico en la red.

2. Descripción

Con este proyecto se pretende dar seguridad a una empresa: Una gestoría con trabajadores internos, trabajadores externos, clientes y proveedores. Los objetivos que debe cumplir el sistema de seguridad para proteger a la empresa se concretan en:

1. Configurar una red interna de ordenadores segura, flexible y optimizada.
2. Permitir el acceso al servidor WebMail que está en una red intermedia (DMZ) y a los datos almacenados en la red interna a los trabajadores externos a través de un túnel en internet utilizando la tecnología VPN. Esta comunicación se realizará de forma segura.
3. Permitir el acceso a los a los servicios web del servidor WebMail que ofrece la organización a los clientes y a los proveedores.

3. Solución propuesta

3.1 Arquitectura de la empresa: Descripción

Nuestra empresa es una gestoría.
Tiene tres departamentos y un área para servidores internos.
Los trabajadores están distribuidos en los tres departamentos de la empresa.
También hay trabajadores móviles, se conectan a la empresa mediante un túnel VPN de acceso remoto, posible por disponer de un firewall ASA 5510.
Hay configurada una DMZ con un servidor webmail.
Los clientes de esta gestoría pueden acceder al servicio web de este servidor.
Los usuarios pueden ser de distintos perfiles:

- Usuarios internos:
  - Usuario interno sin acceso al exterior: Tiene acceso al área de servidores y al servidor webmail de la DMZ
  - Usuario interno con acceso al exterior.
- Usuarios externos:
  - Trabajadores móviles:

    Acceden al correo del servidor webmail de la DMZ
    Acceden al servidor de negocio y base de datos de la LAN a través de una VPN
  - Clientes, Proveedores:

    Acceden al servidor web de la DMZ.

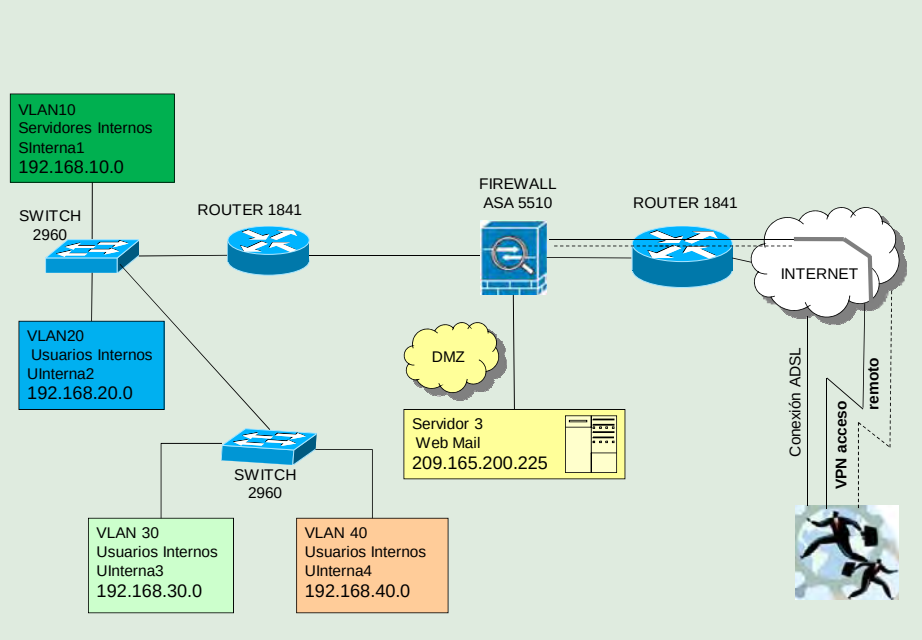


Figura 1 - Arquitectura de acceso seguro basado en servidor VPN

3.2 VLANs de la empresa

La red de la empresa está distribuida en cuatro VLANs.

Las cuatro VLANs, una de servidores y las tres restantes para usuarios internos son:

- VLAN10, servidores internos (VLAN SInterna1).
  - ✓ Dos equipos.
- VLAN20, usuarios internos (VLAN UInterna2).
  - ✓ Diez equipos.
- VLAN30, usuarios internos (VLAN UInterna3).
  - ✓ Doce equipos.
- VLAN40, usuarios internos (VLAN UInterna4).
  - ✓ Ocho equipos.

3.3 Arquitectura de la red local de la empresa

La empresa dispone de dos switchs por el número de equipos conectados a las VLAN. (El modelo 2960 tiene 24 puertos). El switch1 tiene doce equipos conectados y el switch2 20.

Uno de los puertos de cada switch está configurado como port mirroring.

La empresa dispone de dos routers, un router interno y otro externo. Ambos son del modelo 1841.

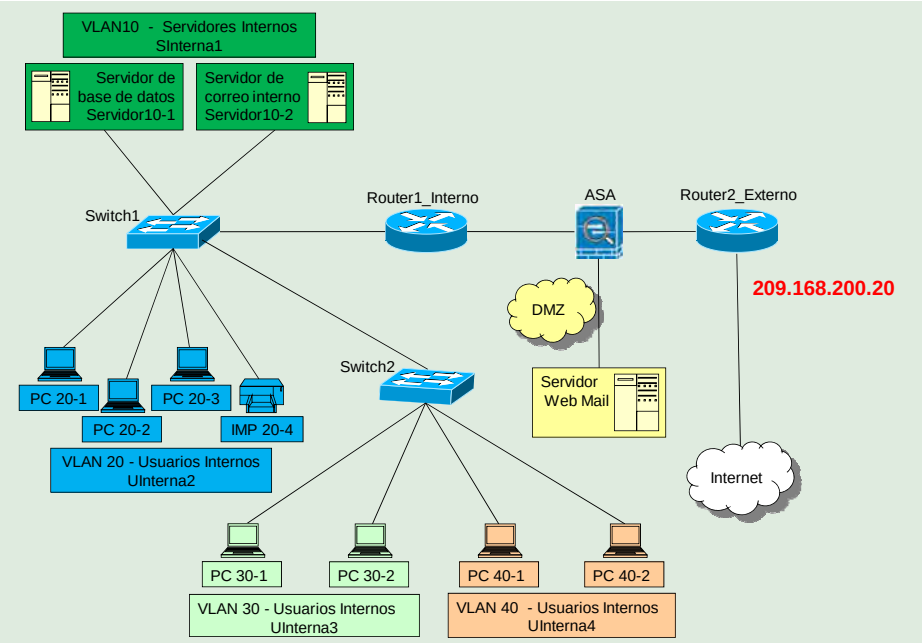


Figura 2. Arquitectura de la red local de la empresa.

3.4 Cortafuegos <<firewall>> y DMZ

Hay un firewall ASA 5510, situado entre los dos routers, para aceptar conexiones IPSec (Seguridad Protocolos de Internet) VPN de los trabajadores móviles de la gestoría y de clientes al servidor web.

Este firewall es el que hace posible el acceso remoto con VPN (red privada virtual).

El objetivo de una DMZ es que las conexiones desde la red interna y la externa a la DMZ estén permitidas, mientras que las conexiones desde la DMZ sólo se permitan a la red externa.

Permite que los equipos de la DMZ puedan dar servicios a la red externa a la vez que protegen la red interna.

Los usuarios internos pueden conectar con la DMZ enrutando a través del firewall, la respuesta del servidor de la DMZ es posible por ser una petición de una máquina interna.

Los usuarios externos conectan con la DMZ configurando PAT en el firewall.

No todos los usuarios internos están autorizados para salir al exterior. Las peticiones HTTP las realizan a través del servidor webmail ubicado en la DMZ.

Los usuarios internos autorizados para conectar con el exterior, lo hacen a través del firewall.

Una de la funciones del servidor webmail es que los usuarios internos le puedan realizar peticiones HTTP y que los usuarios externos (trabajadores móviles) puedan gestionar el correo y la libreta de direcciones a través del navegador web.

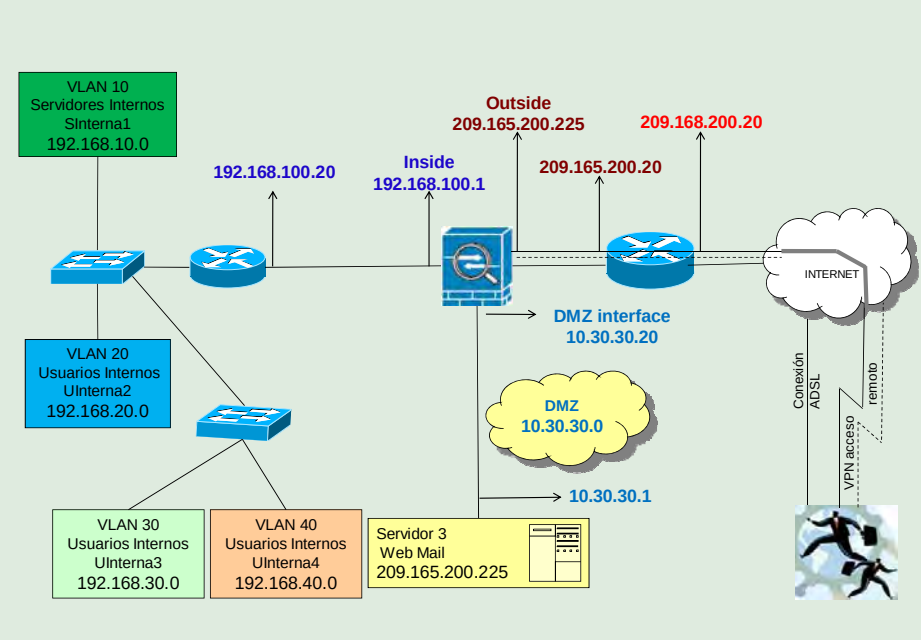


Figura 3. Configuración del acceso remoto con VPN. DMZ.

3.5 NAT y PAT

NAT se utiliza para permitir a los ordenadores de la red interna tener una conexión a Internet utilizando una única dirección IP pública. A través de NAT, los ordenadores son capaces de compartir una única dirección IP pública registrado para acceder a Internet.

También aumentamos la seguridad de la red ya que se oculta la configuración interna de nuestra red privada.

Se aplica NAT (dinámico) en el Router1_Interno y en el firewall, para permitir la conexión de usuarios internos a Internet. Tráfico de salida.

Se aplica PAT (NAT estático) en el Router2_Externo y en el firewall, para permitir exclusivamente el acceso desde el exterior al servidor web (puerto 80 por HTTP) y al servidor de correo (puerto 25 por SMTP). Tráfico de entrada.

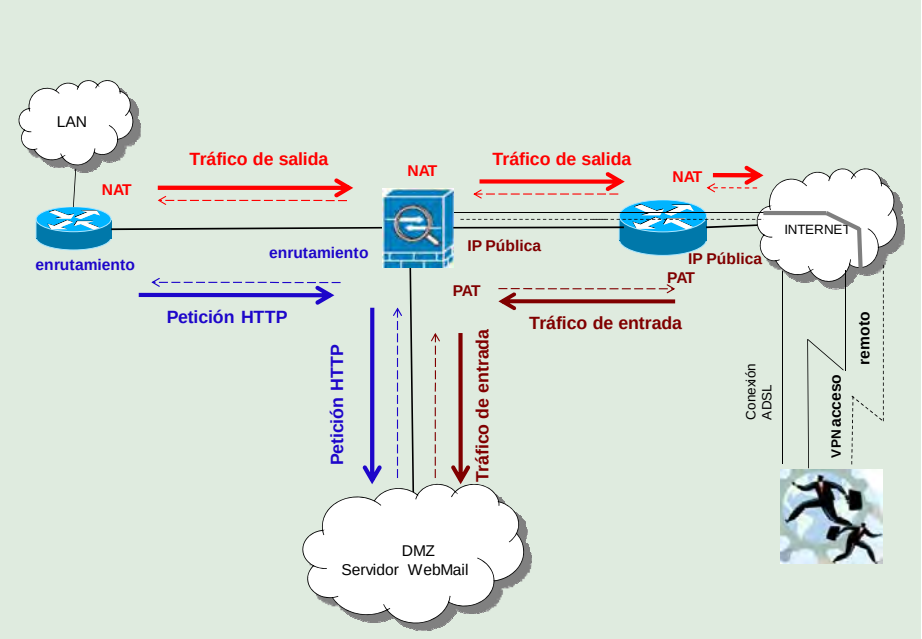


Figura 4. Configuración del acceso remoto con VPN. NAT y PAT.

3.6 Configuración del acceso remoto con VPN

En nuestra empresa hemos implementado un túnel VPN de acceso remoto mediante el firewall CISCO ASA 5510.

Una VPN de acceso remoto permite crear conexiones seguras, o hacer un túnel a través de Internet, dotando de acceso seguro a los usuarios desde sitios remotos

La encriptación y autenticación es a través del conjunto de protocolos IPsec.

Se verifica la identidad de los usuarios, restringiendo el acceso a los no autorizados.

Una vez autentificados, mediante una clave estática compartida, tienen un nivel de acceso similar al que tienen en la red local de la empresa.

Los datos que se van a transmitir a través de la red pública deben ser cifrados para que no puedan ser leídos. Esta tarea la realizamos con algoritmos de cifrado como DES ó 3DES que solo pueden ser leídos por emisor y receptor.

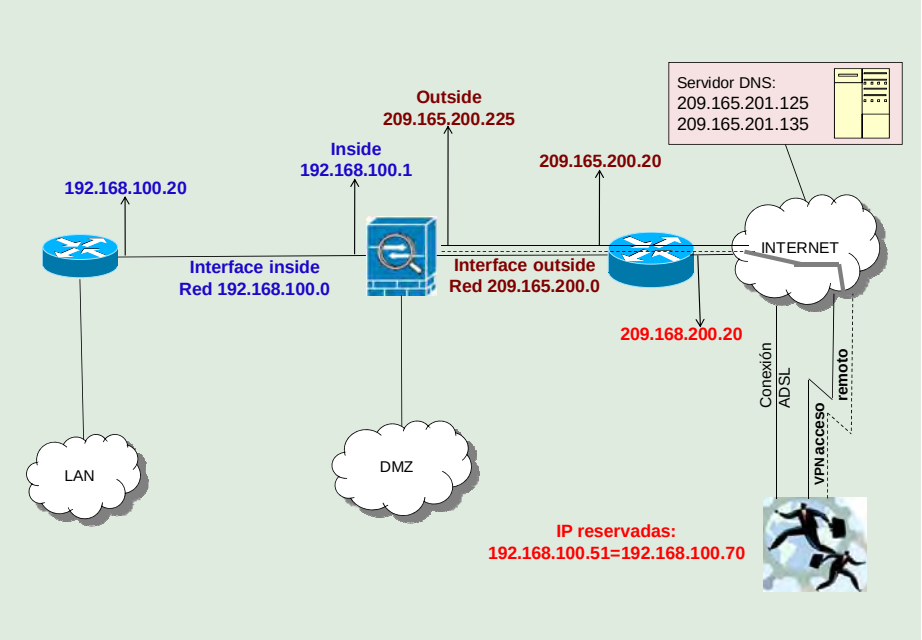


Figura 5. Configuración del acceso remoto con VPN.

4. Conclusiones y trabajo futuro.

La puesta en marcha del proyecto supone una gran mejora en la seguridad de una organización

- En nuestra empresa hemos creado una arquitectura de acceso seguro a red basado en servidor VPN. La red de la empresa está distribuida en cuatro VLAN,s para controlar la escalabilidad, la seguridad y la administración de la red.
- Hay configurada una DMZ (zona desmilitarizada) con un servidor webmail. Es una red local que queda ubicada entre la red interna de la empresa y una red externa, generalmente Internet. Las conexiones desde la red interna y la red externa a la DMZ estarán permitidas, también permite que los equipos de la DMZ puedan dar servicios a la red externa a la vez que protegen la red interna.
- La empresa dispone de trabajadores móviles que se conectan a la misma mediante un túnel VPN de acceso remoto, a través de un firewall ASA 5510.
- La VPN (red privada virtual) va a proporcionar a los usuarios remotos que utilizan una infraestructura pública como Internet la misma conecti-

vidad a la red como si estuvieran sobre la red privada de la empresa. Sin embargo, antes de permitir a un usuario remoto acceder a la red privada es preciso que tomemos ciertas precauciones para asegurar la autenticidad, la integridad de los datos y la encriptación. Así pues, mediante la VPN de nuestra empresa, establecemos una conexión cifrada entre la red privada sobre una red pública como Internet.

- La información que viaja desde la red privada es transportada por una red pública para formar una red virtual y el tráfico es cifrado para que los datos sigan siendo confidenciales.
- La red privada virtual creada en nuestra empresa es de acceso remoto y nos va a permitir conectar con seguridad a los usuarios móviles con la empresa.

5. Referencias.

1. http://es.kioskea.net/contents/internet/vlan.php3. 16 de octubre de 2008.
2. Stallings, William. Comunicaciones y Redes de Computadores. Prentice Hall. 2004
3. Comer, Douglas. Redes Globales de Información con Internet y TCP/ IP. Prentice Hall. 2005
4. José Dordoigne y Philippe Atelin. Redes Informáticas. Editions ENI. Noviembre 2006
5. Juan Desongles Corrales. Materias Informáticas. Editorial Mad. Mayo 2006
6. http://es.wikipedia.org/wiki/Zona_desmilitarizada. 28 Mayo 2009
7. John Wait. Fundamentos de Seguridad de Redes. Especialista en Firewall Cisco. Pearson Educación. 2005
8. http://es.wikipedia.org/wiki/Red_privada_virtual. 26 Junio 2009
9. http://www.osmosislatina.com/soporte/dns.htm. 20 Octubre 2005
10. http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/dns.html
11. http://es.tech-faq.com/understanding-
12. nat.shtml&prev=hp&rurl=translate.google.com

Aplicación informática para la gestión de los usuarios de un Sistema de Envío de Documentos

Alberto Corral Garrido

Tesorería General de la Seguridad Social
Dirección Provincial de la T.G.S.S. de Barcelona
acgtnet1@gmail.com

Resumen. El Sistema de remisión electrónica de datos es un servicio que ofrece la Empresa a otras empresas, agrupaciones de empresas, profesionales y terceros, cuya misión es permitir el intercambio de información entre ambas entidades (la Empresa y sus usuarios) a través de INTERNET en los ámbitos habituales de desarrollo. La aplicación que se presenta en este artículo está pensada para gestionar en todos sus aspectos la actividad de los usuarios del Sistema de remisión electrónica de datos en una provincia, desde su solicitud de alta inicial hasta la finalización de su actividad e, incluso, su posible reactivación. Controla todas las actividades administrativas que realizan y ayuda a los trabajadores responsables de la gestión a cumplir los plazos que marca la ley así como mantener los objetivos de calidad en el ejercicio de sus tareas.

1 Introducción

La principal razón para el desarrollo de esta aplicación es la necesidad de actualizar una aplicación desarrollada a finales del año 2003 utilizando el entorno de programación "Lotus Smart Suite" y basada en un grupo de seis bases de datos en formato "dbf", de la que no se dispone apenas de documentación. Durante el tiempo transcurrido desde su puesta en marcha hasta ahora se han añadido a las dos bases de datos principales más de 9000 y 12000 registros respectivamente, provocando que la aplicación funcione muy lentamente. Además, mientras algunos datos que entonces parecían necesarios no han sido utilizados, otras necesidades nuevas que el sistema no contemplaba se han añadido.

Dos son las peticiones generales requeridas para la nueva aplicación:

- Alcance de la información a tratar. En esta primera petición se requiere al diseñador de la nueva aplicación que obligatoriamente herede toda la información de la anterior aplicación sin perder ningún dato (no se puede partir de cero), manteniendo todas las relaciones e informes: contiene todos los usuarios del Sistema de remisión electrónica de datos de la provincia de Barcelona desde su puesta en marcha.
- Necesidades a cubrir. Esta segunda petición se resume en cuatro tipos de necesidades:

- La anterior aplicación contemplaba como índice de las bases de datos el contenido del campo “DNI” (Documento Nacional de Identidad o Número de Identificación de Extranjero). A medida que el Sistema de remisión electrónica de datos fue creciendo se comprobó que podía existir un mismo “DNI” para varias autorizaciones y que el sistema no lo admitía (estaba diseñado como clave única sin repetición), por lo que los usuarios no tuvieron más remedio que modificar dicho campo añadiendo un ordinal detrás de la letra (DNI: 12345T1, 12345T2, etc.). Esta situación debe resolverse adecuadamente en la nueva aplicación y sin pérdida de datos.
- La aplicación contemplaba un solo tipo de usuario del Sistema de remisión electrónica de datos y en la actualidad existen dos. Se debe habilitar un sistema que permita gestionarlos y, con el menor coste posible de programación, añadir nuevos sistemas que en un futuro puedan aparecer.
- Se han añadido nuevas necesidades de control más detallado de la “vida de un usuario” del sistema, como gestiones de servicio nuevas, informes nuevos, cambios de número de autorización o de sistema, y gestión de tareas pendientes de cada uno de los trabajadors usuarios del sistema.
- Se deberá transformar la información a un entorno de trabajo basado en Microsoft Office 2003, que es el que actualmente se utiliza en la Empresa de Barcelona.

2 Migración de los datos y estructura de la información.

El diseño de la información de la nueva base de datos es el principal reto de la aplicación; ya que, aunque mantiene todos los datos antiguos, la gestión es diferente tanto a nivel de campos clave como de la gestión de la misma.

Para realizar la tarea de migración se ha diseñado un grupo de tres programas de servicio que permiten, a la vez que se hace el traspaso, revisar la información con los trabajadores responsables de la gestión para ver posibles incidencias (errores o inconsistencias en la información) y controlar información que en la nueva aplicación tal vez no debiera aparecer (por desuso o incorrección).

Según consta en el “Manual de instalación” de la documentación general del proyecto, el proceso consta de varios pasos (automáticos algunos y otros que se deben ejecutar de forma manual) cuyo objetivo final es rellenar la nueva base de datos con los datos de la vieja y dejarla lista para ser usada con la nueva aplicación. Sólo cuando la migración está hecha es posible empezar a trabajar con la nueva entrada de información.

La estructura de toda esta información es la siguiente (fig. 1):

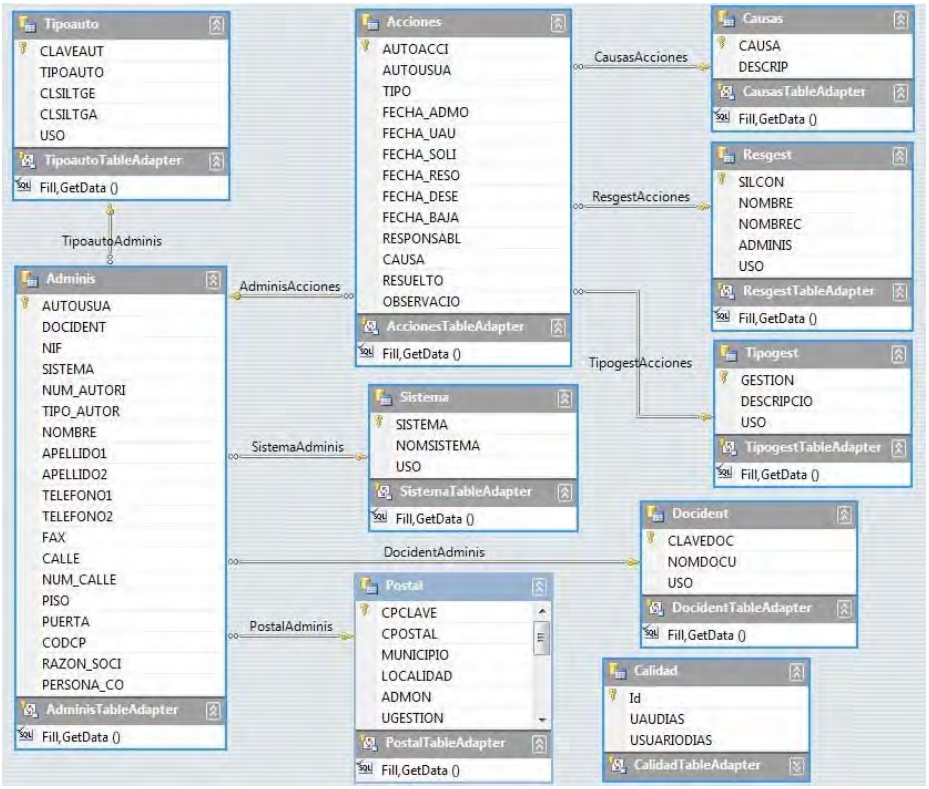


Fig. 1. Diagrama de las diferentes entidades y relaciones que intervienen en la nueva aplicación y que heredan todos los datos de la antigua.

- La tabla “adminis” contiene todos y cada uno de los usuarios del sistema, ya sea en calidad de “solicitantes”, ya sea en calidad de “autorizados”.
  Su clave índice es un campo autonumérico llamado “autousua”.

- La tabla “acciones” contiene todas y cada una de las gestiones de servicio que se realizan sobre dichos usuarios, pasando de un grupo cerrado de cuatro opciones a uno nuevo de quince ampliable a más cuando sea necesario. Su clave índice es un campo autonumérico llamado “autoacci” y se relaciona con una clave externa a “adminis” a través de campo “autousua”.

- La tabla “resgest” contiene a todos y cada uno de los responsables de gestión que pueden trabajar con la aplicación. Controla su perfil de administrador, así como su acceso o no a la aplicación y se basa en la clave segura de acceso general a nuestros sistemas.

- La tabla “postal” contiene todos y cada uno de los distritos postales de la provincia de Barcelona, con datos identificativos de la delegación a la que pertenecen y otros datos diversos.

- La tabla “docident” contiene la información sobre los documentos de identidad.

- La tabla “sistema” contiene la información sobre los diferentes tipos de sistema que pueden existir y que actualmente son dos:
  - Sistema General: para todos los tipos de usuario.
  - Sistema Directo: para usuarios de hasta 15 trabajadores.

  También contiene registros de control referidos a estos dos sistemas.

- La tabla “tipoauto” controla los diferentes perfiles del usuario como forma de clasificación: profesionales colegiados, empresas, terceros, etc, según la clasificación oficial del sistema de la Empresa.

- La tabla “tipogest” controla los diferentes tipos de gestión –acciones individuales- que se pueden realizar en la vida administrativa de un usuario: alta, baja, peticiones varias, etc.

- La tabla “causas” recoge todas aquellas causas objetivas (de oficio, a petición propia, falta documentación, etc.) que antes se anotaban en campos memo. La desaparición de todos los campos memo en la nueva aplicación ha redundado en una mejor velocidad de ejecución.

- La tabla “calidad” recoge los días de que disponen tanto el servicio de atención al usuario como los propios usuarios para responder debidamente a los requerimientos de información y respuesta.

3 Entorno de programación.

El entorno de programación elegido es la plataforma .NET de Microsoft a través del lenguaje de programación C# , concretado a través de dos entornos de programación: Microsoft Visual Studio 2005 y el IDE open source SharpDevelop.

4 Funcionamiento y modos de trabajo

La nueva aplicación se ha diseñado pensando exclusivamente en la forma de trabajar de los operarios, siguiendo el orden natural de las tareas que realizan. El grupo de trabajo está formado por las siguientes personas:

- Dos personas se encargan de los trámites de alta de los solicitantes, comprobando que reúnen los requisitos adecuados. Pueden solicitar información

adicional al usuario y, finalmente, aceptar o denegar su solicitud, solicitar información adicional al usuario y finalmente, aceptar o denegar su petición. También es posible que durante este tiempo, el solicitante desista por diferentes motivos, encargándose de ello también. Todo este proceso queda debidamente recogido en la aplicación a través de accesos directos desde el menú principal.

- Una persona simultanea tareas con el primer grupo, junto con tareas propias relacionadas con usuarios autorizados que solicitan cambios diversos en la autorización. Igualmente, puede pedir información y, finalmente, aceptar o denegar la petición.
- Una persona se encarga básicamente de gestionar las bajas, ya sea de oficio, ya sea a petición propia, así como las peticiones de usuarios secundarios para las diferentes autorizaciones activas.
- Dos jefaturas que gestionan el servicio y que básicamente consultan datos, aunque eventualmente también pueden introducir información, sobre todo en el segundo caso cuando se dan bajas de oficio por inactividad del autorizado. También solicitan con cierta regularidad informes de control de servicio sobre el estado de los usuarios, ya sean solicitantes, denegados y/o desistidos, autorizados y en situación de baja.

En relación con este último punto, la jefatura del departamento decidió incluir expresamente en los requisitos del programa la posibilidad de controlar en todo momento el estado real de los autorizados a los efectos del control de calidad. Es por ello que en el menú de entrada de la aplicación ya aparece un resumen general con los datos más relevantes que permite acceder a un informe detallado de cada uno de los usuarios y su estado real. Esta información se relaciona directamente con los plazos administrativos de contestación a los usuarios (objetivos de calidad) y de respuesta de los usuarios (aprobación, denegación o desistimiento de sus peticiones), tal y como se muestra en la figura 2.

Como era bastante probable que en el trabajo diario varios responsables de gestión accedieran a los mismos registros y, aunque con poca probabilidad, también era posible que intentasen acceder al mismo usuario para editar su información, la aplicación controla adecuadamente los posibles problemas de concurrencia.

Normalmente esta aplicación estará en funcionamiento durante toda la jornada laboral, ya que su acceso es habitual a lo largo de la misma. Pensando en la forma natural de trabajo de los operadores responsables de su gestión, la aplicación presenta una barra de menú que, seguida de izquierda a derecha, recoge todos los procesos que la vida administrativa de un usuario puede seguir (fig. 3).

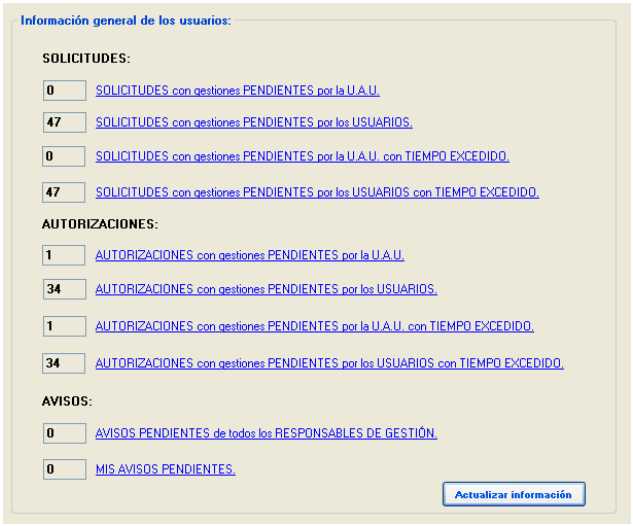


Fig. 2. Menú principal de la aplicación donde se puede ver el resumen general de estado de los usuarios.

Fig. 3. Menú principal del programa con detalle de los accesos directos a las tareas habituales de los trabajadores responsables de la gestión.

Los accesos directos son los siguientes:

- Nuevo solicitante. Es la entrada como usuario en el Sistema de remisión electrónica de datos de la Empresa. La aplicación permite resolver en ese

mismo momento la petición de alta o bien dejarla pendiente en el caso de faltar información (fig. 4).

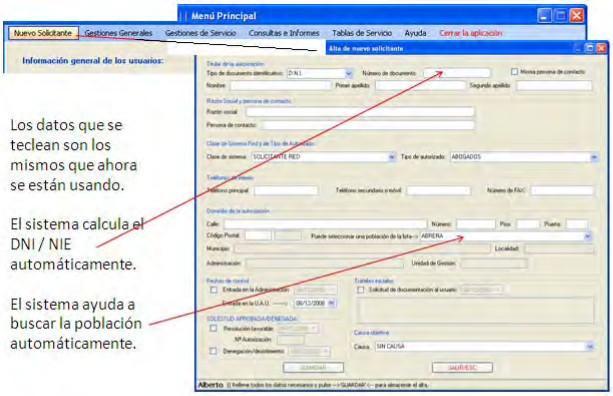


Fig. 4. Menú para el alta de nuevos usuarios.

- Gestiones Generales de servicio. Son aquellas cuyo alcance afecta a la totalidad de la existencia del usuario dentro del sistema.
- Gestiones de servicio. Las más habituales en la vida de un usuario (fig. 5 y 6)

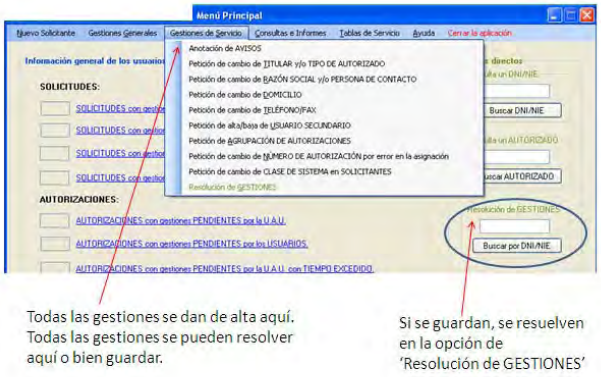


Fig 5. Menú de Gestiones de servicio.

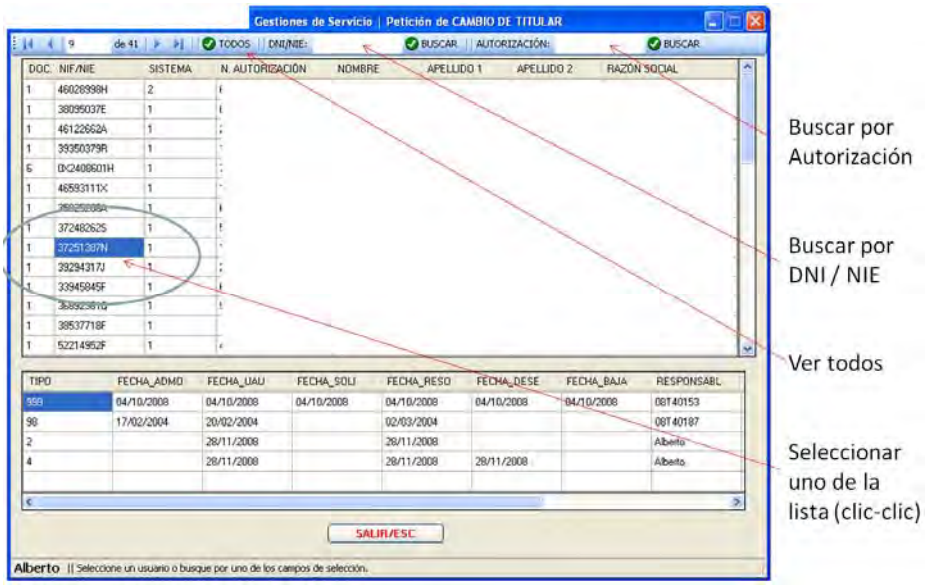


Fig 6. Ejemplo de una gestión de servicio habitual: Petición de cambio de titular de una autorización.

- Consultas e informes. Incluye una lista de consultas estándar según usuario, solicitante, denegado/desistido, autorizado y baja, donde se puede elegir también por tipo de sistema de remisión electrónica de datos. Además y también requisito expreso de la jefatura, un generador manual de consultas que permite elegir la información a buscar a partir de la selección de los campos de datos más relevantes de la base de datos.

- Tablas de servicio. Reservado sólo a aquellos responsables de gestión que tengan permiso de administrador en la aplicación, permite gestionar las ocho tablas de servicio de las que se sirve la aplicación para gestionar adecuadamente toda la información.

5 Conclusiones

La nueva aplicación se pone en marcha a primeros de este año 2009, después de un proceso previo de depuración de datos de la antigua aplicación. Esta depuración es necesaria para una correcta importación de los datos, así como para dejar la información en correcto estado.

Desarrollo de una Herramienta para la Migración de Bases de Datos de Microsoft Access a Oracle.

Antonio Echevarrías Escuder

Unidad Provincial de Informática TGSS-ISM de Valencia
Gerencia de Informática de la Seguridad Social,

Resumen. Elaboración de un sistema autónomo de migración de bases de datos desarrolladas bajo Microsoft Access para su conversión a Oracle 9i. Los principios básicos de funcionamiento de la misma se establecieron a partir de la simplicidad, completa parametrización, rapidez e independencia del entorno de trabajo Oracle. No se consideran en su desarrollo mas que aspectos de datos, dejando de lado los relacionados con todo aquello que suponga código de programación. Cumple con el propósito indicado además de servir como un banco de pruebas para comprender y analizar las estructuras lógicas de almacenamiento de Oracle.

1. Introducción

Este trabajo intenta realizar el desarrollo de una herramienta informática que permita agilizar y resolver el problema de la migración de datos entre entornos de producción diferentes. En concreto, bases de datos generadas en Microsoft Access que deben pasar a ser gestionadas bajo Oracle 9i.

Aunque existen en el mercado algunas soluciones dedicadas a este fin, se ha decidido su realización por varios motivos.

En primer lugar, consideramos se trata de un ejercicio didáctico que permite la mejor compresión de las estructuras de almacenamiento utilizadas por los dos Sistemas Gestores de Bases de Datos.

En segundo lugar, la realización de una herramienta propia permite un mayor control de la misma en todas sus fases, amén de no necesitar licencias ni permisos, aunque es cierto que algunas de las herramientas existentes son libres o proporcionadas por el propio Oracle.

Por último, del análisis realizado sobre algunas de las herramientas existentes, cabe destacar que no todas ellas aportan soluciones de sencillez de manejo ni de estabilidad, como se observó en la realización de algunas pruebas, siendo estos factores clave en los presupuestos de los que partimos. Entre estos factores de estabilidad, cabe destacar la nula o mala gestión de tablas vinculadas de Access en estos productos.

Por otra parte, todas ellas parten de una base de datos ya existente en Oracle, mientras que en nuestro caso se pretende que la propia creación sea realizada por la aplicación.

2. Descripción

La aplicación objeto de este proyecto debía cumplir los siguientes requisitos:

- Ser capaz de crear la base de datos Oracle mediante las informaciones que se le suministren a través de las diversas pantallas de la aplicación.
- Ser sencilla de manejo y parametrizable, permitiendo libertad a la hora de ubicar los diferentes elementos que se generarán. Las conversiones de tipos de datos y demás deberán estar reflejadas en sus correspondientes tablas que deberán poder ser posteriormente modificadas.
- Estar disponible para ofertar la salida en sistemas Windows y/o Linux.
- Permitir la selección de cualquier base de datos Microsoft Access.
- Analizar las estructuras de información de la base de datos seleccionada, realizando la descripción completa de las mismas a partir de las colecciones de objetos existentes en Microsoft Access.
- Realizar la adaptación de las estructuras analizadas a su forma adecuada en Oracle, atendiendo a reglas de nomenclatura, tipología de datos, etcétera.
- Generar los mecanismos necesarios para poder realizar la creación de la base de datos Oracle (ficheros ejecutables de lanzamiento, ficheros de script de Oracle, etcétera) homóloga a la base de datos de Microsoft Access seleccionada inicialmente.
- Realizar la exportación de los datos contenidos en la base de datos seleccionada de acuerdo al formato tipo que se determine. Esta exportación de datos será optativa y, en su caso, estará acompañada de la creación de los elementos Oracle necesarios para su posterior carga en la base de datos objetivo (ficheros de script, tablas especiales, etcétera).
- La automatización buscada será aplicada solo a la fase de análisis y generación de resultados, permitiendo que la correspondiente a la creación y carga de datos en la base destino de Oracle sea supervisada por el administrador del sistema antes y durante su ejecución.
- Se asumirán como válidos una serie de parámetros iniciales a la hora de la generación de los ficheros necesarios para la correcta ejecución de Oracle (fichero “init.ora” entre otros).
- El sistema deberá permitir la separación de datos e índices en sus correspondientes espacios de tablas de Oracle, rompiendo así la unicidad de almacenamiento que mantiene Microsoft Access de forma obligatoria.
- No se realizará migración alguna de aquellos aspectos que estén relacionados con programación, esto es, código utilizado para formularios y/o controles.

3. Conceptos básicos previos.

3.1. Estructuras de la información en Microsoft Access y Oracle.

Para el desarrollo de la aplicación es necesario poder analizar de forma concreta la forma en que Microsoft Access organiza la información para su gestión.
Así, en las siguientes figuras se observa la manera en que jerarquiza la misma, forma que nos permite, mediante el correspondiente código VBA, acceder a cada uno de los objetos y propiedades que lo conforman.

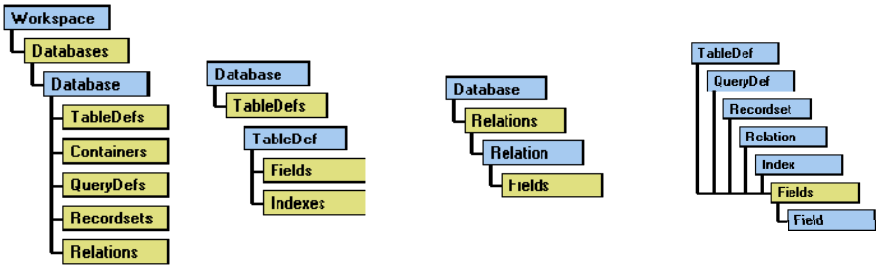


Figura 1 Organización de la información en Microsoft Access.

De manera análoga a la anterior, es preciso para el tratamiento de la aplicación comprender la forma en que se estructura la información en Oracle.

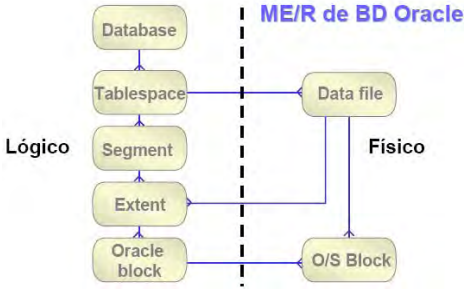


Figura 2 Organización de la información en Oracle.

Atendiendo a estas dos formas diferentes, el sistema deber realizar la traducción de las estructuras existentes en la base de datos de Microsoft Access a su forma homóloga en Oracle, creando para ello los tablespaces necesarios tanto para almacenamiento de datos e índices como para la gestión propia de Oracle (tablespaces de Undo, temporales y de log). Es fundamental entender la diferente forma en que ambos sistemas tratan el almacenamiento tanto físico como lógico de la información.
Sirva como ejemplo que, a nivel físico, Microsoft Access gestiona todos los elementos en un único archivo físico, mientras que Oracle lo gestiona realizando la división en diferentes archivos.

3.2. Orígenes de datos externos.

Entendemos bajo este término aquellas agrupaciones de datos contenidas en uno o más ficheros que no son directamente gestionadas por el SGBD con el cual se está trabajando.

Oracle tiene varios mecanismos para soportar orígenes de datos externos, el denominado **SQL*Loader** y la técnica de las **tablas externas**. Su uso más común en ambos casos es el almacén de datos cuando se cargan normalmente grandes cantidades de datos desde un sistema transaccional.
Una tabla externa se define mediante metadatos que describen los tipos de columna Oracle y la correspondencia entre los datos externos y dichas columnas. También es necesario un controlador de acceso para acceder a los datos externos, que es proporcionado por él de forma predeterminada para archivos planos.
En nuestro trabajo, decidimos apostar por la utilización de las tablas externas, permitiéndonos ello crear los scripts necesarios para ejecutar de forma automatizada la carga de los datos.
Como requisitos básicos para su uso se pueden citar la necesaria definición en la instancia de un directorio externo así como la creación de la propia definición de la tabla externa.
Una vez realizados los pasos anteriores, permiten trabajar con ellas mediante la utilización de sentencias SQL, tal y como se realizó para la carga de datos mediante los correspondientes INSERT.

4. Fases de la aplicación.

La aplicación desarrollada sigue las siguientes fases:

1. Obtención y mantenimiento de la información necesaria para su funcionamiento:
   Sistema Operativo sobre el que trabajará Oracle (en esta fase, solo Windows aunque se dejó todo parametrizado para la posterior inclusión de entornos Linux).
   Rutas (ORACLE_BASE y ORACLE_HOME) en las que está instalado.
   Nombre de dominio, si existe.
2. Selección de la base de datos a migrar.
3. Análisis de la estructura de la base de datos a migrar.
   Análisis y anotación de tablas, campos, índices y relaciones de la base de datos seleccionada, incluidas las propiedades necesarias de cada uno de los objetos (valores predeterminados de los campos, …).
4. Adaptación de la estructura analizada a su forma homóloga en Oracle.
   Utilizando los datos recogidos en la anterior para proceder a su reconversión atendiendo a las características de Oracle, con la correspondiente aplicación de las reglas de nomenclatura exigidas por él, realizando un proceso de conversión de nombres en aquellos objetos en los que así resulta necesario.
   Además, se realiza la conversión de tipos de datos utilizando para ello una tabla que permite al usuario especificar la forma en que un tipo de dato concreto de Access se reconvierte en Oracle, al existir tipos diferentes e incluso características diferenciadas entre ellos.
5. Elaboración de las herramientas necesarias para la creación de la base de datos Oracle.

Se generan todos los ficheros de script “sql” y “bat” necesarios para la creación de la base de datos de destino, utilizando para ello la sintaxis propia de Oracle así como una serie de plantillas básicas almacenadas en tablas del sistema, lo cual permite una mayor parametrización y flexibilidad a la aplicación.
Se genera el correspondiente fichero INIT.ORA de manera automática, asumiendo diversos valores para los parámetros en él establecidos. Para ello, se determinó seguir las recomendaciones existentes en el fichero INITSMPL.ORA referidas a bases de datos de tipo “medium”.
No se aplican las correspondientes acciones que se deben realizar en los ficheros LISTENER.ORA y TNSNAMES.ORA ya que nuestro programa no tiene por qué acceder al servidor en el cual está instalado Oracle.
Todos estos parámetros pueden ser corregidos si es preciso por el administrador de la base de datos antes de su ejecución.

6. Elaboración de las herramientas necesarias para la creación de las estructuras propias de la base de datos requeridas en Oracle.
   Creación de los scripts necesarios para la generación de tablas, índices y relaciones en Oracle.
   Se aplica directamente la sintaxis existente en Oracle para cada uno de ellos.

7. Migración de los datos en una forma asumible para su posterior importación en Oracle.
   Se realizará la migración de los datos contenidos en las tablas de la base de datos de Access mediante la exportación de los mismos a ficheros de texto de acuerdo con las siguientes características:
   - Separadores de campo y de registro los símbolos “;” y retorno de línea respectivamente.
   - Campos delimitados por los caracteres comillas dobles.
   - Todos los datos se exportarán como texto, independientemente de su tipo de dato asociado de origen. Se fuerza así a que la traducción al correspondiente tipo asociado en Oracle se realice mediante SQL Loader.
   - Se realizará la importación haciendo uso de las “tablas externas” en Oracle.
   - Los datos migrados se modifican para que no existan entre ellos valores nulos ni contengan caracteres asociados a los saltos de línea.

   - Elaboración de las herramientas necesarias para llevar a cabo la importación de los datos en la base de datos Oracle.
     Ficheros “bat” y script “sql” de creación de las tablas externas, consultas de inserción en las tablas de Oracle creadas y eliminación de las tablas externas creadas.

El esquema básico de ejecución a seguir es el siguiente:

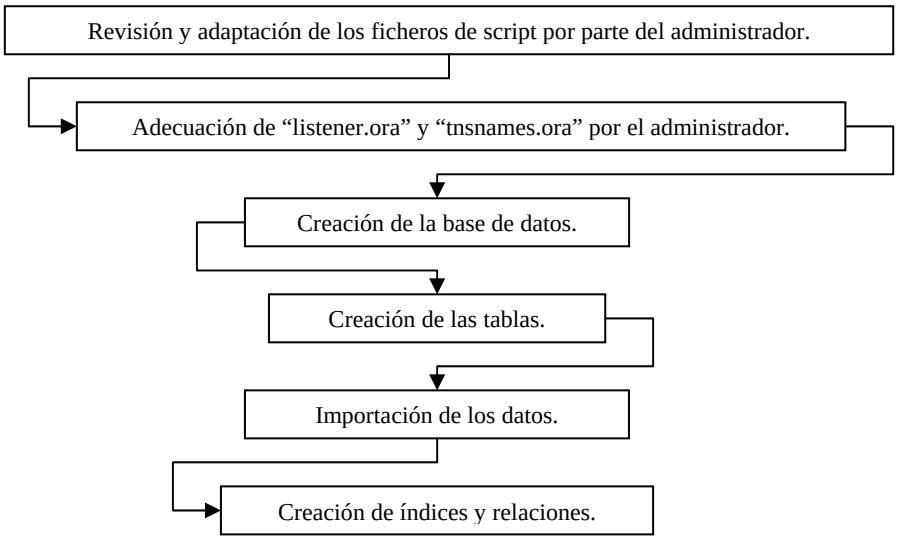


Figura 3 Esquema de ejecución de los productos generados.

5. Funcionamiento de la aplicación.

Siguiendo los criterios establecidos para su desarrollo, la aplicación se ejecuta en determinadas fases como si se tratara de un asistente, dándole así un aspecto de sencillez suficiente.
Esto, unido a la parametrización realizada, la hace tanto sencilla como adaptable.
A continuación se muestran algunas de las pantallas utilizadas.

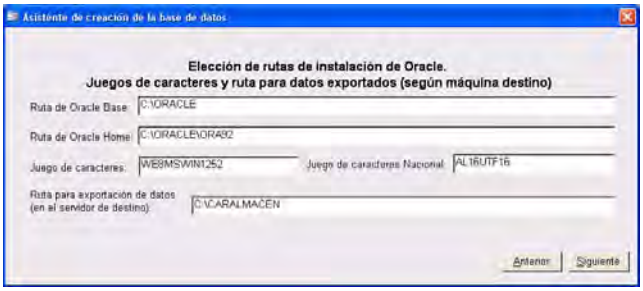


Figura 4 Pantalla de captura de datos básicos.

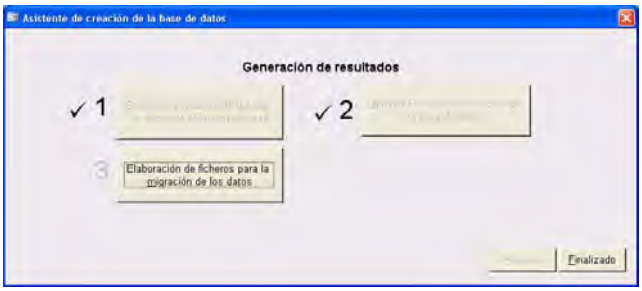


Figura 5 Pantalla de generación de ficheros.

6. Pruebas realizadas.

Para la realización de las pruebas del sistema se crearon dos bases de datos diferentes. La primera, denominada "Almacén", es una base de datos normal de Microsoft Access y a la que se le han generado de forma automática una serie de registros en cantidad suficiente para poder realizar una estimación de tiempos de tratamientos y carga de datos (en concreto, la tabla Solicitudes contenía 500.000 registros, cantidad más que suficiente para una prueba de este tipo).
La otra base de datos, denominada "Ganadero", es, en realidad, dos bases de datos de Microsoft Access que contienen información vinculada entre ellas, comprobando así la ejecución del aplicativo tanto para realizar la exportación de los datos vinculados como los no vinculados.
Tras la realización del proceso, se procedió a efectuar la migración de los ficheros generados al entorno Oracle y su ejecución de acuerdo con el esquema antes visto.
Para justificar el mismo, se realizaron sendas cargas por cada una de las bases de datos de prueba al objeto de establecer definitivamente la mejor forma de realizarla, tal y como se aprecia en la siguiente tabla.

Modo de carga	Índices y relaciones	Carga de datos	Tiempo total
Primero carga de datos y después creación de índices y relaciones	31 segundos	18 segundos	49 segundos
Primero creación de índices y relaciones y después carga de datos	2 segundos	77 segundos	79 segundos

Junto a ello, se realizaron las pruebas correspondientes de integridad relacional, siendo el resultado de ellas satisfactorio.

7. Conclusiones y mejoras.

El desarrollo realizado funcionó de forma correcta realizando la totalidad de aspectos fijados para él, resultando en cuanto a rapidez y sencillez bastante robusto.
Si bien otros productos de carácter comercial incorporan mayor número de funcionalidades, creemos que este desarrollo es totalmente funcional desde la propia creación de la base de datos hasta la exportación completa de los datos.
No obstante, quedan aspectos que podrían ser incorporados en una fase posterior tales como:

- Adecuación del aplicativo para generar bases de datos Oracle sobre plataformas Linux.
- Dar soporte a los campos de Microsoft Access de tipo autonumérico, incorporando trigers para ello.
- Incorporar el tratamiento de las restricciones incorporadas a campos y tablas en Microsoft Access
- Incorporar el tratamiento de las características de las relaciones de Microsoft Access tales como borrado en cascada y actualización en cascada. Se optaría tal vez en este caso por la elaboración de los correspondientes trigers en Oracle.
- Posibilitar la carga de los parámetros solicitados al usuario para la creación de la base de datos de Oracle desde un fichero de texto. De esta forma, podríamos generar plantillas de carga y liberar, si fuese preciso, al usuario de tener que introducir los datos cuando el mismo no es el administrador del sistema.

8. Bibliografía.

Date, C.J.: *Introducción a los Sistemas de Bases de Datos*, Addison-Wesley Iberoamericana, USA
Ditzel, L,: Oracle Server 9i Quick Reference Guide , Germany, 2004
Tablas Externas – Dos ideas, http://www.dosideas.com/wiki/Tablas_Externas
External Tables, http://www.orafaq.com/node/848, OracleFaq's
http://www.mcs.csueastbay.edu/support/oracle/doc/10.2/server.102/b14215/et_concepts.htm#g1017623, California State University
http://www.oracle.com/technology/tech/migration/focusareas/access.html, Oracle
Murray, C.: *SQL Develper Supllementary Information for Microsoft Access Migrations. Release 1.5*. *E12154-01*. ORACLE, April 2008
Silberschatz, A., Korth, H., Sudarshan, S.: *Fundamentos de Bases de datos.4ª Edición* Editorial McGrawHill. Madrid (España). 2002
Oracle SQL Developer,
http://www.oracle.com/technology/products/database/sql_developer/index.html, Oracle, Consultada el 30 de marzo de 2009, 12:50:35

Proyecto de Gestión de Ficheros Adabas

Ivan Eguilleor Villena

Centro de Desarrollo de Aplicaciones, Área de Afiliación
Gerencia de Informática de la Seguridad Social

Resumen:Este proyecto pretende ser una herramienta de gran utilidad para programadores en entornos NATURAL / ADABAS, pues permite de una forma muy visual acceder a la descripción de los ficheros ADABAS y sobre todo, saber en todo momento, como se relacionan estos entre si.

Aprovechando la existencia de un Servidor de Aplicaciones y una página propia de la Gerencia de la que se puede enganchar, el citado proyecto es mas viable tanto técnica como económicamente.

Según esto, la seguridad de las páginas y ficheros relacionados con ellas, correrá a cargo de los cortafuegos anteriores y posteriores al "proxy" de la Gerencia, así como del Servidor de Aplicaciones

1. Introducción

Este es el resultado de la aplicación de los conocimientos adquiridos en las Jornadas Nacionales sobre Aplicación de las Tecnologías de la Información y Comunicaciones Avanzadas (ATICA 2007 - 2009)

2. Objetivos del proyecto.

Uno de los objetivos principales es conseguir una navegabilidad muy amigable y de una forma muy intuitiva.

Los servicios que ha de ofrecer la aplicación WEB, quedan definidos de la siguiente forma:

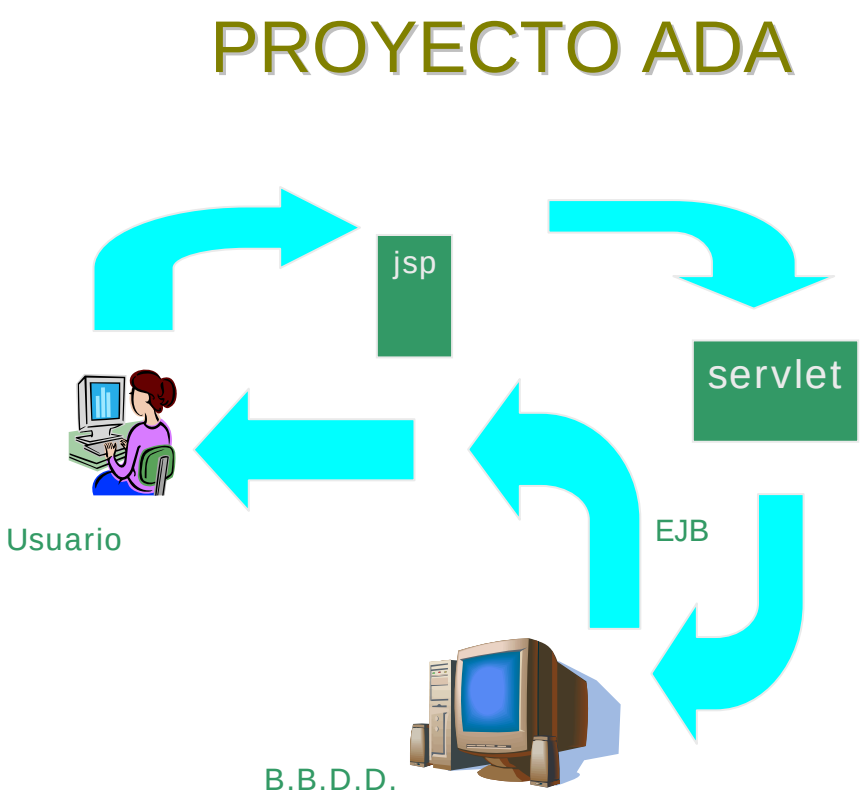
- Administración de ficheros:
  - Creación de ficheros
  - Eliminación de ficheros
  - Inserción de campos en un fichero
  - Eliminación de campos de un fichero
- Consultar ficheros por nombre.
- Listar todos los ficheros disponibles para su posterior consulta tras seleccionar uno de ellos.

3. Diagramas:

A continuación se muestran los diagramas más relevantes de la aplicación. Esto es:

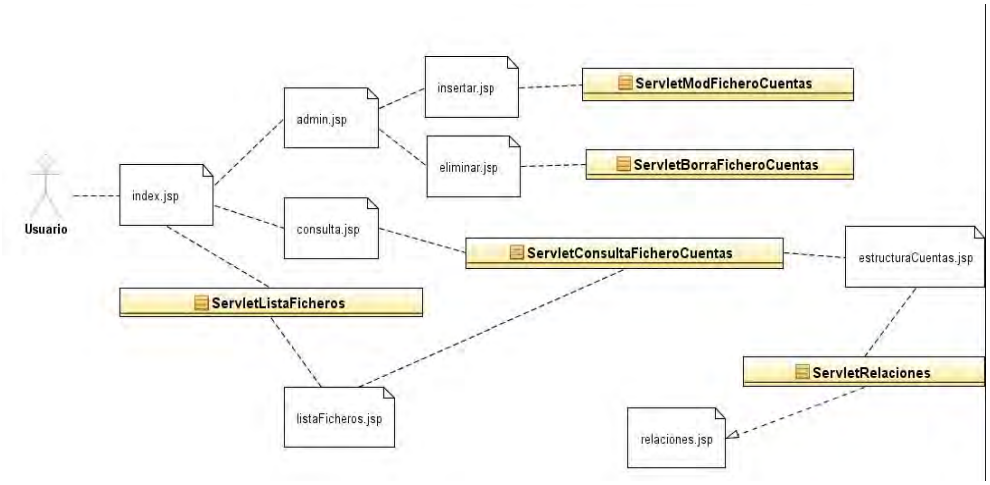
- Flujo
- Navegación
- Casos de Uso
- Clases

3.1 Diagrama general de flujo de información:



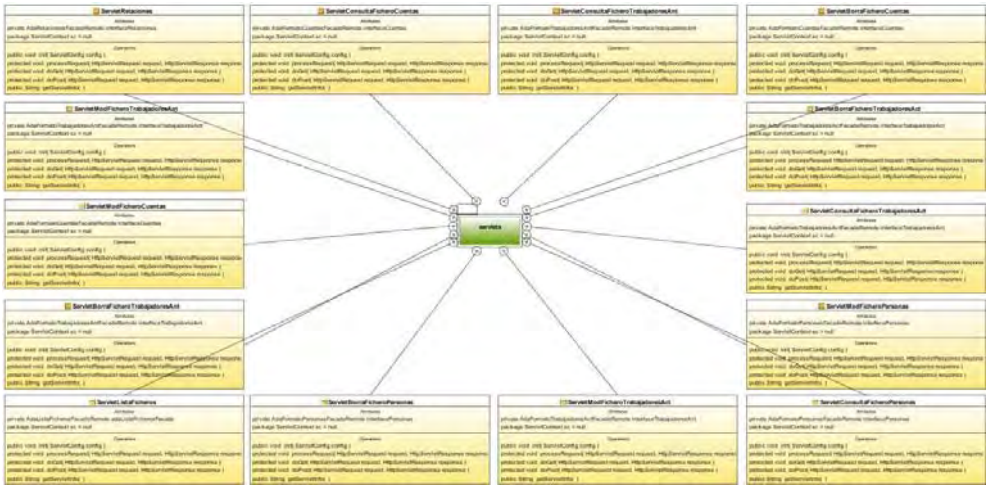
3.2 Diagrama de Navegación

El diagrama de navegación es el que se muestra a continuación de forma esquemática y reducida (pues solo se muestra la secuencia relativa al fichero de cuentas):



3.3 Diagrama de Casos de Uso

El diagrama de casos de uso, como se aprecia, muestra la alta cohesión y bajo acoplamiento de los módulos, factor muy positivo a la hora de desarrollar aplicaciones:

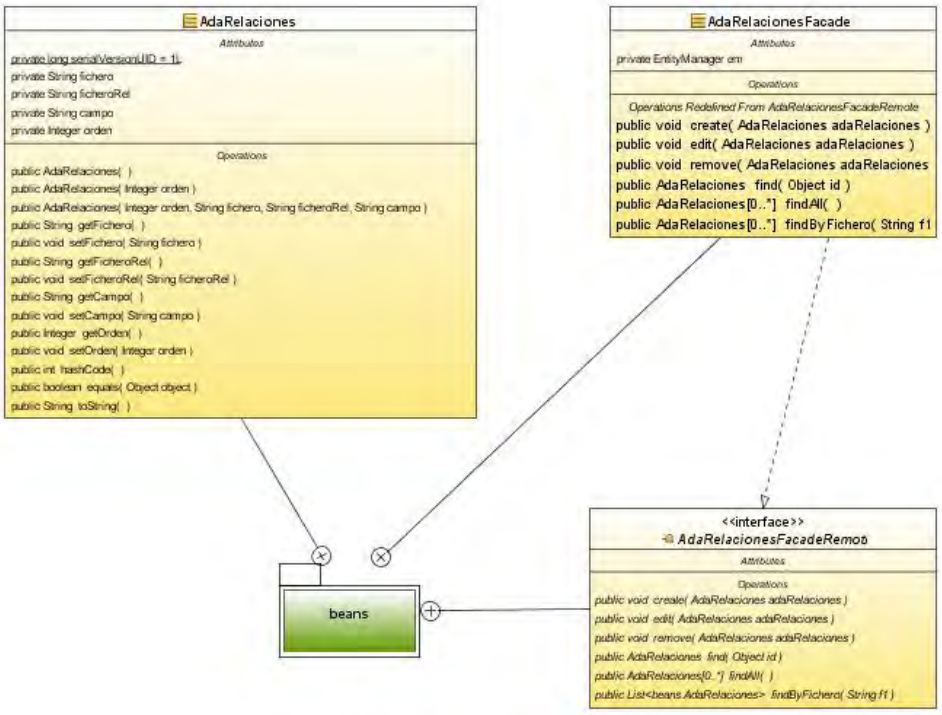


3.4 Diagrama de Clases

A continuación observamos el detalle de las clases relacionadas con uno de los ficheros por mayor claridad. Como se aprecia, por cada fichero aparecen tres clases:

- Clase
- ClaseFacade: Implementa los métodos
- ClaseFacadeRemote. Es la interface

Esto se debe simplemente a la utilización de Patrones de Diseño para acceder a los datos.

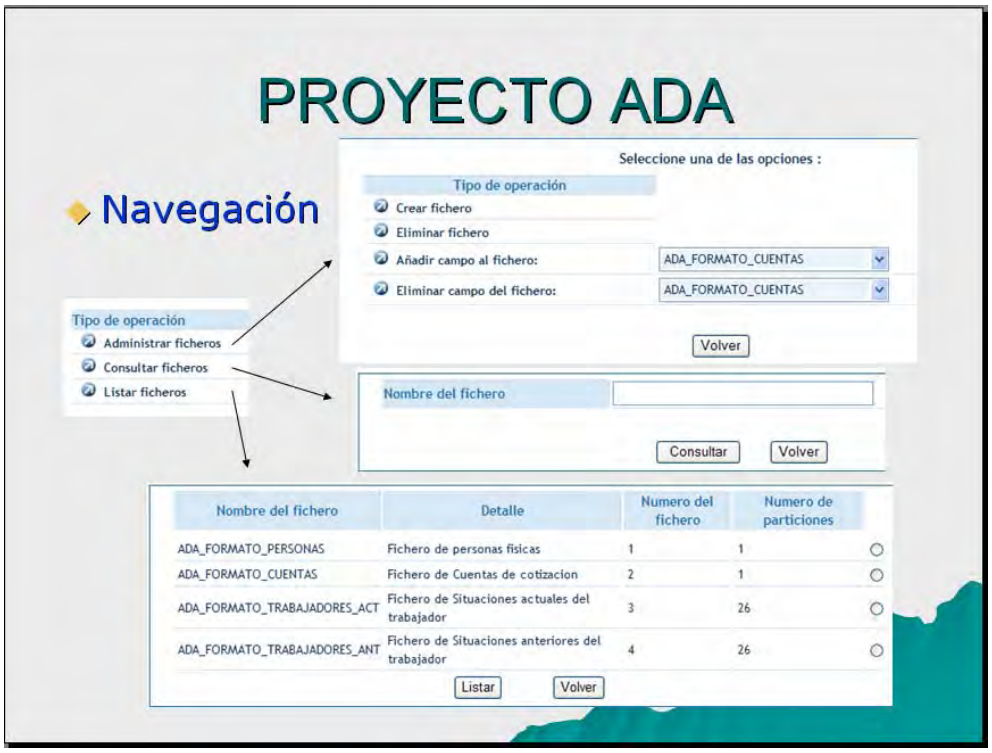


4. Navegación.

En este punto vemos la navegación entre las distintas opciones de la aplicación.

4.1 Página principal.

Desde la página de inicio (index.jsp) se accede a las distintas opciones del menú, según se muestra en el siguiente diagrama:



Las páginas utilizan hojas de estilos (css) para conseguir una presentación homogénea.

4.2 Administración de ficheros.

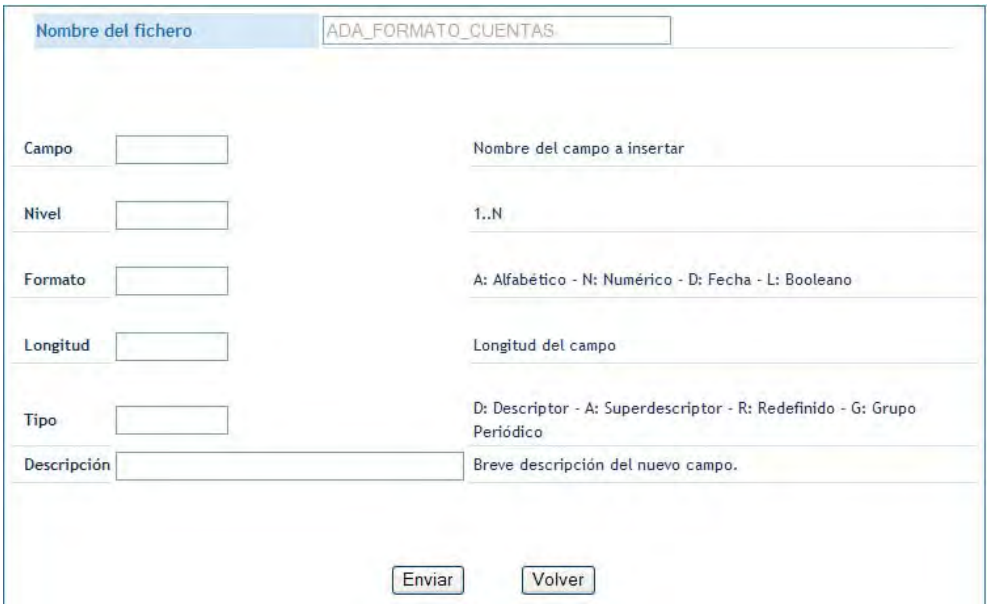
Con esta opción es posible tanto añadir como eliminar campos a determinado fichero.

Las opciones de Crear y Eliminar ficheros no están disponibles por motivos de seguridad

4.2.1 Añadir campo al fichero:

Se despliega una lista con los todos los ficheros del sistema, para seleccionar el deseado.

Una vez seleccionado el fichero, aparece la siguiente pantalla

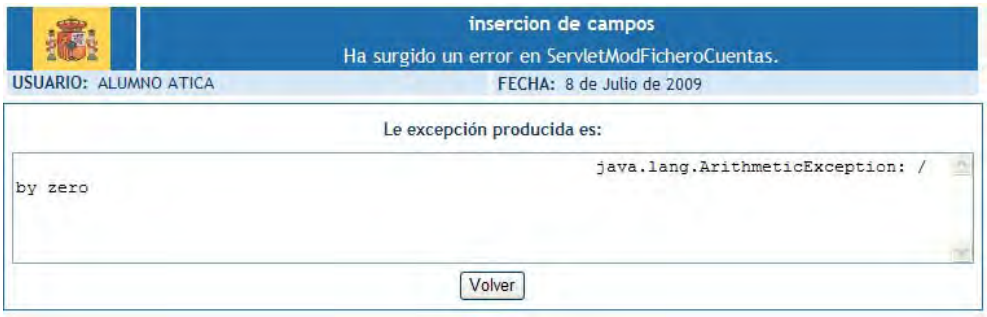


Debemos rellenar todos los datos, que tras pasar el proceso de validación, nos retorna al menú de gestión.

En este caso, sabremos que el proceso se ha realizado correctamente

Como se puede ver, a la derecha de los campos a rellenar se facilita información sobre el formato y descripción de los campos a introducir.

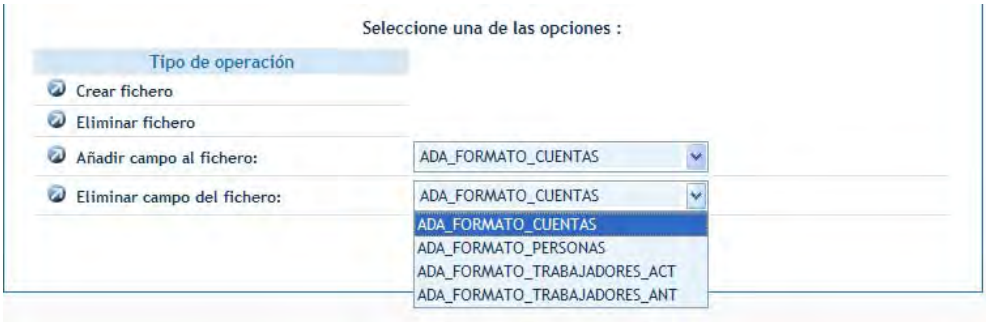
En caso de error se mostrará la pantalla genérica de error:



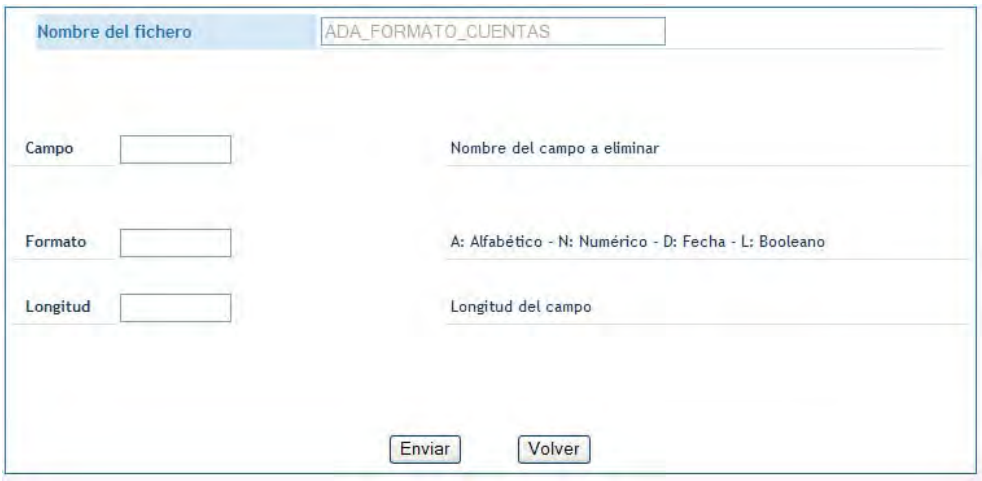
Esta pantalla es de gran ayuda al administrador de la aplicación, pues muestra el módulo que ha fallado y la información relativa al tipo de error.

4.2.2 Eliminar campo del fichero:

Análogamente a la inserción se permite la eliminación de campos



… donde introduciremos los campos necesarios



En este caso, aunque con el nombre del campo sería suficiente, se pide formato y longitud por seguridad.

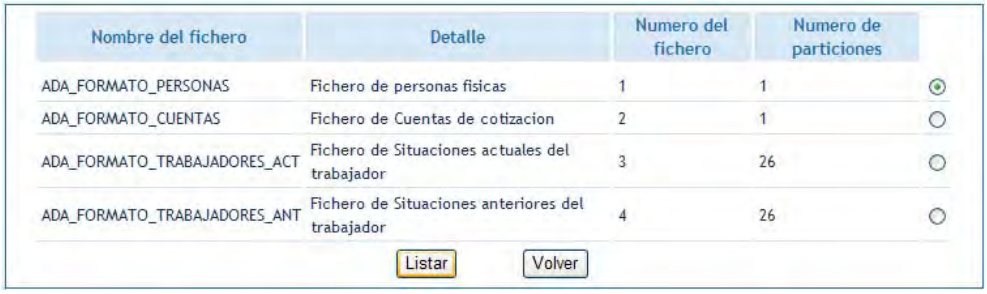
4.3 Consultar ficheros.

Esta opción permite consultar directamente un fichero, sin pasar por el proceso de listado, útil en el caso de que la lista de ficheros sea considerable.

En este momento podemos acceder a una de las opciones más relevantes del proyecto, que consiste en visualizar con que otros ficheros se relaciona el presente y por medio de que campo.

4.4 Listar ficheros:

Mediante esta opción, accedemos a la consulta de ficheros de igual forma que en el paso anterior, pero seleccionando el fichero de una lista, mediante un “option”.



Nombre del fichero	Detalle	Numero del fichero	Numero de particiones	
ADA_FORMATO_PERSONAS	Fichero de personas fisicas	1	1	◉
ADA_FORMATO_CUENTAS	Fichero de Cuentas de cotizacion	2	1	○
ADA_FORMATO_TRABAJADORES_ACT	Fichero de Situaciones actuales del trabajador	3	26	○
ADA_FORMATO_TRABAJADORES_ANT	Fichero de Situaciones anteriores del trabajador	4	26	○

Listar Volver

Mostrando de igual forma todos los detalles del fichero.

Nombre del fichero	ADA_FORMATO_PERSONAS				
Campo	Nivel	Formato	Longitud	Tipo	Descripción
DNI	1	A	10	DESCRIPTOR	
NOMBRE	1	A	15		
APE1	1	A	20		
APE2	1	A	20		
NUM_TRABAJADOR	1	N	10		
DOMICILIO	1	A	52	GRUPO	
TIPO_VIA	2	A	2		
NOMBRE_VIA	2	A	36		
NUMERO_VIA	2	N	5		
BIS_VIA	2	A	2		
ESCALERA	2	A	2		
PISO	2	A	2		
PUERTA	2	A	3		

Ver relaciones Volver

5. Agradecimientos:

- A la Gerencia Informática de la Seguridad Social por apostar por sus empleados.
- A la UAH por aportar todo el equipo tanto técnico como humano, en particular a D. José Maria Gutiérrez, tutor del proyecto.
- A los compañeros, por amenizar los momentos mas duros y ayudar puntualmente a resolver ciertas dudas.

6. Referencias:

- **Unified Modeling Language**. http://www.uml.org.
- **Sun Microsystems.** http://www.**java**.com/es/download.
- **NetBeans.** http://www.netbeans.org.

Sistema para la gestión de autorizaciones de aplicaciones a usuarios en la intranet provincial de Huesca. Aplicación AUTAPLI

Mª José Foncillas Sanz

Unidad Provincial de Informática de Huesca
Gerencia de Informática de la Seguridad Social,

Resumen. Este proyecto consiste en la creación de una aplicación web que permita llevar a cabo la gestión de los accesos a las aplicaciones web de la intranet provincial de la Dirección Provincial Conjunta de Huesca. La aplicación permitirá gestionar las autorizaciones que los usuarios SILCON de la organización tienen sobre las aplicaciones existentes y la obtención de informes sobre las aplicaciones y las autorizaciones gestionadas. Al ser una aplicación web, será accesible desde cualquier ordenador de la organización que disponga de un navegador. La aplicación será gestionada por el personal de la Unidad Provincial de Informática que son los encargados de la administración de los accesos a las aplicaciones propias.

1. Introducción

El objetivo fundamental del aplicativo desarrollado, con el nombre AUTAPLI es administrar los accesos a las aplicaciones propias de la Dirección Provincial Conjunta por parte de los usuarios de la misma. Al igual que el sistema de confidencialidad SILCON de la Seguridad Social controla los accesos a las distintas aplicaciones de la organización, el aplicativo AUTAPLI se encargará de mantener las relaciones *Usuario-Aplicación* que posibiliten el acceso a las aplicaciones residentes en sus servidores.

Las aplicaciones web propias de la DPC son accesibles a través de la intranet provincial; en función del usuario conectado aparecen sólo aquellas aplicaciones a las que tiene acceso. Para poder determinar qué aplicaciones tiene autorizadas cada usuario, es necesario que existan asociaciones entre aplicación y usuario autorizado.

En la actualidad existe una sencilla aplicación que realiza la gestión de autorizaciones de forma individualizada y con una interfaz de usuario poco elaborada. Con el desarrollo de este proyecto se pretende mejorar las funcionalidades existentes y el entorno de trabajo del cliente.

Nota

El texto completo de la presente ponencia no se ha incluido en el libro de actas de las Jornadas por tratar temas confidenciales y será objeto de una separata que se entregará a los asistentes durante su presentación oral.

Sistema de Gestión de Vacaciones

Mª del Carmen García Alcaraz

Gerencia de Informática de la Seguridad Social.
Centro de Coordinación. Área de Desarrollos Externos.

Resumen. El principal objetivo de este trabajo es aplicar los conocimientos adquiridos durante los dos años de estudios del Plan Ática, especialización en Desarrollo de Aplicaciones Web, creando una aplicación que realice la gestión de las solicitudes de vacaciones del personal de una empresa, utilizando tecnología Java, un gestor de base de datos relacional y un entorno Web en el que el usuario utilice un navegador de Internet como vía de acceso a la aplicación.

1 Introducción

El propósito fundamental de esta aplicación, que denominaremos SGV, será facilitar la solicitud por parte de los empleados de una empresa de sus vacaciones, automatizando el proceso completo de las solicitudes, desde la aprobación/rechazo de las solicitudes por parte del gerente como responsable directo, hasta la validación por el responsable administrativo, manteniendo en todo momento la información accesible a los distintos usuarios del sistema, en función del rol que tengan asignado, pero ofreciendo la confidencialidad necesaria en el acceso a la información.

1.1 Objetivos

El sistema desarrollado ofrecerá una serie de ventajas y beneficios a los usuarios:

- Para los empleados:
  - Sistema basado en Web que permite al usuario acceder a la aplicación de una forma sencilla y realizar sus solicitudes.
  - Poder conocer el estado actual de todas sus solicitudes.
  - Recibir notificación de las solicitudes rechazadas y de las validadas.
- Para los responsables:
  - Acceso a los datos de todas las solicitudes del personal a su cargo.
  - Facilidad a la hora de acceder a las solicitudes que aún no han sido gestionadas y de aprobar/rechazar o validar las solicitudes.
  - Obtención de estadísticas y gráficas de distribución de las solicitudes, permitiendo controlar y evitar la existencia de puntos críticos.
  - Reducir el uso de papel y el trasiego de documentación impresa entre los responsables, evitando así que pueda ser accedida por terceras personas.

1.2 Restricciones del sistema

Las únicas restricciones del sistema serán las referentes al acceso a la aplicación, que estará limitado a los empleados de la empresa, y una vez dentro de ésta, a las tareas que puede realizar el usuario en función del rol al que pertenece.
Para gestionar este control se utilizarán claves de acceso asignadas a los usuarios. Además, cada usuario tendrá un rol o perfil dentro del sistema, que le será asignado al ser dado de alta, pudiendo ser:

- SOLICITANTE : Todos los usuarios de la aplicación tienen este rol.
- GERENTE : Responsable de uno o varios Solicitantes de una gerencia.
- RESPONSABLE ADMINISTRATIVO : Está a cargo de todos los gerentes.

1.3 Requisitos del sistema

Se identifican los requisitos que deberá cumplir nuestro sistema, entre ellos:

- Permitir realizar solicitudes de vacaciones a los usuarios de la aplicación.
- Identificación de los usuarios del sistema mediante clave de acceso.
- Control del número de días de vacaciones a solicitar.
  Inicialmente cada empleado dispone de 22 días laborables anuales, de los que habrá que ir descontando días en función de las solicitudes realizadas.
- Cada solicitud debe pasar primero por el gerente, que podrá aprobar o rechazar la solicitud.
- Las solicitudes aprobadas por el gerente pasarán al control del responsable administrativo para su validación e impresión para firma.
- Las solicitudes que han sido rechazadas por el gerente y las validadas por el responsable administrativo son notificadas al solicitante de forma automática.
- El usuario de tipo Solicitante debe poder crear nuevas solicitudes de vacaciones, consultar sus solicitudes y dependiendo del estado en que se encuentren podrá eliminar la solicitud o modificar algún dato y visualizar las solicitudes de vacaciones ya validadas por el administrativo.
- Ente las funciones que puede realizar el usuario de tipo Gerente destacamos la consulta de las solicitudes realizadas por los empleados pertenecientes a su gerencia, visualizando los datos de cada solicitud. Una vez consultada la solicitud el gerente la podrá Aprobar o Rechazar. También podrá obtener estadísticas para saber cuanta gente de su gerencia ha solicitado un periodo concreto, o que proyectos pueden quedar sin personal. Esta información se visualiza gráficamente en un calendario.
- Dentro de las funcionalidades del usuario de tipo Responsable Administrativo está la validación de las solicitudes de vacaciones realizadas en cada una de las gerencias y que ya han sido aprobadas por el gerente. Una vez validada la solicitud se podrá imprimir para que el responsable la firme.

2 Análisis del Sistema

2.1 Catálogo de requisitos

Descripción de los requisitos del sistema, diferenciando según el tipo de requisito. Aunque en el proyecto se han definido muchos más requisitos, aquí describiré a modo de ejemplo alguno de ellos, tanto de tipo funcional, de seguridad y de datos.

- **RQ-F-01** Requisito funcional Prioridad: ALTA
  La configuración del menú con las funciones que puede realizar un usuario depende del TIPO de USUARIO (otorgado por el Admin. del Sistema).
- **RQ-F-04** Requisito funcional Prioridad: ALTA
  Cada SOLICITANTE puede listar todas sus SOLICITUDES (en cualquier ESTADO), consultando todos los datos de la solicitud. Obteniendo además el acumulado de días en cada ESTADO, así como el número de días pendientes.
- **RQ-F-05** Requisito funcional Prioridad: MEDIA
  El sistema permitirá al SOLICITANTE visualizar sus solicitudes de forma gráfica en el calendario del año actual.
- **RQ-F-08** Requisito funcional Prioridad: ALTA
  El sistema permitirá al GERENTE consultar los datos de cada una de las solicitudes pendientes de aprobar y APROBAR / RECHAZAR la solicitud.
- **RQ-F-09** Requisito funcional Prioridad: MEDIA
  Cuando el GERENTE rechaza una solicitud el sistema notifica mediante un CORREO al SOLICITANTE que su solicitud ha sido RECHAZADA por el GERENTE.
- **RQ-S-01** Requisito de seguridad Prioridad: ALTA
  No permitir el acceso a la Aplicación SGV a usuarios ajenos a la empresa. Este control se gestiona mediante USUARIO y PASSWORD almacenados en BBDD.
- **RQ-S-02** Requisito de seguridad Prioridad: ALTA
  Cada USUARIO tiene un rol definido en el momento de darlo de alta en el sist. Este rol o TIPO de USUARIO determina las funciones que puede realizar el USUARIO, determinando las opciones del MENU de acceso a la aplicación.
- **RQ-D-01** Requisito de datos Prioridad: ALTA
  El campo de ESTADO de una solicitud siempre será de salida, no puede ser modificado por un usuario. El cambio de ESTADO será siempre fruto de un proceso en la lógica de la secuencia que debe seguir una solicitud dentro de la aplicación.

2.2 Diagrama de Casos de Uso

Mediante los diagramas de casos de uso representamos el modelo de comportamiento, detallando las funciones que el sistema ofrece a los usuarios, definiendo actores, casos de uso y las relaciones existentes entre ellos.

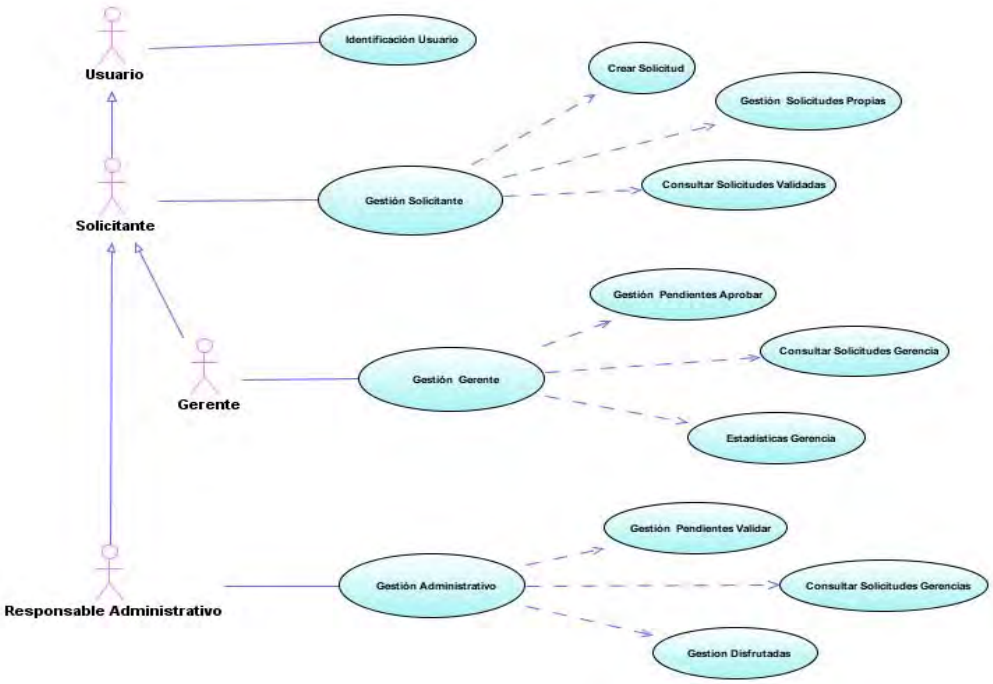


Figura 1. Diagrama de Caso de Uso SGV

Veamos a modo de ejemplo el desarrollo de los Casos de Uso de Gestión Gerente.

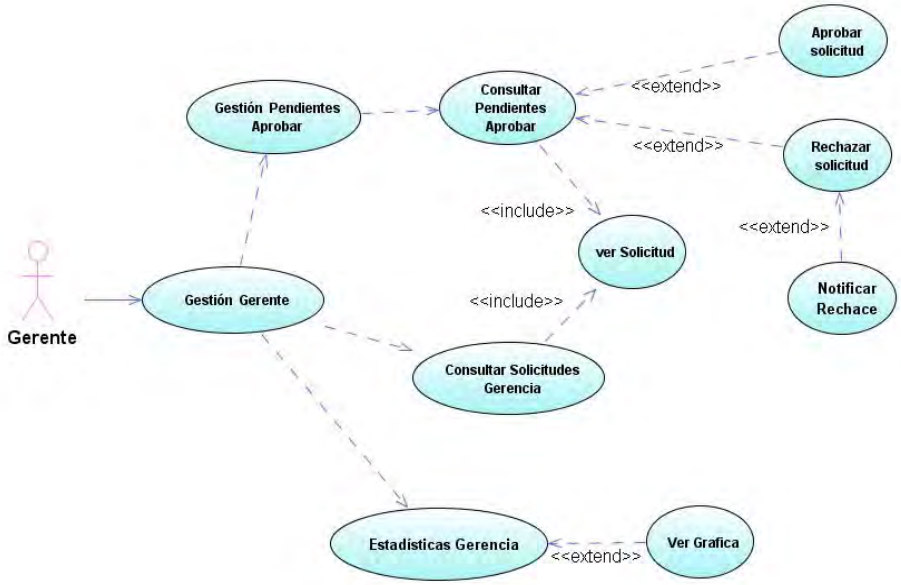


Figura 2. Diagrama de Caso de Uso Gestión Gerente

2.3 Modelo de Objetos Conceptual

Con este diagrama representamos las entidades y las relaciones existentes entre ellas.

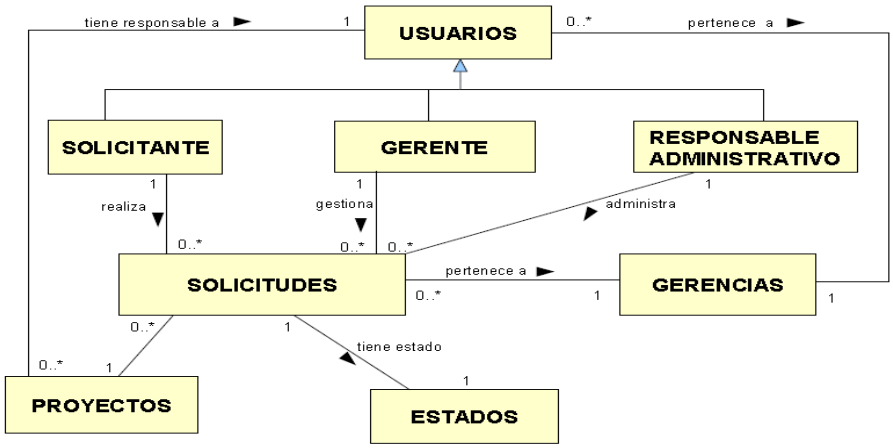


Figura 3. Modelo de Objetos

2.4 Diagrama de clases

A partir del Modelo de Objetos Conceptual definimos las clases (del dominio de negocio) del sistema. En nuestro caso debemos resolver como implementar la relación de herencia de las clases Solicitante, Gerente y Responsable Administrativo con la superclase Usuarios. Como las subclases no tienen atributos propios, se considera que la mejor solución es definir una única clase Usuarios que tenga un atributo que indique el tipo de usuario.

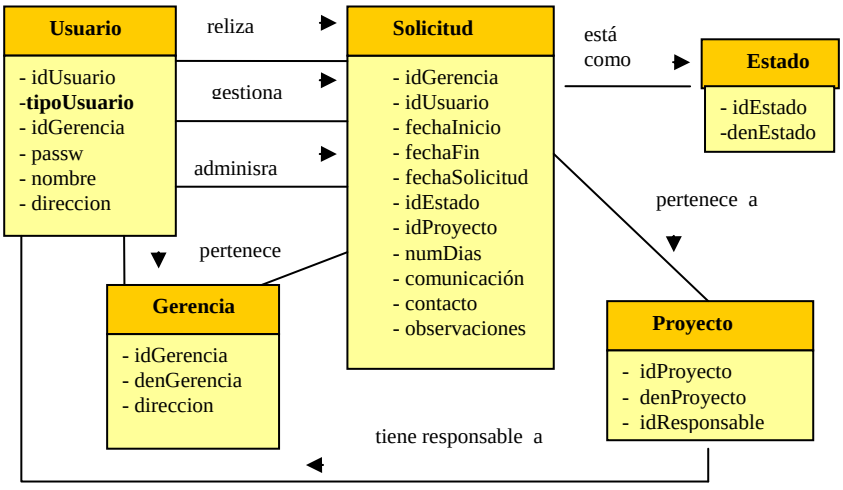


Figura 4. Diagrama de Clases en fase de análisis

2.5 Diagrama de Estados

Un diagrama de estados muestra la secuencia de estados por los que pasa un objeto a lo largo de su ciclo de vida, indicando que eventos hacen que pase de un estado a otro, así como las respuestas y acciones que genera. Este tipo de diagrama es muy útil en el caso de nuestro objeto Solicitud, siendo la secuencia de estados de una solicitud la representada a continuación y coincidiendo cada estado con uno de los existentes en la tabla de Estados (Petición, Aprobada, Rechazada, Validada, Disfrutada).

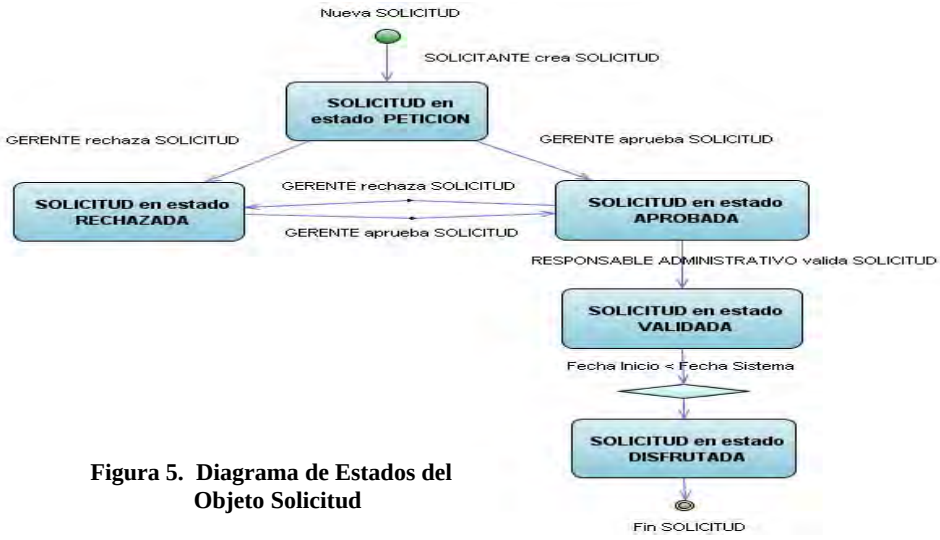


Figura 5. Diagrama de Estados del Objeto Solicitud

2.6 Interfaces de Usuario

La interfaz del sistema con el usuario queda determinada por todas las pantallas e informes que forman la aplicación. Se ha tratado de mantener un diseño y funcionamiento lo más unificado posible, utilizando para ello hojas de estilo. Todas las pantallas, excepto la inicial de conexión, están formadas por un área izquierda donde se visualiza el logotipo de la aplicación, el nombre del usuario, la gerencia a la que pertenece y el menú con las opciones disponibles según el tipo de usuario. En la parte inferior existe el botón *Desconectar* que permite al usuario cerrar su conexión y volver a la pantalla inicial del sistema. Cuando el usuario tiene alguna notificación
referente a sus solicitudes de vacaciones aparecerá el botón *Correo* que al pulsarlo mostrará las notificaciones enviadas al usuario por los responsables de la tramitación de sus solicitudes cuando éstas han sido rechazadas por el gerente o validadas por el responsable administrativo. El resto de la pantalla conforma el área de contenido, donde se presentará la funcionalidad seleccionada en el menú. El área tendrá tamaño fijo, con barra de desplazamiento vertical si es necesario. Tendrá una cabecera con el tipo de usuario y la opción seleccionada en el menú, que se visualizará en diferente color según el perfil del usuario.

3 Diseño del Sistema

3.1 Niveles de arquitectura

La arquitectura lógica del sistema desarrollado es un modelo de tres capas o niveles, donde podemos diferenciar claramente la capa de servicios de usuario o capa de presentación que contiene las interfaces de usuario, la capa de servicios de negocio que es la que contiene la lógica o dominio del problema y una tercera capa de servicios de datos formada por la base de datos y el sistema gestor de la base de datos.

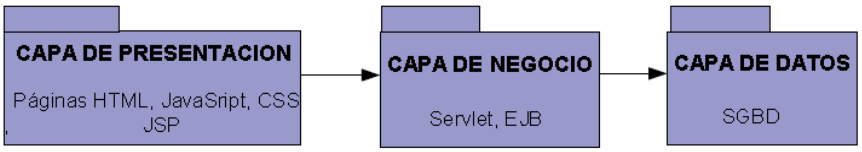


Figura 6. Niveles de arquitectura del sistema

Esta arquitectura se basa en el patrón **Modelo-Vista-Controlador**. El objetivo de este patrón es separar el modelo de datos de su representación de cara al usuario y de la interacción de éste con la aplicación. Este objetivo se consigue mediante la división de la aplicación en tres partes:

- El Modelo, que contiene la lógica de negocio de la aplicación.
  Es implementado por un conjunto de clases Java, utilizando para ello los EJB (Enterprise JavaBeans).
  En nuestro sistema emplearemos EJBs de Entidad (Entity EJBs).
- La Vista, que es la encargada de mostrar al usuario la información, utilizando para ello páginas web. La tecnología empleada será páginas JSP (Java Server Pages) que nos permiten generar las páginas dinámicamente. Utilizaremos CSS (hojas de estilo) en el diseño de las páginas.
- El Controlador, que recibe e interpreta la interacción del usuario, actuando sobre el modelo y la vista de la forma adecuada para producir cambios de estado en los datos. En nuestro sistema se desarrolla mediante Servlets, que actuarán de intermediarios entre vista y modelo.

3.2 Diseño de clases

En esta fase de diseño se transforma el modelo creado en la fase de análisis, obteniendo el diagrama de clases de diseño, donde definimos todas las clases que intervienen en el sistema, tanto las del nivel de datos como las del nivel de negocio, donde detallaremos sus atributos y las relaciones existentes entre las clases.

3.3 Diseño físico de datos

En función de los requerimientos del sistema establecidos en los requisitos de datos y de seguridad, se ha diseñado una base de datos con la estructura física representada en el siguiente Modelo Entidad-Relación. Además de las entidades representadas en el Modelo E/R, se crean otras tablas que permitan gestionar ciertas funcionalidades de la aplicación (CALENDARIOS, MENUS, etc..).

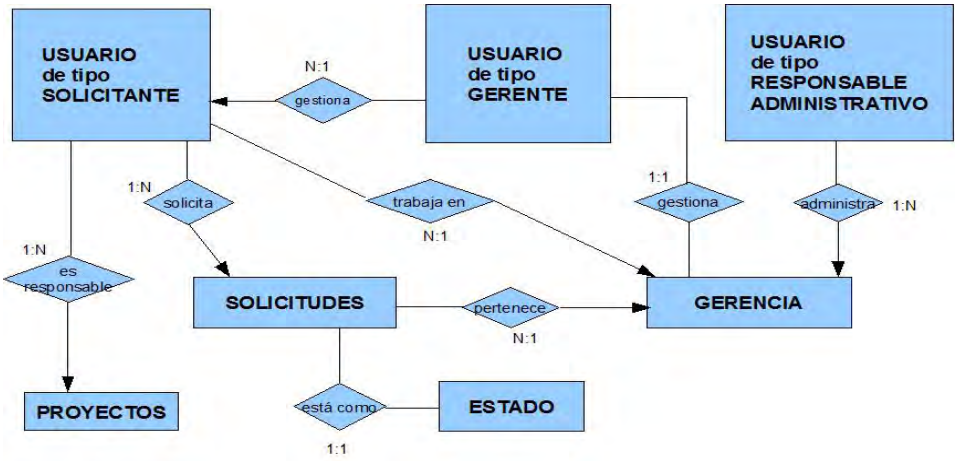
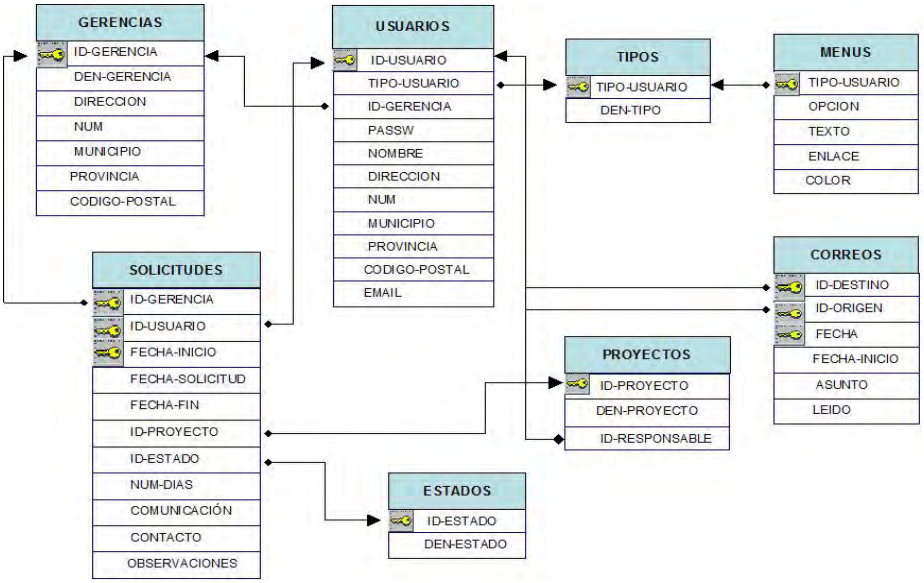


Figura 7. Modelo Entidad Relación

Partiendo del Modelo E/R realizamos el diseño lógico de las tablas que conformarán nuestra base de datos.



4 Funcionamiento de la aplicación SGV

Una vez que el usuario accede a la aplicación introduciendo su clave de acceso, el sistema generará el menú correspondiente según el tipo de usuario y las funcionalidades que tiene permitidas. El menú generado será uno de los siguientes:

Pantalla Solicitante

Pantalla Gerente

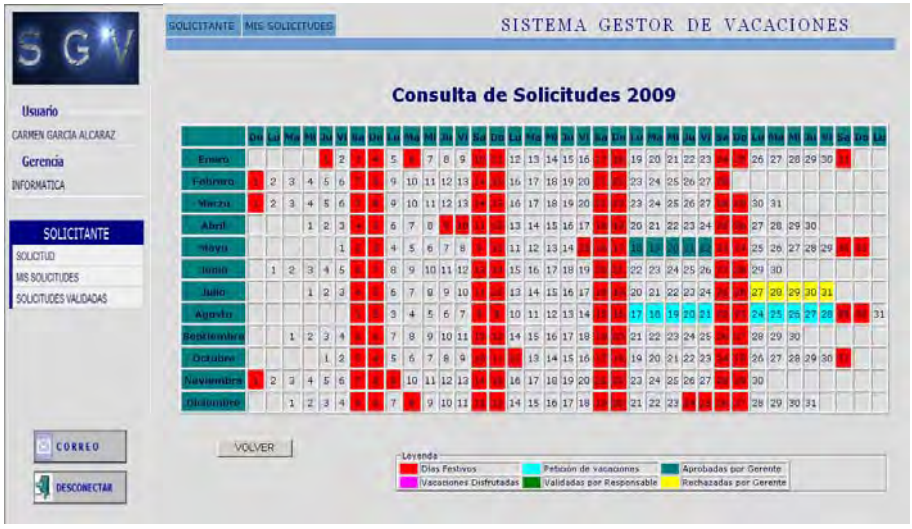
Pantalla Administrativo

Veamos algunas de las pantallas y funcionalidades que conforman la aplicación.

4.1 Funcionalidad Solicitante

Pantalla Crear Solicitud

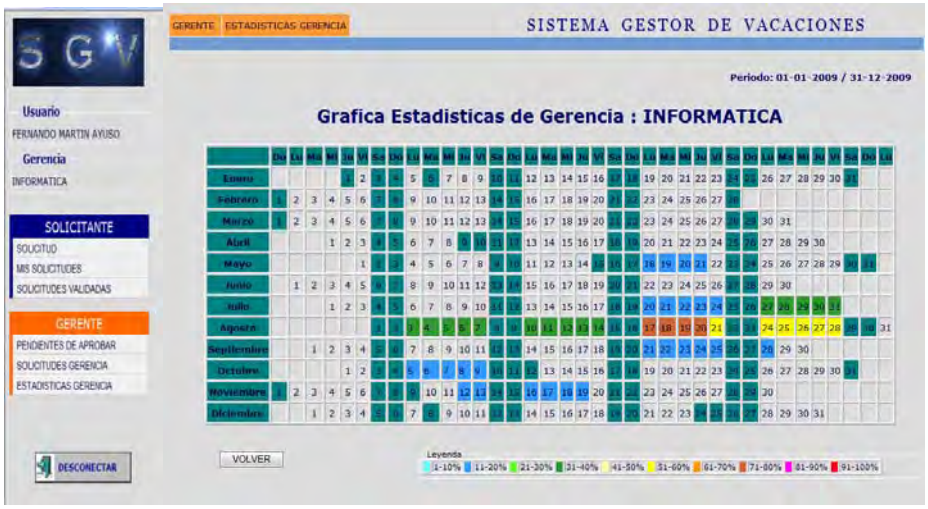
Pantalla Ver Solicitud en Consultar Solicitudes



Pantalla Ver Gráfica en Consultar Solicitudes , visualiza las solicitudes de un solicitante en el calendario del año actual, utilizando diferentes colores según el estado en que se encuentre cada solicitud.

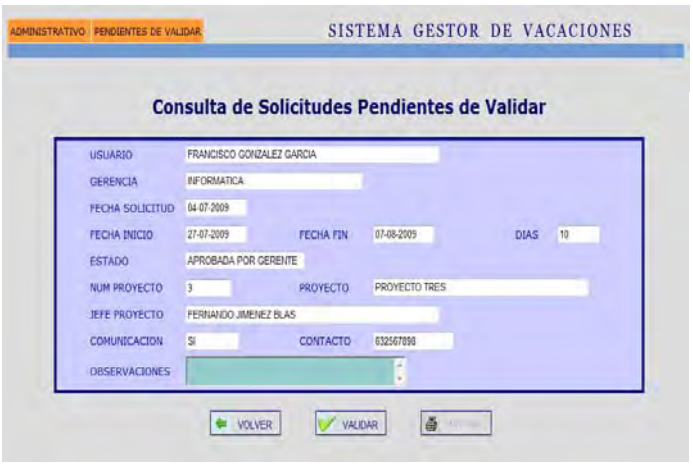
4.2 Funcionalidad Gerente

El SGV ofrece al gerente las herramientas necesarias para tener una visión general de las solicitudes de vacaciones del personal a su cargo y poder realizar la tarea de aprobar o rechazar las solicitudes en función de estadísticas e informes generados. Entre las pantallas que utiliza destacamos la que visualiza gráficamente las estadísticas, mostrando el tanto por ciento de solicitudes para cada día.



4.3 Funcionalidad Responsable Administrativo

Las herramientas que el SGV ofrece al responsable administrativo le permiten tener una visión general de las solicitudes de vacaciones del personal de todas las gerencias, validar las solicitudes ya aprobadas por el gerente y obtener el impreso para su firma.



Una vez validada la solicitud se activa el botón IMPRIMIR que al ser pulsado genera un fichero XML con los datos referentes a la solicitud, asociándole un XSL y una hoja de estilo para dar al documento el formato deseado.

4.4 Funcionalidad Notificaciones

El SGV se encarga de notificar automáticamente al solicitante cuando una solicitud suya es rechazada por el gerente o cuando es validada por el responsable administrativo. El solicitante que tiene notificaciones tendrá activado el botón *Correo* en su menú y al pulsarlo se presenta en pantalla todas las notificaciones que tenga pendientes.

Por cada notificación tenemos la posibilidad de consultar la solicitud que la ha producido mediante el botón *Ver* que nos mostrará la pantalla de consulta de una solicitud. Una vez que se ha consultado la notificación el usuario podrá borrarla.

5 Conclusiones

Una vez completada la fase de diseño se realizaron pruebas del sistema, dando como resultado un correcto funcionamiento de la aplicación SGV, lo que hace que personalmente me sienta satisfecha con el resultado obtenido al constatar el aprovechamiento de los conocimientos adquiridos durante estos dos años de estudios, así como la necesidad de investigar sobre otros aspectos desconocidos por mí hasta ese momento.

6 Referencias y Bibliografía

- Calendario selector de fechas: HTML Calendar Widget httpp://www.dynarch.com
- Eckel, Bruce: "Piensa en Java". Prentice Hall Ed. 2007
- Documentación entregada durante los cursos del Plan Ática
  - Apress.Beginning.EJB.3.Application.Development.Sep.2006.pdf
  - Apress.Beginning.Java.EE.5.From.Novice.to.Professional.Oct.2006.pdf
  - Apress.Pro.EJB.3.Java.Persistence.API.May.2006.pdf
  - Head First Servlets and JSP.pdf
  - Manning.EJB.3.in.Action.Apr.2007.pdf
- Java , JSP: http://www.programacion.com/java/
  http://www.java.com/es/
  http://www.forosdelweb.com
  http://www.javahispano.com
- HTML: http://www.programacion.com/html/
- EJB's : http://java.sun.com/products/ejb/docs.html
  http://www.webtaller.com
- Desarrollo Web, CSS. http://www.desarrolloweb.com/
- WebEstilo. http://www.webestilo.com/
- Netbeans. http://www.netbeans.org/index.html
- UML: http://www.uml.org/
  http://es.wikipedia.org

Sistema de Gestión de Equipos Informáticos
Localizador de Elementos en Oficinas (LEO)

Aurelio Gonzalo de Francisco

Centro Provincial de Informática de Zaragoza
Gerencia de Informática de la Seguridad Social
aurelio.gonzalo@giss.seg-social.es

Resumen. LEO es una aplicación Web que desarrolla un Sistema de Gestión de Equipos Informáticos. Integra un Localizador de Elementos en Oficinas que permite buscar equipos y representarlos mediante iconos sobre planos de planta. Desplegada en un servidor de aplicaciones, será accesible desde cualquier equipo de la organización, conectándose a una página web en la Intranet.

1 Introducción

El Sistema de Gestión de Equipos Informáticos LEO (Localizador de Elementos en Oficinas) tiene como objetivo principal mejorar la calidad de la atención al usuario de un equipo informático de la organización y, por ende, la atención al cliente final al agilizar la resolución de las incidencias en los puestos de atención al público.

Está orientado sobre todo a facilitar la localización de dichos elementos y así disponer inmediatamente de información sobre un equipo, el usuario que lo utiliza y los equipos y usuarios de su entorno, con el fin último de contribuir a solucionar los problemas de funcionamiento que se presenten.

LEO dispone de las interfaces de usuario necesarias para el mantenimiento actualizado de la información de oficinas, equipos, usuarios, teléfonos y un histórico de notas donde pueden registrarse las actuaciones llevadas a cabo en un equipo. Y todo esto es posible llevarlo a cabo on-line, porque una información ha de estar actualizada para ser de utilidad.

El mantenimiento del sistema facilitará la ejecución de otras tareas como las de inventario, control de direcciones IP, listas de teléfonos, actuaciones masivas sobre equipos, etc.

2 Requisitos del sistema

Se han identificado los requisitos que se enumeran a continuación:

- En el sistema se distinguirán tres tipos de roles para los usuarios de acceso al sistema: Invitado, Administrador y Administrador General.
- Los usuarios tipo Administrador General estarán a cargo de crear/introducir la información de oficinas y usuarios.
- Los usuarios tipo Administrador podrán crear y editar la información referida a los equipos informáticos, teléfonos, notas, etc. En general tendrán acceso completo al sistema excepto en la parte reservada al Administrador General.
- El usuario Invitado tendrá acceso directamente a consultas, dado que al conectarse a la Intranet de la organización se exige clave de usuario y contraseña. Los usuarios administradores deberán identificarse ante el Sistema LEO con su particular clave de acceso.
- Se creará una base de datos normalizada compuesta de las siguientes tablas: Oficinas, Marcas, Marcas_Modelo, Tipos de Equipo, Usuarios, Administradores, Teléfonos, Equipos, Equipos_Usuarios y Notas.

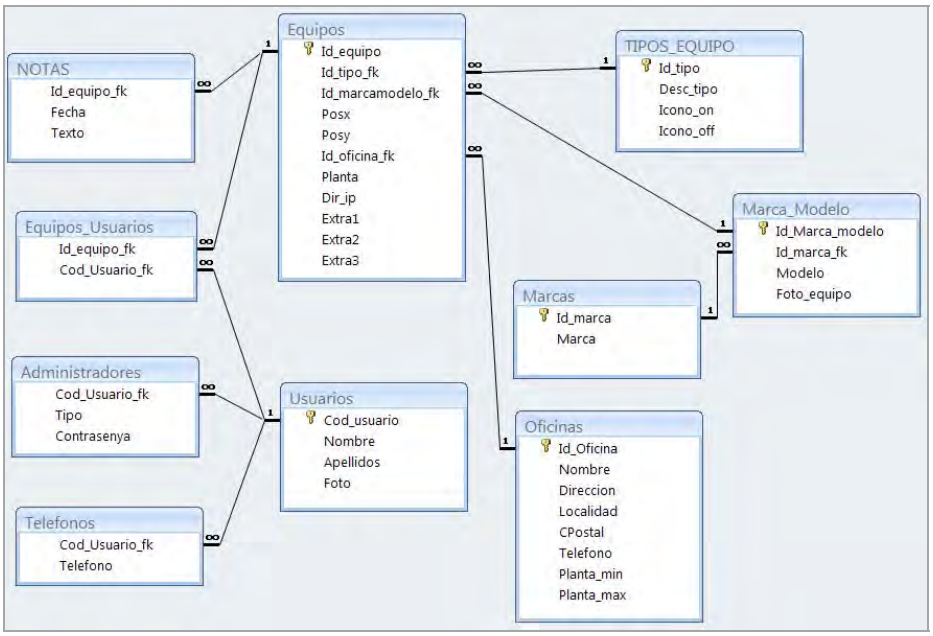


Figura 1. Diagrama de la base de datos.

3 Análisis de Alternativas

La aplicación que se pretende llevar a cabo es una mejora sustancial de la que viene siendo utilizada en la actualidad y que adolece de una tecnología en parte obsoleta y cuyo diseño de base de datos necesita ser revisado.

Se utiliza un antiguo programa MsDos, en lenguaje Clipper, que genera ficheros .xml. Al finalizar cada serie de modificaciones, es preciso volver a regenerarlos.

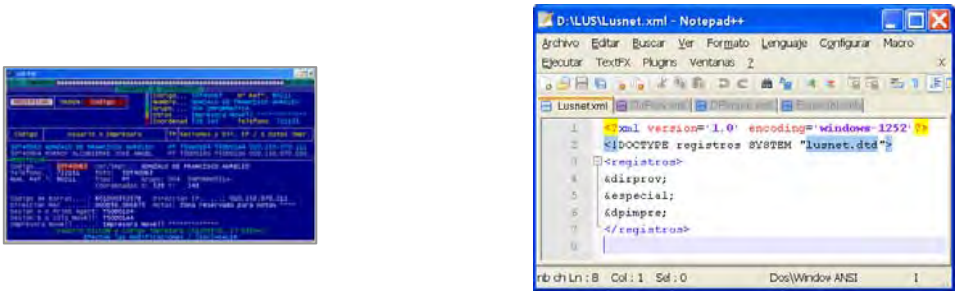


Figura 2. Aplicación, situación anterior.

El acceso remoto desde otras oficinas al programa MsDos es prácticamente inviable. Las consultas remotas son posibles porque los ficheros .xml son colocados en un servidor de ficheros y utilizados por una página web, que funciona únicamente en el navegador oficial de la organización: Internet Explorer y que, además, requiere la instalación en el cliente de un analizador .xml concreto, si bien todos los equipos ya lo incorporan desde su primera instalación.

De la experiencia obtenida con la aplicación actual, se deduce la necesidad de desarrollar una aplicación a medida, que garantice el acceso a la información en todo momento con independencia de las características del equipo cliente y que, además, permita el mantenimiento de los datos on-line a fin de que la información esté siempre actualizada.

Para ello, se propone utilizar tecnología JAVA EE sobre un servidor de aplicaciones, lo que permitirá el acceso con un navegador de Internet, a través de una página web, desde cualquier terminal de la Intranet y con independencia de la localidad en la que se encuentre ubicado.

4 Diagramas

En el documento de análisis del sistema se han incluido los siguentes diagramas detallados de casos de uso:

- Usuario Invitado, acceso al sistema sin permisos especiales.
- Usuarios Administradores (ver figura 3)
- Entrada, acceso al sistema.
- Acceso a los menús de Usuarios, Administradores, Equipos, Teléfonos, Tipos de Equipo, Notas, Marcas, Modelos, Oficinas y Localizador.

Igualmente, se ha incluido el Modelo de Clases.

Como muestra, se reproduce en la fig. 3 el Diagrama de Casos de Uso de Usuarios Administradores.

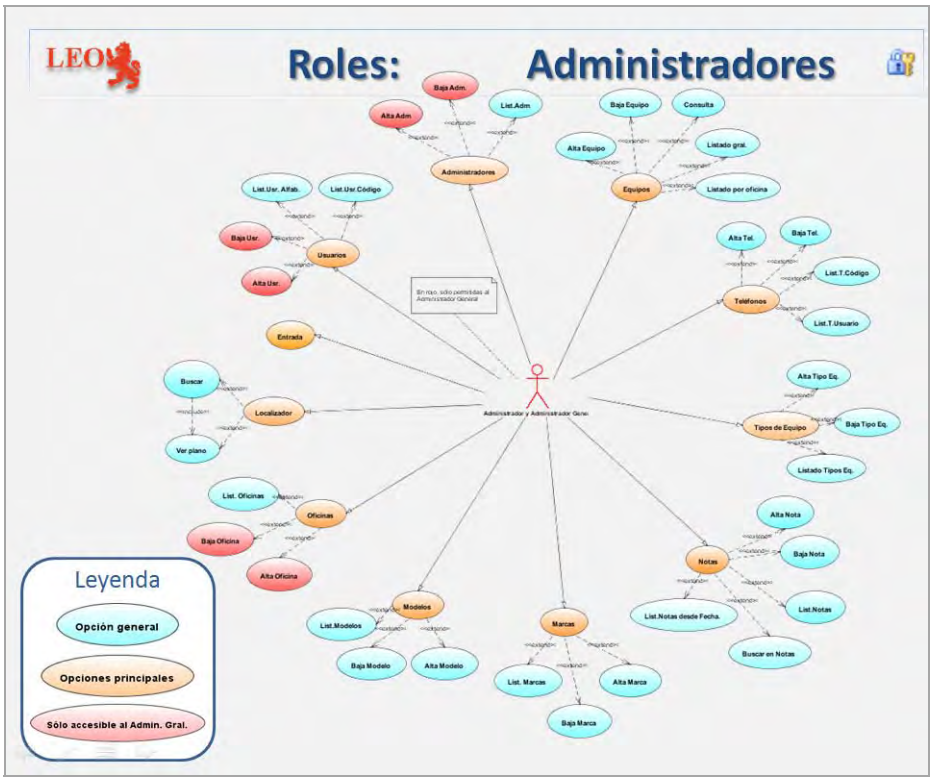


Figura 3. Diagrama de Casos de Uso Usuarios Administradores.

5 Estructura física de la base de datos

El proyecto incluye en este apartado el Diagrama de la Base de Datos reproducido en la fig. 1 y la relación detallada de cada una de las tablas que la componen. Como muestra, se reproduce la Tabla de Administradores:

Tabla de Administradores			
Nombre del campo	**Tipo de datos**	**Tamaño**	**Descripción**
Cod_Usuario_fk	Texto	8	Identificador que toma uno de los valores *Cod_usuario* de la tabla *Usuarios*.
Tipo	Entero	2	Niveles de acceso al sistema: 0 para usuario invitado, 2 para Administrador y 1 para Administrador General.
Contrasenya	Texto	10	Contraseña de entrada al sistema, sólo para usuarios administradores.

Figura 4. Tabla de Administradores

6 Manual de Usuario

Tras realizar una descripción general del sistema, cuyo resumen hemos podido leer ya en la introducción de este artículo, se indica que LEO tan sólo requiere que el equipo cliente disponga de un navegador de Internet, de una versión reciente y con JavaScript activado. La estructura de menús incluye los iconos que el usuario encontrará en la interfaz de la aplicación y las acciones que representan.

Estructura de menús							
Inicio	*Menú*	*Opciones*					
	Localizador			*Buscar*	*Ver Oficina*		
	Usuarios	Altas	Bajas			*List. Alfabético*	*List. Código*
	Administradores	Altas	Bajas			*Listado*	
	Equipos	Altas	Bajas	*Consultas*		*Listado*	*List. Oficina*
	Teléfonos	Altas	Bajas			*List. Número*	*List. Usuario*
	Tipos de Equipo	Altas	Bajas			Listado	
	Notas	Altas	Bajas	Buscar		List. Equipo	List. Fecha
	Marcas	Altas	Bajas			Listado	
	Modelos	Altas	Bajas			Listado	
	Oficinas	Altas	Bajas			*Listado*	
	Cerrar Sesión						

Figura 5. Estructura de Menús

El color utilizado en las opciones de menú se corresponde con los tres tipos de roles para los usuarios de acceso al sistema: ***Verde letra cursiva*** Permitido a todos los usuarios, `Azul letra Courier` Permitido a los usuarios Administrador y Administrador general y **Rojo subrayado** Sólo permitido al Administrador general.

Seguidamente se describen exhaustivamente cada uno de los elementos que integran la aplicación, indicando las instrucciones para la introducción de datos en cada campo de cada formulario y el comportamiento esperado. Todo ello ilustrado con abundantes reproducciones gráficas de formularios y listados.

Basta observar el diagrama de casos de uso reproducido en la fig. 3 para entender la imposibilidad de resumir la gran cantidad de opciones con las que puede interactuar el usuario. Seguidamente reproduciré algunos extractos del manual referidos a algunas de las partes más significativas.

6.1 Alta de Equipos

Figura 6. Formulario de Alta de Equipos

Inicialmente se nos presentan los campos en blanco y las listas desplegables indicando que hagamos una selección.

Instrucciones: **Código** Se introducirá el código de barras que el equipo lleva adherido en una etiqueta con el anagrama de la organización, siempre que lo tenga. En su defecto, otro identificador único. **Tipo** Seleccione uno de los tipos de equipo disponibles en la lista desplegable, que corresponde a los dados de alta con la opción de menú *Gestión de Tipos de Equipo*, ordenados alfabéticamente. Si no lo encuentra en la lista, deberá darlo de alta previamente.
Y se continúa indicando las instrucciones para cada uno de los campos.

Comportamiento: **Código** Si el equipo ya estaba dado de alta, se muestra al usuario el mensaje *"¿Desea consultar sus datos?"*. Si responde afirmativamente se rellena todo el formulario con sus datos del equipo. **Tipo, Modelo y Ubicación** Son listas desplegables donde el usuario selecciona un valor. Al elegir un modelo de equipo, se muestra su fotografía.
Igualmente, se continúa describiendo el comportamiento de cada uno de los campos.

6.2 Listado de Equipos

Este informe presenta la totalidad de los equipos por orden de código. Se ha optado por incluir cada uno de los campos de la ficha de equipos en una columna, aun cuando ello haga que se sobrepase con creces el ancho de la pantalla. No obstante, se ha habilitado la posibilidad de ocultar las columnas que no se consideren interesantes en un momento dado, haciendo un simple clic sobre el título de la columna.
Con List.Oficina podría listar todos los equipos de una misma planta.

Figura 7. Listado de Equipos

6.3 Localizador

El Localizador constituye una parte esencial de esta aplicación hasta el punto de que ha sido en gran parte diseñada para suministrarle la información que precisa. Cuando un usuario que se encuentra atendiendo al público tiene problemas con el funcionamiento de los equipos informáticos, conocer rápidamente los datos del equipo e impresoras que utiliza y de los equipos de su entorno permitirá una respuesta ágil, que redundará sin duda en la calidad del servicio.

Figura 8. Localizador

La pantalla inicial del Localizador muestra unas sugerencias para ayudar a entender las posibilidades que ofrece al usuario. Siguiendo una de sus indicaciones "*Introduzca una secuencia de caracteres lo más corta y significativa posible*", hemos introducido jesus (es indiferente utilizar mayúsculas o minúsculas). La respuesta presenta una tabla con vínculos que al hacer clic nos llevarán al plano de la planta donde se encuentra el puesto de trabajo (ver fig. 9) y donde podrá consultar con facilidad los datos de su equipo y de los de su entorno. La secuencia de caracteres podría ser parte de una dirección IP, el modelo o marca de de una impresora, la extensión telefónica, etc…

Al hacer clic sobre una fila, hará que se muestre ese equipo gráficamente posicionado sobre el plano de la oficina.

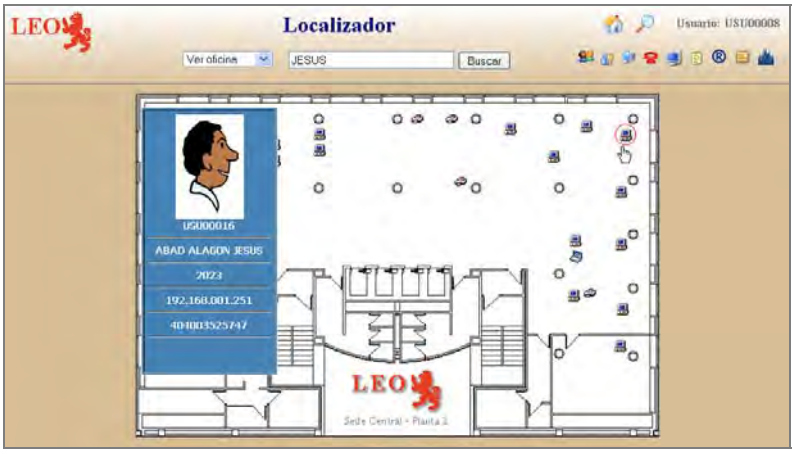


Figura 9. Localizador mostrando un plano

Observe el círculo rojo en la parte superior derecha de la imagen. Indica que esa es la posición en la que está ubicado el equipo. Al acercar el puntero del ratón el círculo desaparece y se muestra la ventana con de datos del equipo y del usuario habitual si lo tiene, que puede ver superpuesta a la izquierda. Al retirar el puntero del ratón, la ventana de datos desaparece y se muestra el plano. El posicionamiento del ratón sobre cualquier otro icono, hará que se muestren sus datos en la parte del plano contraria a su ubicación, a fin de que la ventana de datos no oculte la parte del plano por la que nos desplazamos.

7 Conclusiones.

Aunque la aplicación presenta ya un grado de funcionalidad aceptable, el plazo fijo de entrega del proyecto y su volumen han hecho que no hayan cabido algunas opciones de menú que considero necesarias, ni tampoco otras posibilidades, que se desarrollarán fuera de proyecto, y que ya funcionan como prototipos separados. Por ejemplo, un sistema de posicionamiento de equipos sobre el plano por arrastre, para obtener sus

coordenadas. Unido a una lista con los códigos y datos de los equipos de la oficina, que ya es posible obtener de un programa externo, permitirá una entrada de datos muy rápida y flexible.

Con las técnicas aprendidas en el Plan Ática, el objetivo perseguido de mejorar la calidad de la atención al usuario, es factible. La información necesaria podrá encontrarse al alcance, de una manera rápida y con la ventaja de ser posible su actualización on-line. El esfuerzo realizado ha sido muy grande, pero los conocimientos adquiridos ofrecen un balance final muy positivo.

Desarrollo Aplicación Web. Net "MIALMA"

José Carlos Hernández Roldán.

Unidad Provincial de Informática de Sevilla.
Gerencia de Informática de Seguridad Social.

Resumen. La aplicación "MIALMA" es fruto del proyecto fin de estudio del Plan ATICA (Aula de Tecnologías de la Información y Las Comunicaciones Avanzadas) en la especialidad de Desarrollo Web. Fue propuesta a la Universidad de Alcalá de Henares como práctica alternativa a los supuestos ofrecidos y aceptada en su momento.

Quiero enviar desde aquí el más sincero de los agradecimientos, tanto a la Universidad de Alcalá de Henares como a la Gerencia de Informática de la Seguridad Social por la excelente formación que hemos recibido.

1. Introducción

La aplicación <<Mi almacén de elementos informáticos>>, de aquí en adelante, "MIALMA", se ha desarrollado con la intención de ofrecer un sistema de control e inventario del material informático que la Gerencia de Informática de la Seguridad Social (GISS), pone a disposición y mantiene en la Dirección Provincial y red de Centros de Atención e Información de la Seguridad Social (CAISS) en el Instituto Nacional de la Seguridad Social (INSS) de Sevilla.

Conocer en cada momento, la ubicación de Monitores, Cpu's e Impresoras asignados o asociados a los usuarios de la Entidad.

Poder recabar información a través de la aplicación, de los equipos y elementos que constituyen la herramienta habitual de trabajo de estos usuarios.

Pretende ser la base fundamental de consulta, para proporcionar datos en caso de ser requeridos por personal de la Unidad provincial de Informática (UPI), o de la Gerencia de Informática de la Seguridad Social.

Esta propuesta, tiene como objetivo, ser el complemento de otras herramientas, por ejemplo, el "Action Request System" (Remedy), que a nivel nacional ha implantado la GISS en esta y otras Direcciones provinciales del INSS, la Tesorería General de la Seguridad Social (TGSS) y el Instituto Social de la Marina (ISM).

La aplicación es susceptible y deja abierta la posibilidad de escalar, hacia el control e inventario de servidores, que se asignarán al "*Responsable del Departamento"*, así como inventariar otros elementos, como pueden ser portátiles, relojes de control de incidencias, etc.

2. Descripción

Los "*Usuarios"*, una vez autorizados a través de su código SILCON y contraseña, mediante el aplicativo Web, rellenarán un formulario de alta en el sistema "MIALMA".
El formulario contendrá datos de usuario, datos del equipo o elemento que tiene asignado, se reflejará también, su ubicación (centro o departamento) y roseta a la que va a ser, o ya está conectado.
Los usuarios serán los encargados de solicitar el cambio de ubicación de un elemento informático asociado. Cuando se dé el caso, solicitarán su baja en "MIALMA".
Un equipo informático, que esté asignado a un usuario, no podrá ser asociado a otro, en tanto, no se produzca la baja en el primero.
Los datos se verificarán por los "*Administradores del sistema"*, que podrán ser modificados, en base a los datos obrantes en la Unidad Provincial de Informática. El Administrador se encargará de gestionar el sistema.
En una nueva versión del aplicativo se creará un nuevo rol "*Responsable de Departamento"*, que tendrá acceso a información de los equipos del personal a su cargo. Podrá solicitar la baja de elementos informáticos asociados a uno o varios usuarios de su grupo.

3. Especificación Requisitos Técnicos.

Para el desarrollo del sistema descrito en este documento se utilizó .NET.
El gestor de base de datos para el almacenamiento de la información será SQL-Server 2000 o superior, montado sobre un Sistema operativo Windows 2003 Server, donde estará instalado el Servidor Web con los servicios de IIS (Internet Information Server).
El usuario utilizará el explorador Internet Explorer versión 7.0 o superior para interactuar con la aplicación.

4. Análisis Alto Nivel.

Este sistema, tiene dos tipos de roles, bien diferenciados: *Administrador* y *Usuario* del aplicativo.

El *Administrador* será el encargado de gestionar el aplicativo. Supervisará, dará altas, bajas, consultará y modificará, podrá imprimir determinados listados de los usuarios asociados a los equipos informáticos.
El *Usuario*, una vez validado a través de su código SILCON y su correspondiente contraseña, podrá dar de alta los elementos asociados y la ubicación de estos. Podrá consultar, en el momento que precise o se le requiera, sólo sus datos. Cuando se dé el caso, solicitará su baja o modificación en "MIALMA".

Los datos de *usuario* serán: código SILCON (que lo identifica unívocamente), nombre y apellidos, DNI, teléfono, contraseña. Se incluyen campo tipo persona (administrador/usuario) y situación (alta/baja).
Los datos de *equipo asociado* serán: Nº de GISS (código de barras), modelo, fabricante, IP, roseta y comentarios.
La ubicación del equipo informático se hará a través del código ubicación, incluirá nombre y dirección de los distintos departamentos y centros del INSS.
Por último, se crea una tabla de *elementos informáticos* donde se incluyen categoría y tipo, por si en una nueva versión se decide introducir nuevos elementos.

a. Requisitos funcionales.

En el sistema habrá dos roles:

- Usuarios.
  - ✓ Alta
  - ✓ Consulta.

- El Administrador o Administradores (Gestionan el aplicativo).
  - ✓ Alta.
  - ✓ Consulta.
  - ✓ Modificación.
  - ✓ Borrado.
  - ✓ Listados.

b. Requisitos de datos.

- Usuario.
  - ✓ Se valida mediante SILCON.
  - ✓ Introduce Datos Persona.
  - ✓ Introduce Datos Equipo y Ubicación.

- Administrador.
  - ✓ Se valida mediante SILCON.
  - ✓ Mantenimiento de Usuarios, Equipos, Elementos y Centros.
  - ✓ Explotación mediante Listados.

c. Requisitos no funcionales.

- Usuario (no administrador).
  - ✓ Sólo accede a los datos de su usuario SILCON.

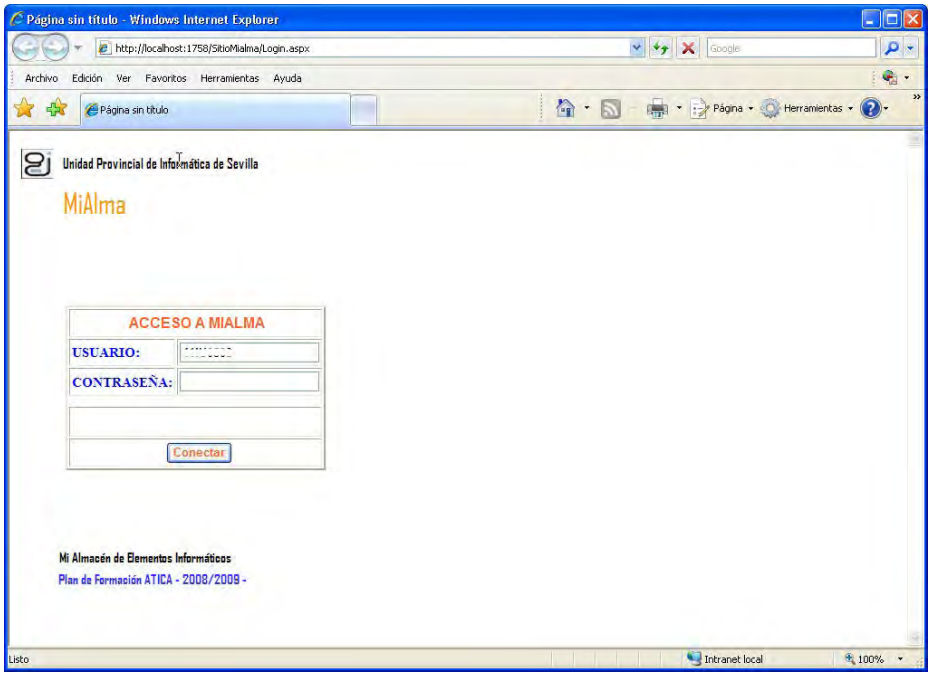
- El Administrador o Administradores.
  - ✓ Acceden a todas las opciones del sistema.
    - Usuarios.

- Departamentos.
- Elementos.
- Equipos.
- Equipos Usuario.

5. Diseño del Sistema.

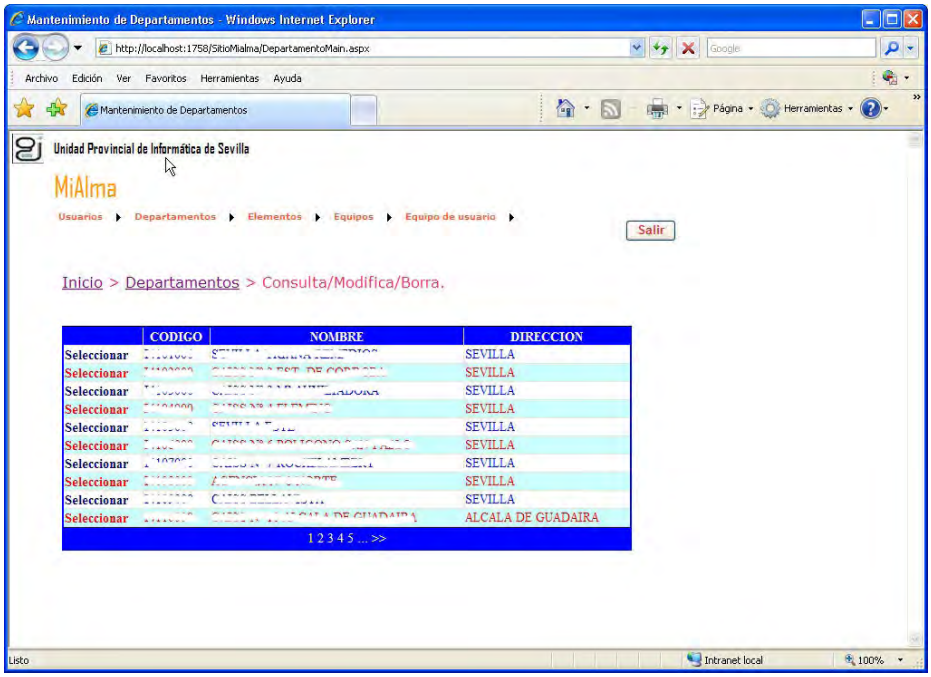
❖ Interfaces.

El acceso al aplicativo se realizará a través de la pantalla realizada al efecto, donde se solicita el código de usuario SILCON y la contraseña.
Los distintos tipos de interfaces, que serán formularios de datos (alta), formularios de mantenimiento (consulta, modificación y en su caso eliminación) y pantallas de informes (listados), son de tonos y colores agradables a la vez que llamativos, con la intención de que se convierta en un entorno amigable y que provoque la atención de los usuarios.
Los menús serán intuitivos, con detalles de la tarea que realizan.



La intranet provincial será la herramienta donde se ubicará el proyecto “MIALMA”.

Los usuarios dan Alta y consultan sus datos a través de formularios creados al efecto. El administrador/es será el encargado del mantenimiento de la intranet.



6. Diseño de la base de datos.

- Tablas de la base de datos.

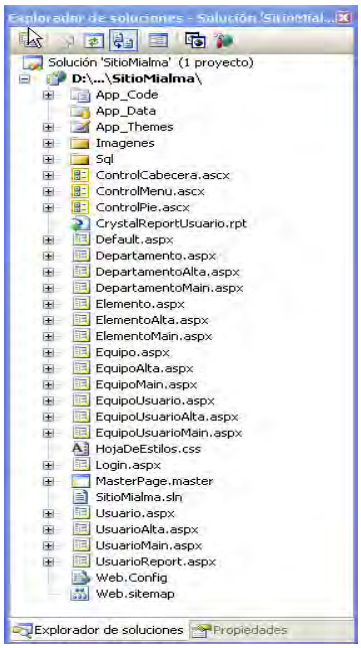
  ✓ USUARIO. Nombre de la tabla que contiene todos los usuarios (usuarios normales y administrador/es) del aplicativo. El campo **SILCON** es clave primaria de la tabla "usuario".

  ✓ EQUIPOASOCIA. Nombre de la tabla que contiene los equipos asociados o asignados a los usuarios de la aplicación. El campo **GISS** es clave primaria de la tabla "equipoasocia".

  ✓ ELEMENTO. Nombre de la tabla que contiene los elementos disponibles para que los usuarios den alta y consulten. El administrador/es gestiona esta tabla, pudiendo introducir nuevos registros. El campo **CATEGORIA** es clave primaria de la tabla "elemento".

  ✓ DEPARTAMENTO. Nombre de la tabla que contiene los departamentos donde están ubicados los equipos asociados de los usuarios de la aplicación. El campo **CODIGO** es clave primaria de la tabla "departamento".

7. Especificación elementos de la aplicación.



❖ Esquema de los ficheros que componen el sistema.

A continuación se describen algunos, de estos ficheros:

✓ Login.aspx. Se trata del fichero, con la página de acceso al aplicativo. Una vez introducido usuario y contraseña, se comprueba que están incluidos en la base de datos del sistema y que la contraseña es correcta.

✓ Default.aspx. Página principal de la aplicación o acceso al aplicativo. Se muestran los datos del usuario validado, y los permisos de que dispone. Dispone de menú contextual, según sea usuario normal o administrador/es.

✓ Web.Config. Es el fichero de configuración global del sistema. Se establece en él, la cadena de conexión con la base de datos “MIALMA”.

8. Casos de prueba.

- ❖ Especificación plan de pruebas.

Con el plan de pruebas se encontraron los posibles errores en la aplicación "MIALMA". Se comprobó que las rutinas funcionan de acuerdo a las especificaciones, en el menor tiempo posible y con el menor esfuerzo.
Se establece una planificación de estas pruebas:

- ✓ Listado de pruebas Unitarias.
- ✓ Listado de pruebas de Integración.
- ✓ Listado de pruebas del Sistema.

A modo de ejemplo:

Identificador PU001	**LOGIN.**
Objetivo	Comprobar que los usuarios del sistema se validan correctamente.
Entrada	Código SILCON y contraseña.
Salida	Pantalla principal

9. Conclusiones.

- ❖ Herramienta de localización rápida de equipos y usuarios.
- ❖ Ámbito de aplicación: Dirección provincial y red CAIS del INSS de Sevilla.
- ❖ Sistema complementario de otras aplicaciones implementadas.
- ❖ En una nueva versión, se podrá escalar hacia el control de servidores, relojes de control de presencia, etc.
- ❖ Los administradores de la aplicación podrán imprimir información contenida en el sistema. Posibilidad de listar los usuarios de la base de datos. Se llevará a cabo, una nueva versión con informes que incluyan listas de equipos por departamentos y/o centros, usuarios, etc.
- ❖ Se han conseguido implementar todas las funcionalidades que se detallaron en la propuesta inicial. Como todos los programas, es susceptible de mejoras, que, con el tiempo y el uso cotidiano, serán requeridos por los administradores y usuarios del aplicativo.

- Se ha elaborado un manual de usuario y otro de administrador.

10. Acrónimos utilizados y glosario.

Autenticación: Proceso de identificar tanto el cliente como el servidor.
CAISS: Centros de Atención e Información de la Seguridad Social.
GISS: Gerencia de Informática de la Seguridad Social.
IIS: Internet Information Server. Conjunto de servicios para los ordenadores que funcionan con Windows. Este servicio convierte a un ordenador en un servidor de Internet o Intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web).
INSS: Instituto Nacional de la Seguridad Social.
ISM: Instituto Social de la Marina.
MIALMA: Mi Almacén (de elementos informáticos).
Nº de GISS: Código de barras de los equipos informáticos.
Servicio Web: Se trata de una aplicación que permite la transferencia de datos sobre HTTP (Protocolo de Transferencia de Hipertexto) y SOAP (Protocolo de Acceso a Datos Simple).
SILCON (Código): Sistema de Información Laboral y de Confidencialidad.
TGSS: Tesorería General de la Seguridad Social.
UPI: Unidad Provincial de Informática.

11. Referencias.

- "Material aportado por la UAH".
- "El libro de Visual C# 2005". James Foxal- Editorial Anaya.
- "Visual Estudio 2008. Desafía todos los retos".dotNetMania" Grupo Weboo.
- "Enciclopedia de Microsoft - Visual C#"- Fco. Javier Ceballos - Editorial Ra-Ma.
- "http://www.lawebdelprogramador.com/"

Gestión de Líneas

Héctor López García
Gerencia de Informática de la Seguridad Social

Resumen. Realización de una Base de Datos y Página Web para la Gestión de las Líneas de Comunicaciones (Voz y Datos). El origen de este proyecto viene determinado por la necesidad de los usuarios internos que gestionan dichas comunicaciones para poder visualizar toda la información que disponen de una forma gráfica y global, sin necesidad de entrar en detalle ya que disponen de una herramienta para ello.

Nota

El texto completo de la presente ponencia no se ha incluido en el libro de actas de las Jornadas por tratar temas confidenciales y será objeto de una separata que se entregará a los asistentes durante su presentación oral.

CitaMed

Aplicación web para el control de las citas previas con el Servicio Médico de Empresa

Carolina Loza Eguiluz

Centro Provincial de Informática de Gipuzkoa

1. Introducción

Los Servicios Médicos de Empresa se crearon por Decreto 1036/1959, de 10 de junio y la Orden de 21 de noviembre de 1959 que aprobó el Reglamento de los mismos, que posteriormente fue modificado por el Real Decreto 39/1997, de 17 de enero y éste a su vez por el Real Decreto 604/2006, de 19 de mayo. Su finalidad era la prestación de servicios sanitarios en las empresas.

La Ley de Prevención de Riesgos Laborales y el Reglamento de los Servicios de Prevención determinaron la integración de los Servicios Médicos de Empresa en los Servicios de Prevención Propios, colaborando, en la actualidad, con las entidades profesionales que desarrollan las tareas de Vigilancia de la Salud en el marco de la Ley de Prevención de Riesgos Laborales.

2. El Servicio Médico en Gipuzkoa. Composición y funcionamiento

El Servicio Médico de Empresa en Gipuzkoa está compuesto por un Médico especialista en medicina del trabajo y un ATS y entre sus actividades está la asistencia sanitaria de urgencia y la atención primaria del personal de las Entidades Gestoras de la Seguridad Social.

Para recibir esta asistencia sanitaria de atención primaria, los empleados tienen que llamar previamente por teléfono al Servicio Médico para concertar una cita.

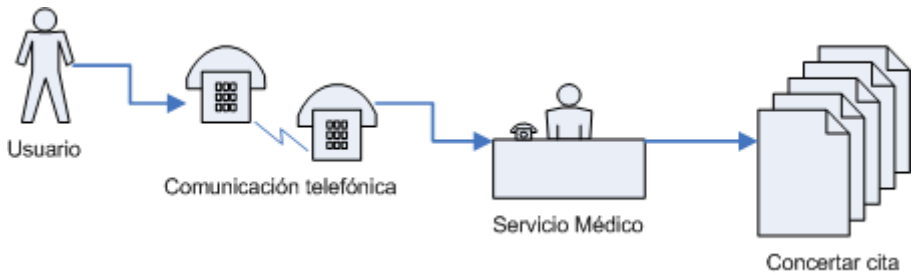


Figura 1 Procedimiento para concertar una cita

Este procedimiento presenta varios inconvenientes:

- Las citas solo se pueden concertar de una en una y en el horario de trabajo del personal médico.
- Este procedimiento supone una sobrecarga de trabajo para el personal del servicio médico.

3. Solución tecnológica desarrollada

Debido a que los potenciales usuarios que se conectan al sistema se encuentran dispersos en varias oficinas a lo largo de la provincia se ha optado por desarrollar una aplicación web.

El servidor de la aplicación y de la base de datos está ubicado en la zona segura de servidores de la Dirección Provincial y los clientes se conectan al servidor a través del navegador de Internet, sin necesidad de ninguna instalación previa, aprovechando la red local.

La aplicación se ha realizado utilizando la tecnología .NET de Microsoft, Framework 3.5. El entorno de desarrollo ha sido Microsoft Visual Studio 2008 y el lenguaje de programación C#. La base de datos se ha construido en un servidor SQL Server 2000.

4. Desarrollo de la aplicación

La aplicación se ha desarrollado basándose en el modelo de tres capas, aunque manteniéndose las tres dentro del mismo proyecto.

4.1. Presentación

La capa de presentación se compone de una serie de páginas aspx con formularios que conforman toda la interfaz de usuario.

Todas las páginas contienen un control de usuario llamado *Cabecera* que se encarga de mostrar el nombre y el departamento al que pertenece el usuario conectado al sistema y el menú que le corresponda según la página en la que se encuentre.

Este control lleva incorporada la referencia a la hoja de estilos CSS para que todas las páginas de la aplicación tengan un aspecto coherente.

Se ha utilizado tanto controles web como controles HTML de servidor.

4.2. Clases de negocio

Se han creado una serie de clases que representan los diferentes aspectos de la capa de negocio de la aplicación.

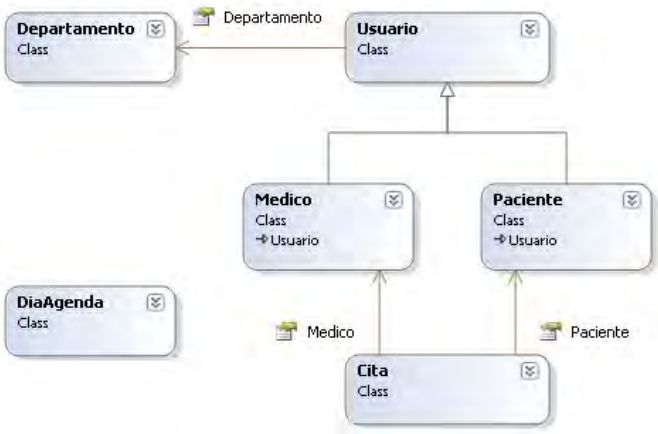


Figura 2 Clases de la capa de negocio de la aplicación

La clase *Usuario* representa cualquier usuario que se conecte al sistema. Es una superclase donde están encapsuladas las propiedades y los métodos comunes de todos los usuarios. Las clases *Medico* y *Paciente*, que representan los diferentes perfiles de usuario que se conectan a la aplicación, heredan de esta clase.
La clase *Cita* es la clase principal de la aplicación y representa una cita concertada entre un usuario paciente y un usuario médico para una consulta médica.

4.3. Clase de acceso a datos

Se ha creado la clase *AccesoDatos* que contiene la capa de acceso a los datos de la aplicación. En ella están los métodos que se encargan de conectarse con la base de datos y devolver los objetos necesarios a la capa de negocio.

Se ha utilizado el objeto *DataReader* de ADO.Net para las consultas a la base de datos. Para ello se utiliza el procedimiento siguiente:

- Se abre la conexión
- Se recupera el DataReader
- Se vuelca la información en los objetos correspondientes de la capa de negocio
- Se cierra la conexión

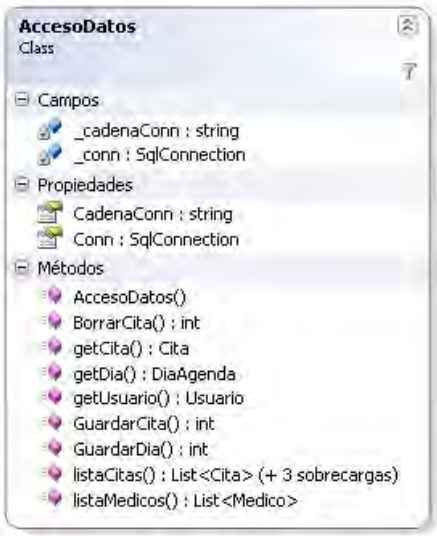


Figura 3 Clase de acceso a datos

El método *GuardarCita*, que se emplea tanto para modificar los registros de las citas como para las altas, utiliza un procedimiento almacenado que se encarga de comprobar que la hora de la cita seleccionada no está ya ocupada y que el usuario paciente no tiene ya una cita concertada para el día seleccionado.

5. Funcionamiento de la aplicación

5.1. Acceder a la aplicación

El acceso a la aplicación se hace a través de una primera pantalla de validación.

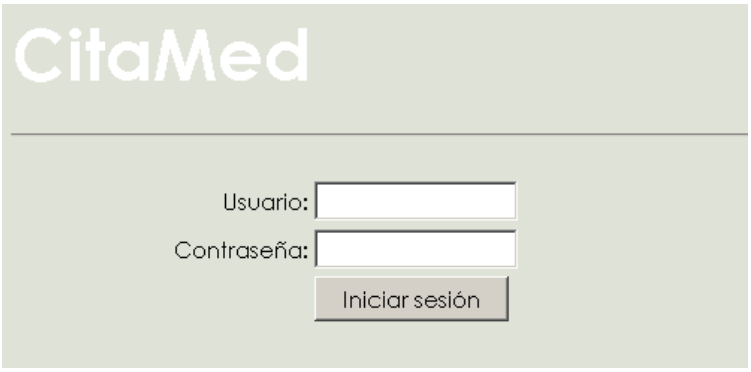


Figura 4 Pantalla de validación de usuarios

A la aplicación tienen acceso dos perfiles distintos de usuarios:

- **Usuario paciente**: Es cualquier empleado de las Entidades Gestoras de la Seguridad Social que se conecta al sistema para concertar una cita.
- **Usuario médico**: Es un miembro del Servicio Médico de Empresa que se va a encargar de atender las consultas médicas. Este usuario puede ser a su vez *Administador* del sistema.

Una vez superado el proceso de validación, la aplicación muestra el menú de opciones correspondiente al perfil de usuario conectado.

Las opciones que tiene un usuario paciente son:

- Concertar una cita nueva
- Consultar mis citas

Las opciones de que dispone un usuario médico son:

- Gestionar agenda
- Gestionar citas

Y si además es un usuario administrador:

- Gestionar usuarios

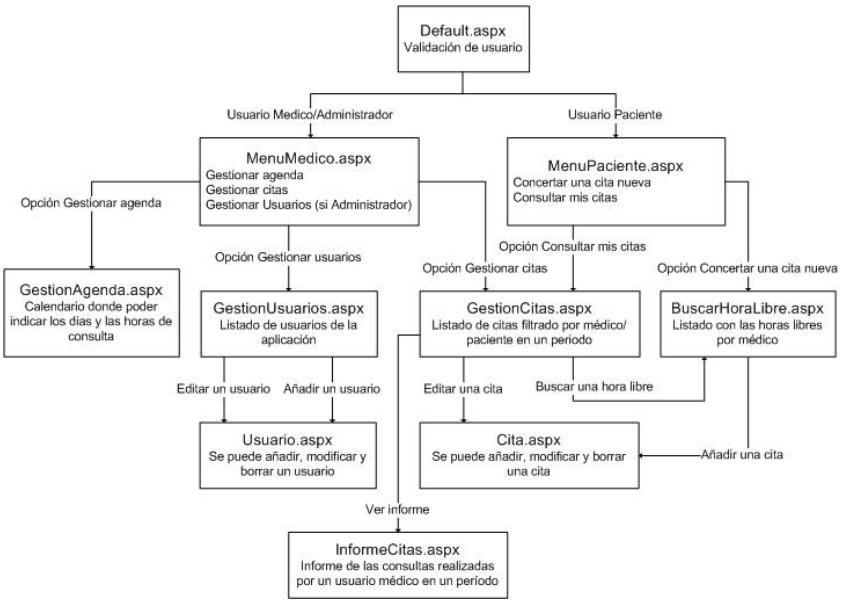


Figura 5 Modelo de navegación de la aplicación

Todas las pantallas de la aplicación disponen de este menú en la parte superior.
También en la parte superior de las páginas se muestra el nombre del usuario conectado al sistema y el departamento al que pertenece.

5.2. Concertar una cita

Cuando un usuario paciente selecciona la opción de *Concertar una cita nueva* se le presenta una pantalla donde debe seleccionar el día y el miembro del Servicio Médico al que desea efectuar la consulta médica.

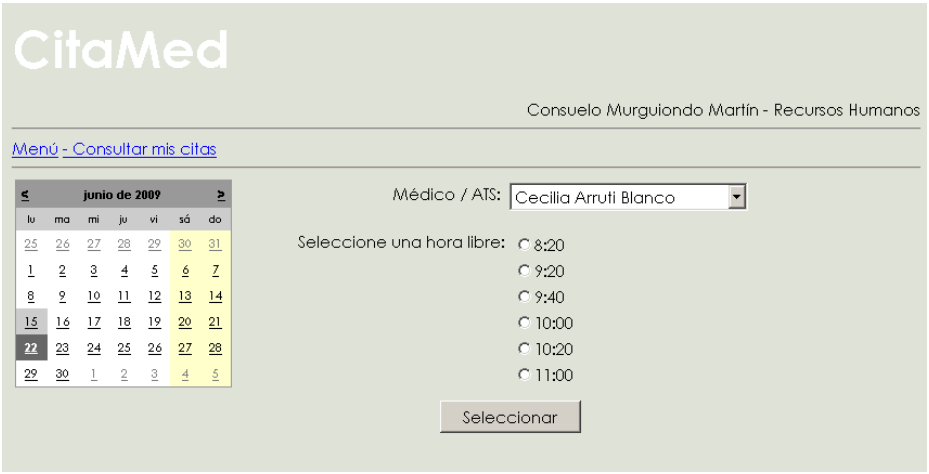


Figura 6 Pantalla para concertar una cita nueva

El sistema presenta una lista de las horas libres disponibles para ese día. El usuario médico debe especificar a través de la opción *Gestionar agenda*, los días que va a pasar consulta, el horario y la duración prevista para cada consulta. A partir de esta información y de las citas ya concertadas, el sistema genera las horas que quedan libres.
Una vez seleccionada la cita, el usuario debe especificar el motivo de la consulta.

5.3. Gestión de citas

Tanto el usuario paciente como el usuario médico tienen acceso a la opción *Gestionar citas*. La diferencia estriba en que el usuario paciente solo puede ver las citas que ha concertado mientras que el usuario médico puede ver las citas que los pacientes han concertado con él. Solo en el caso de que el usuario médico sea además administrador del sistema, podrá acceder a todas las citas.
Para acceder a las citas debe introducirse uno o varios de los criterios de búsqueda que aparecen en el formulario: médico/ATS que atiende las consultas, paciente que ha pedido las citas y/o período de tiempo en que se han concertado las citas.

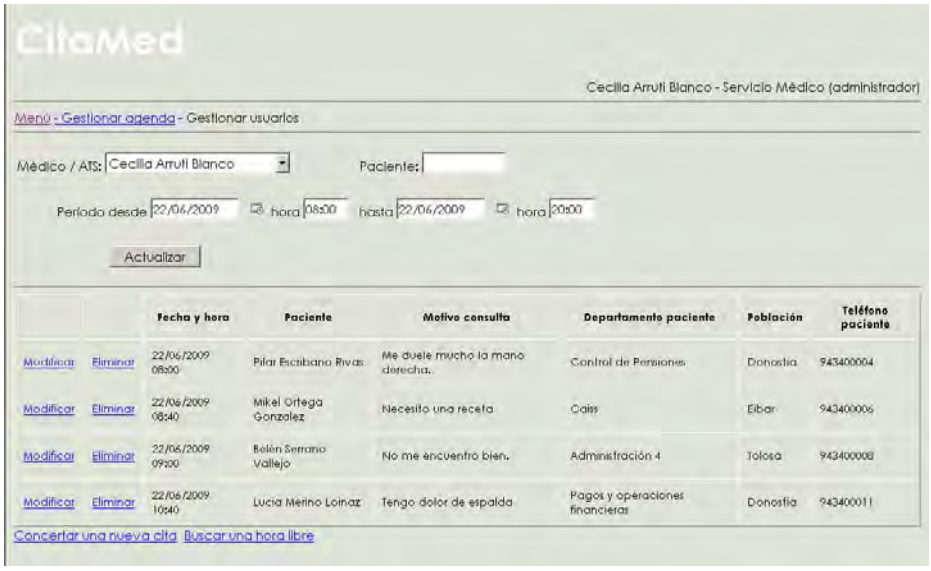


Figura 7 Pantalla de gestión de citas activas

Las citas solo se pueden modificar y/o eliminar si están activas, es decir, si son citas para hoy o para un día futuro.
La información que aparece en pantalla cuando un usuario médico está consultando citas activas es:

- El día y la hora de la cita
- El paciente que va a acudir a la consulta
- El departamento al que pertenece
- La población en la que tiene su puesto de trabajo
- El teléfono del paciente
- El motivo de la consulta

El botón *Modificar* de la lista redirecciona la aplicación a la pantalla donde se editan los datos de la cita.

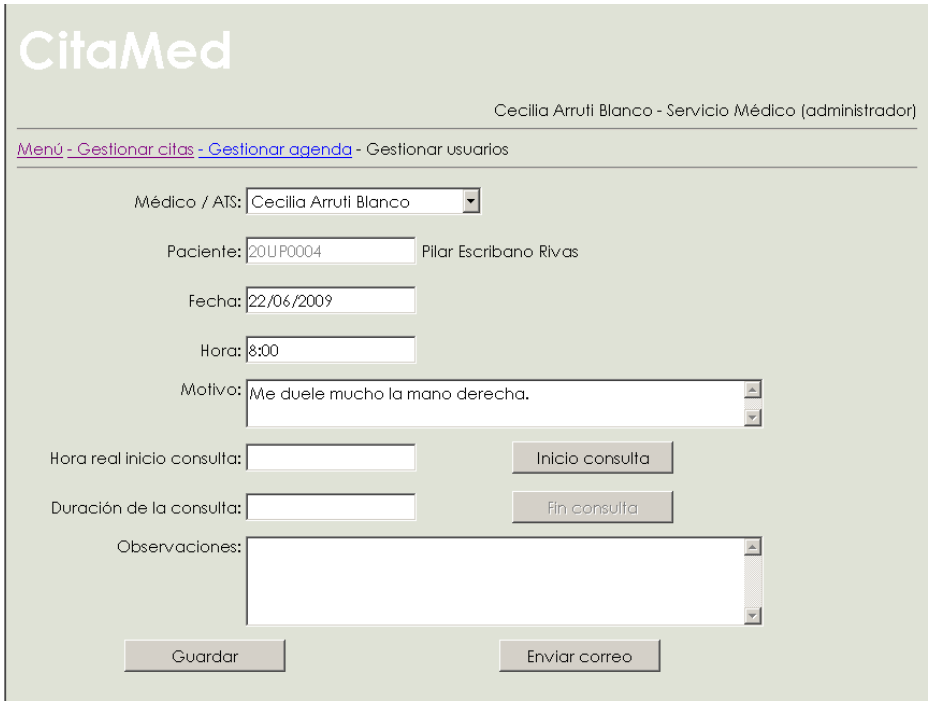


Figura 8 Pantalla de edición de una cita

La información que contiene una cita es la siguiente:

- El usuario médico que atiende la consulta
- El usuario paciente que concierta la cita
- El día de la cita
- La hora prevista de inicio de la consulta
- El motivo de la consulta
- La hora real de inicio de la consulta
- La duración de la consulta
- Las observaciones del médico

La información de la hora real de la consulta y su duración puede introducirla el médico manualmente o haciendo clic en los botones *Inicio consulta* y *Fin consulta* cuando se producen estos momentos.

Cuando un usuario médico está consultando citas pasivas la información que aparece en pantalla es:

- El día y la hora de la cita
- El paciente
- El motivo de la consulta
- La hora real de inicio de la consulta
- Su retraso sobre la hora prevista
- La duración de la consulta
- Las observaciones del médico

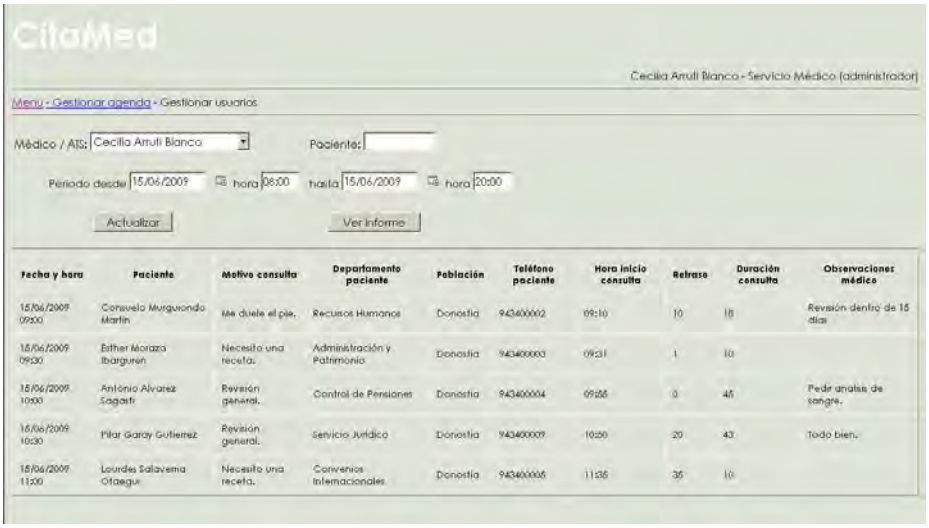


Figura 9 Pantalla de consulta de citas pasivas

6. Conclusiones

Con esta aplicación el personal médico puede gestionar su agenda definiendo su calendario y horario de atención a usuarios.

El usuario puede conectarse al sistema en cualquier momento y asignarse una hora de consulta de entre las que tenga libres el personal médico.

Con esta herramienta el personal del Servicio Médico puede organizar la agenda médica cuando se programan reconocimientos médicos generales.

El personal médico conoce en todo momento el estado de su agenda.

El sistema ofrece al personal médico la posibilidad de ponerse en contacto por correo electrónico con sus pacientes citados para comunicarles un cambio de hora, etc.

Con la información recogida durante la visita la aplicación ofrece al personal médico datos estadísticos sobre la duración de las consultas y la demora en su inicio.

Sistema de Gestión de Confidencialidad

Ana Moreno Gracia
Unidad Provincial de Informática de Tarragona
Antonio Fernández Labrador
Unidad Provincial de Informática de León
Susana López Reche
Unidad Provincial de Informática de Málaga

Resumen.

En este documento se van a exponer todos aquellos requisitos que definan el comportamiento del Sistema de Gestión de Confidencialidad (SGC). Debido a que en las entidades gestoras de la Seguridad Social se trabaja sobre datos sensibles y confidenciales, es necesario que se establezca un control en los permisos de los funcionarios a las diferentes transacciones informáticas que acceden a dichos datos. El sistema de gestión de confidencialidad pretende controlar todas las solicitudes de transacciones que se realicen en una entidad. Dicho proceso se gestionará a través de una herramienta Web (SGC).
El sistema permitirá, por una parte, la gestión de los usuarios, de los grupos de transacciones y de las diferentes áreas organizativas de la entidad. Por otro lado, efectuará el seguimiento y control de los diferentes estados por los que transcurre el ciclo de vida de una solicitud.

1. Introducción

En la actualidad los jefes de área solicitan las transacciones para que el administrador del sistema de confidencialidad realice las gestiones pertinentes para su posterior autorización.

La solicitud de transacciones parte tanto de los jefes de departamento como de los jefes de área directamente, siendo indispensable el visto bueno del jefe de área para su elevación al administrador

Dichas peticiones se hacen por diferentes medios: correo electrónico, formulario en papel con las firmas pertinentes, incidencia comunicada a través del programa de gestión de incidencias, etc.

El administrador debe guardar copia de las peticiones realizadas y gestionar las que cumplan los requisitos exigidos.

Con el nuevo sistema se pretende homogeneizar las peticiones por parte de todas las áreas, almacenar los datos de las solicitudes en una base de datos en la que queda

constancia de los pasos seguidos en su tramitación y utilizar un sistema que permita un único canal de comunicación independientemente de dónde se encuentre ubicado físicamente el departamento, ya que las entidades gestoras poseen departamentos en diferentes edificios y localidades.

Con este sistema de solicitudes de transacciones también se pretende tener un sistema jerárquico en el que no se permita realizar autorizaciones arbitrarias de transacciones, distintas para cada usuario, sino que se garantice que los usuarios con una misma función laboral dispongan de iguales autorizaciones.

En definitiva, mejorar el sistema actual de solicitudes para conseguir una mayor seguridad y eficacia.

2. Aplicación

La aplicación funcionará bajo Windows XP con la tecnología Java, y la interfaz debe ser la misma en todos los centros, permitiendo al usuario interactuar con la aplicación bajo un entorno Web con un navegador de Internet.

Los componentes de la aplicación serían: Un servidor, que estará implantado en el departamento de informática para el gestor de la base de datos, y un servidor de aplicaciones para la información.

Los PCs de los usuarios que estarán implantados en las oficinas de las entidades.

La arquitectura del sistema es un modelo de tres niveles o capas:

- Capa de presentación
- Lógica de negocio
- Acceso a datos.

3. Usuarios del Sistema e identificación de roles

Para poder acceder a la aplicación el usuario tiene que estar previamente autorizado y tener un rol asignado.

Se identificará con su código de usuario y password, siendo posible en este momento el cambio de contraseña, ya que la inicial la asigna el administrador del sistema.

Figura 1.-Página de inicio

En el sistema se distinguirán cuatro tipos de roles para los usuarios con acceso al sistema: Usuario final, Jefe de departamento, Jefe de área y Administrador.

Cabe indicar que los usuarios deben pertenecer a uno y exclusivamente a uno de los roles que a continuación se detallan, describiéndose brevemente esas operaciones, clasificadas por rol:

- Usuario final: Los usuarios pertenecientes a este rol podrán realizar las operaciones de consulta de todos los grupos, y a los que él, como usuario adscrito a un determinado departamento, tiene acceso autorizado.

- Jefe de departamento: Los usuarios pertenecientes a este rol podrán realizar las tareas de alta de solicitudes de transacciones para el personal destinado en su departamento.

- Jefe de Área: Los usuarios pertenecientes a este rol, aceptarán o rechazaran las solicitudes realizadas por los jefes de departamento pertenecientes a su área. Se encargarán de las tareas de alta de solicitudes de transacciones para sus jefes de departamento. También realizarán las peticiones de altas y bajas de usuario para todo el personal de su área.

- Administrador: Los usuarios pertenecientes a este rol son los encargados de validar las solicitudes de transacciones previamente aprobadas por los jefes de área, lo que implica autorizar al funcionario en cuestión al grupo de

transacciones solicitad. Así mismo validará y efectuará el alta o baja de los usuarios indicados por los jefes de área.

4. Modelo de comportamiento del sistema

Sistema de gestión de usuarios

El Jefe de área realiza la solicitud de alta de un usuario cuando éste se incorpora a la entidad, asignándole un área y departamento según su puesto de trabajo.

Cuando un usuario causa baja en la organización, también lo comunica al administrador con una solicitud de baja.

El usuario administrador del sistema que realiza la validación de las solicitudes de alta y baja de usuarios, solicitadas por los diferentes jefes de áreas.

Dicha validación implica un cambio de estado y fecha de la solicitud y en el caso del alta, el alta de un nuevo usuario final en el sistema.

En el caso de baja de un usuario implica el cambio de situación de dicho usuario, pero se mantiene el registro con sus datos, para conservar el historial de accesos a transacciones**.**

Sistema de gestión de solicitudes de transacciones

El usuario final es el funcionario de la entidad asignado a un departamento y para el que su jefe realiza las peticiones de transacciones.

Su Jefe de Departamento que podrá crear solicitudes de transacciones para los usuarios de su departamento, según las funciones que les tenga encomendadas.

El Jefe de Área será el responsable de uno o varios departamentos y encargado de aceptar o rechazar cada una de las solicitudes pendientes realizadas por los jefes de departamento de su área. Fig. 2

Finalmente el Administrador que está a cargo de la administración de la confidencialidad, validará cada una de las solicitudes aprobadas por los jefes de área y dará de alta la petición de transacciones incluida en la solicitud. Fig. 3

Figura 2.- Listado Solicitudes por Área

Figura 3.-
Solicitudes pendientes de validar por el administrador

5.-Despliegue de la aplicación e implementación de posibles mejoras

La aplicación se desplegará en un servidor de aplicaciones con contenedor de ejb's mediante el fichero .ear que se genera al compilar dicha aplicación para su distribución.
También se deberá proceder a la creación de la base de datos, y las correspondientes tablas, para lo cual se deberá ejecutar el script de creación de tablas en el servidor de datos correspondiente.
Se puede acompañar dicha distribución de una copia del manual del usuario, para que se tenga conocimiento del funcionamiento de la aplicación.

Para mejorar la implantación y la seguridad de la aplicación, la misma, al estar usando datos sensibles, debería ser desplegada en un servidor que sirva páginas en modo seguro, para enviar dicha información codificada, para lo cual tendríamos que disponer de un certificado.

6.- Conclusiones

Con esta aplicación se ha pretendido realizar una herramienta de ayuda al administrador de confidencialidad de las entidades gestoras, con el objetivo de mejorar la gestión y estructurar los trámites a seguir para la solicitud de inclusión de un usuario en el sistema informático de la entidad, así como la solicitud de transacciones asociadas a los mismos, en congruencia con las funciones a realizar. Su función sería la de facilitar la realización de dichas labores, así como agilizar la tramitación de las mismas, manteniendo la información disponible a los usuarios del sistema, según los roles asignados a cada uno de ellos.

7.- Referencias

- Tutoriales facilitados por la UAH.
- Piensa en Java. Bruce Eckel
- Aprenda Java como si estuviera en primero.-Universidad de Navarra
- Java Script - Edición Especial Autor: Paul McFedries
  ed.Prentice Hall

- http:.//www.javahispano.org

- HTML con XHTML y CSS autor: Elizabeth Castro Ed. Anaya

Acceso Nacional a SARTIDO

Juan Carlos Ogueta Sáenz

SARTIDO – La Rioja
Instituto Nacional de la Seguridad Social,

Resumen. Detectar un problema es el origen de la solución. Tras describir la situación actual de acceso local al SARTIDO, y apuntar una solución que proporcione el acceso Nacional , se podrán corregir situaciones indeseables, se reducirán los tiempos de gestión y en definitiva se producirá una sustancial mejora en la calidad del servicio que prestamos.

1. Agradecimientos.

A todos y cada uno de los que han hecho posible que este programa formativo se haya podido llevar a cabo (profesores, alumnos, colaboradores, representantes de la administración), incluidos los familiares de estos que con su apoyo y comprensión lo hicieron posible.

Comité asesores funcionales / Legislativos.

D. José Mª. Montón Lana	-Co-Administrador SARTIDO .La Rioja
D. José Luís Montalvo Fdz	Secretario EVI INSS - La Rioja
D. Joaquín Maestra Lorente	Jefe Sec.Control de Pensiones. INSS - La Rioja
D. Javier Arróniz González	Jefe Sec .Jubilación Mte. y Supervivencia. INSS

Comité asesores Técnicos.

D. Juan Ogueta Minguijon.	Universidad de Zaragoza
D. Pedro Jesús Gimeno Benedi	Ministerio de Justicia – Madrid

Unidad Provincial de Informática – La Rioja

Administradores del SARTIDO de Cádiz, Guipúzcoa y Murcia.

Nota

El texto completo de la presente ponencia no se ha incluido en el libro de actas de las Jornadas por tratar temas confidenciales y será objeto de una separata que se entregará a los asistentes durante su presentación oral.

Sistema de Gestión de Permisos de Sartido Aplicación PERSAR

Maria Luisa Osuna Mora[1]. Coro García Arenaza[2]

[1] Instituto Nacional de la Seguridad Social Dirección Provincial de Cádiz
MARIA-LUISA.OSUNA@INSS.seg-social.es

[2] Instituto Nacional de la Seguridad Social Dirección Provincial de Gipúzkoa.
CORO.GARCIA@INSS.seg-social.es

Resumen. Con la realización de este proyecto se pretende poner en práctica los conocimientos adquiridos durante los cursos de formación del Proyecto Ática, impartidos en la Universidad de Alcalá, Especialización en Administración de Sistemas Itinerario BBDD.
El objetivo es diseñar una Base Datos para ser explotada a través del entorno web . De forma general la aplicación PERSAR permitirá gestionar las distintas solicitudes de permisos y tipos de accesos de los usuarios a los expedientes de prestaciones del Instituto Nacional de la Seguridad Social. Éstos deberán ser previamente autorizadas por el Subdirector correspondiente y, una vez supervisadas, se enviarán al responsable de la Seguridad del Fichero para que proceda a la autorización de acceso al fichero a los usuarios con los permisos solicitados.
De esta manera, se podrán consultar las solicitudes y autorizaciones de acceso a los usuarios con los permisos solicitados y disponibles para su consulta, ya que el fichero de Solicitudes de Acceso estará siempre actualizado.

Nota

El texto completo de la presente ponencia no se ha incluido en el libro de actas de las Jornadas por tratar temas confidenciales y será objeto de una separata que se entregará a los asistentes durante su presentación oral.

Gestión del Conocimiento Calidad

Antonio Jesús Pérez Reina[1]

[1] TGSS, Dirección Provincial de Málaga, Unidad Provincial de Informática, 29007 Málaga, España.

Resumen. La tecnología ha dado la clave para realizar toda una serie de procesos que ahora pueden automatizarse o estructurarse y que, por lo tanto, permiten gestionar, algo más, eso que denominamos conocimiento. Lo que gestionamos en realidad, son las condiciones, el entorno y todo lo que hace posible y fomenta dos procesos fundamentales: la creación y la transmisión de conocimiento. Para ello es vital observar e interpretar el funcionamiento de las organizaciones. En este artículo se presenta un sistema donde la información entrante, debido a unos protocolos de actuación, se gestiona para ubicarse como conocimiento de una forma ágil y eficaz, de acceso fácil e intuitivo, en un único repositorio, donde se comparte el conocimiento acumulado Online, lo cual permitirá su actualización y mantenimiento, aprovechando el conocimiento práctico de expertos y especialistas. Dirigiendo la información departamentalmente según su contenido, convirtiendo la información en conocimiento directo y claro, apoyado con la presentación de una Intranet basada en un nuevo gestor de contenidos y construida usando tecnología de la Web 2.0, diseñado para este cometido en Software de libre distribución y manteniendo la compatibilidad con los estándares actuales y una absoluta integración con las Intranet corporativas más importantes e implantadas en las medianas y grandes empresas.

1 Introducción

La Sociedad de la Información (en adelante S.I.) demanda una adaptación del Conocimiento en la Gestión Empresarial, así como en el desarrollo de los servicios y productos ofrecidos. Esta adaptación asume un coste de Recursos Humanos y materiales que debe ser tomado como inversión, demandando la propia empresa una adaptación a sus empleados, ofreciéndoles los recursos necesarios para llevar a cabo el cambio.

En el ámbito de la Calidad se requiere un plus además de los nombrados, como es la transparencia y eficacia, siendo juzgado este proceso con mayor crítica debido a las connotaciones propias, unido a las pautas impuestas por una organización de ámbito estatal en ofrecer un servicio de Calidad a sus clientes. Estos referentes suponen un handicap añadido a la consecución de una mejora y adaptación de la Gestión del Conocimiento, además de los inherentes como son el coste de su implantación y la dedicación de personal a estos fines.

La adaptación debe entenderse como la transferencia del conocimiento y experiencia existente en las personas y la transformación del conocimiento tácito en conocimiento

explicito, de manera que se encuentre documentado y almacenado para que cualquiera pueda hacer uso del mismo cuando sea necesario.

En este artículo se tratará de mostrar las carencias encontradas en el ámbito de la Gestión del Conocimiento y el esbozo del proceso propuesto que el riguroso trabajo de campo y posterior análisis de la situación actual han ayudado a dibujar [1]. Solo partiendo de la necesidad llegaremos a una óptima solución que nace de la fusión de los dos activos más importantes que tiene una empresa, los datos y las personas. Se ha trabajado en buscar una mejora, un sistema que aporte más y mejores servicios con menos recursos, a través de la optimización de la información, atendiendo a la demanda social. La administración actual no puede soportar problemas del siglo XXI sostenidas en una organización del siglo XIX.

Se utilizará el modelo EFQM integrado en el R.D. 951/2005 por el que se establece el marco general para la mejora de la calidad en la Administración General del Estado [2] [3]. Este modelo nos ofrecerá una coherencia y visión integrada a la gestión, tomando como base la integración de las tecnologías de la información [4]. A partir de aquí se marcará un lenguaje común interna y externamente, que será utilizado para conocer el estado actual de nuestra gestión y como guía inspiradora en nuestra mejora continua [5]. El objetivo es desarrollar un sistema que garantice la Gestión del Conocimiento respetando la orientación hacia el cliente [6], el desarrollo e implicación de las personas y la gestión por procesos y hechos.

2 Análisis Previo

Una definición clara y concisa sobre la Gestión del Conocimiento quizás no existe, pero podemos aproximarnos diciendo que consiste en optimizar la utilización de este recurso mediante la creación de las condiciones necesarias para que los flujos de conocimiento circulen mejor [7].

2.1 ¿Por qué ahora, la gestión del conocimiento?

Por diversas razones fundamentales; la primera es por la denominada nueva economía, economía del conocimiento o de la información, conceptos que son progresivamente más importantes, como recurso y también como producto. La preocupación por este aspecto hace que se plantee la necesidad de que el capital en forma de conocimiento que posee la organización, se quede dentro circulando entre todos sus miembros [8].

Otro elemento que ha tenido una importancia vital es el hecho que las nuevas tecnologías han aportado toda una serie de herramientas y metodologías que permiten desarrollar acciones relacionadas con el conocimiento que antes no podían llevarse a cabo.

3 Deficiencias actuales y estudio de la Gestión del Conocimiento

Actualmente las empresas y administraciones gestionan en menor o mayor medida la información y los datos, sin llegar a gestionar lo que llamamos conocimiento. Para lo cual en su primera fase se ha identificado el problema y se han definido las posibles causas [5] de la dispersión del conocimiento:

- Acelerado cambio de la normativa.
- Criterios aleatorios de clasificación y acceso a la información.
- Desconocimiento de las vías de acceso a la información.
- Insuficiente conocimiento de la forma de circulación.

3.1 Identificación del problema

Los trabajos realizados para definir las carencias de los procesos seguidos en la distribución y acceso a la información en el presente proyecto, tomando como ámbito de aplicación de estudio una empresa mediana-grande de ámbito provincial [9] se hacen partiendo de los siguientes campos de trabajo:

- Estudios de resultados del EFQM [9].
- Encuestas verbales a los responsables de los departamentos [1].
- Cuestionario de valoración dirigido a todos los usuarios [1] [10].

3.1.1 Estudio de resultados del EFQM

De acuerdo con las conclusiones contenidas en el Modelo EFQM las organizaciones excelentes, deben desarrollar y mantener los niveles de conocimiento y capacidad de las personas de la organización. Para ello han de identificar los medios que son necesarios para canalizar la información de y hacia las personas que la integran y de y hacia los usuarios externos de la organización (clientes o administrados).

Los escasos canales de información establecidos no garantizan que la información fluya en sentido ascendente y descendente. Se ignora o no se utilizan, se producen interrupciones, sobretodo, en los tramos intermedios.

En ocasiones la falta de comunicación de los cambios a todos los grupos de interés pertinentes, produce que el gestor interno se informe del cambio o de las modificaciones legislativas por los clientes [11].

Se debe potenciar la gestión del conocimiento en la organización creando unidades de investigación y desarrollo en materia de organización del trabajo con el fin de sistematizar la gestión a nivel corporativo. Además, se deben crear y fomentar los canales oportunos para recoger el conocimiento de todos los individuos y facilitar su acceso [12] [13].

3.1.2 Encuestas verbales a los responsables de los departamentos

Las encuestas se han realizado de forma no estructurada a lo largo de todo el tiempo de trabajo correspondiente a la fase de identificación del problema. Las res-

puestas obtenidas han coincidido en las referencias hechas a los puntos débiles del sistema actual [1].

3.1.3 Cuestionario de valoración dirigido a todos los usuarios.

Con el fin de conocer la opinión basada en la experiencia de todo el personal de la organización, y utilizar los estudios resultantes más allá de lo marcado en una comparativa de benchmarking se envió un cuestionario con preguntas relativas a la calidad, acceso y transmisión de la información. Se diseñó para detectar los puntos donde se encuentran las deficiencias de información (ausencia, redundancia y calidad), de acceso y de transmisión de la información a clientes internos y externos. Remitido en formato HTML, y dividido en tres bloques, cuyos resultados manifiestan con claridad las deficiencias actuales y la apuesta mayoritaria por la utilización de las tecnologías de la información y el uso de un único canal que integre todo el conocimiento, destacando los resultados a tres de las veinte preguntas del cuestionario.

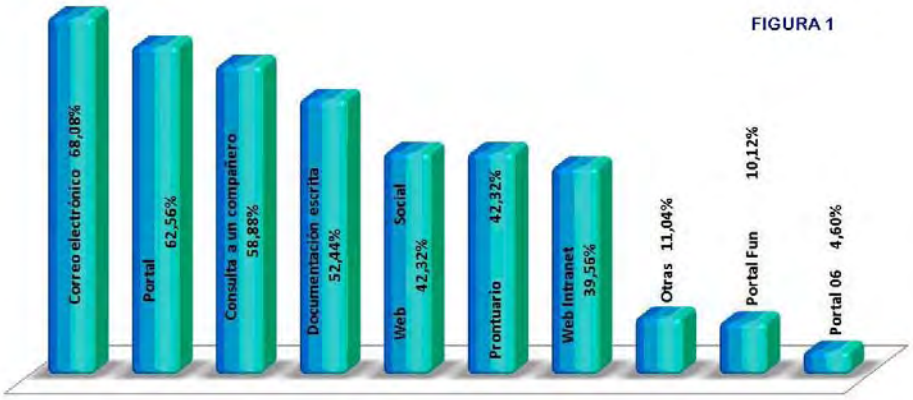


Fig. 1. Entorno al 40% de los consultados utilizan hasta siete canales distintos de acceso a la información.

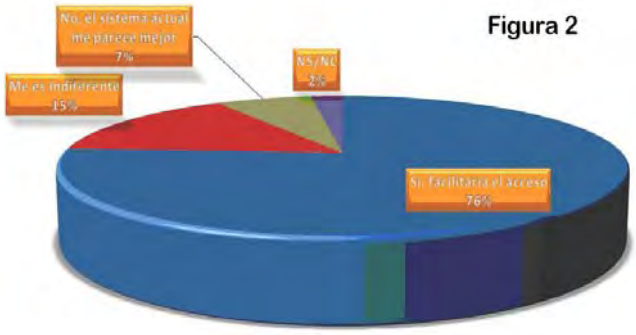


Fig. 2. Un 76% prefieren canalizar a través de un único sitio el acceso a toda la información, destacando que un 93% apuesta por el formato digital.

3.2 Descripción del problema y definición de la acción

La concreción del problema y la definición de las acciones a desarrollar han sido objeto de un amplio debate entre los responsables de los departamentos y el elaborador del presente proyecto. Un sistema que recoge todo tipo de datos e información y lo transforma en conocimiento a través de un aprendizaje continuo en la recopilación, almacenaje y acceso de estos datos y del conocimiento inherente a las personas que forman la organización. Un sistema que acaba con las interpretaciones de forma genérica y masiva, ofreciendo un sistema homogéneo y eficaz, utilizando para ello el conocimiento aplicado unívocamente por el responsable.

El sistema recoge como soporte la propuesta de una Intranet Provincial [1] versátil y en concordancia de compatibilidad con las Intranets más implantadas por las empresas en su ámbito Nacional, la cual pueda accederse a través de la misma vía y con la misma facilidad, a un documento conciso, a modo de guión o lista de instrucciones concretas, que le permite en el menor tiempo posible gestionar un caso y sus variantes. Se define un método perfectamente estructurado de creación y publicación de protocolos de actuación, que aprovecha el conocimiento práctico de expertos y especialistas. Así, la acción a emprender queda definida en función de los siguientes criterios:

- Acceso fácil e intuitivo y "Online" a las fuentes del conocimiento.
- Disponer de una fuente única y sistematizada para la gestión.
- Compartir y facilitar el conocimiento acumulado.
- Distribución adecuada del conocimiento.

3.3 Metodología

A partir del estudio de los archivos existentes, analizando los formatos, relevancia y accesos, se realiza la división según su acceso (intranet corporativa, Boletines oficiales, portales fun y 06 (portales gestionadas por la central de la empresa implantada como soporte a sus provincias), prontuario faq, notas y circulares interiores, etc.) y según la materia (normativa, instrucciones, circulares, jurisprudencia, dictámenes, modelaje, etc.). Dado los mecanismos a través de los cuales llegan a los usuarios las modificaciones normativas y de gestión, tanto internos (personal de la empresa) como externos (clientes), nos exige distribuir la presente exposición en tres áreas:

- Archivos de Consulta
- Normativa de Gestión (Generales, específicas, internas y restringidas)
- Consultas Externas

4 Proceso y Procedimiento

Se parte de dos procesos fundamentales inherentes a la gestión del conocimiento: la creación y la transmisión de conocimiento.

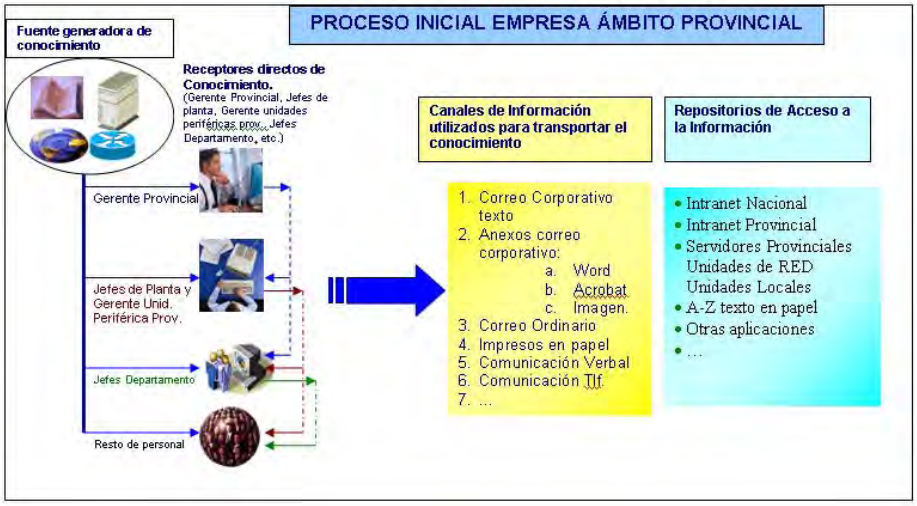


Fig. 3. Gráfico del flujo actual de la información donde se muestran las deficiencias y marca el punto de partida de trabajo del presente proyecto.

4.1 Aspectos del proceso a mejorar

Los análisis previos han detectado deficiencias que serán objeto de las mejoras propuestas, entre las que destacan; la diversidad de las fuentes de entrada y formatos, inexistencia de roles para la gestión del proceso, ambigüedad y diversidad de formatos, clasificación y asignación duplicada, inexistencia de patrón de actuación en su reparto, distribución temporal excesiva, disponibilidad limitada, diversidad de herramientas para su acceso, falta de accesibilidad, canalizar la información por grupos de destinatarios poco coherentes, así como Intranet opacas.

4.2 Propuestas de mejora del proceso

Se proponen las siguientes mejoras, perfectamente definidas en subprocesos, para corregir las deficiencias detectadas:

- ✓ Detección de todas las fuentes de entrada de conocimiento.
- ✓ Creación de roles para la gestión del proceso: El Elaborador y el Supervisor, ambos tramitadores del conocimiento.
- ✓ Digitalizar la información en un mismo formato.
- ✓ Clasificar y asignar la información entrante a departamentos.
- ✓ Distribución temporal, accesibilidad y máxima disponibilidad..
- ✓ Adaptar, resumir y descartar la información entrante y su formato.
- ✓ Mantenimiento de destacados, novedades e históricos.

4.3 Proceso propuesto

Se propone un sistema que ofrece un flujo automatizado y ordenado de entrada del conocimiento, corrige las deficiencias del actual sistema, resultando un modelo transparente en todas sus fases, donde la información entrante, debido a unos protocolos de actuación, se gestiona para ubicarse como conocimiento de una forma ágil y eficaz mediante los mismos canales de información, en un único repositorio, lo cual permitirá su rápido acceso, actualización y mantenimiento, favoreciendo un aspecto crítico como es la disponibilidad.

La información se dirige por departamentos y a los demandantes directos según su contenido, se convierte la información en conocimiento conciso y claro. Este planteamiento es flexible, según las necesidades departamentales pueden asignarse los roles de Supervisor y Elaborador, siendo únicos o plurales. Se incorpora un sistema de consulta a expertos Online, utilizando y adaptando para ello la tecnología ya implementada y diseñando nuevas herramientas para este fin.

Fig. 4. El proceso propuesto está formado por ocho subprocesos debidamente especificados, dos tipos de agentes y una intranet de soporte.

El subproceso Publicar explica la esencia del presente trabajo, es el de menor complejidad y el de mayor importancia de todos, de este depende lo que en su inicio, fueron datos, pasó a tratarse como información y una vez publicado se convertirá en conocimiento. Resume la esencia de la Gestión del Conocimiento donde la calidad es responsabilidad de todos en especial de los niveles de dirección, por esto el propietario del subproceso propuesto es el supervisor.

La Intranet Provincial propuesta y creada para el presente proyecto encapsula un nuevo sistema de Gestión de Contenidos, construido usando tecnología de la Web 2.0 [16], utilizando para ello herramientas Software de libre distribución, lo cual incide en un coste cero y asegura compatibilidad con las herramientas actuales.

El trabajo realizado presenta como anexo la entrega de una Intranet como repositorio de conocimiento dinámico, la cual implementa el estado de la información mediante semáforos. Estos iconos luminosos indican el estado en el cual se encuentra la publicación, informando en todo momento del estado de gestión antes de su volcado completo, evitando retraso en su publicación.

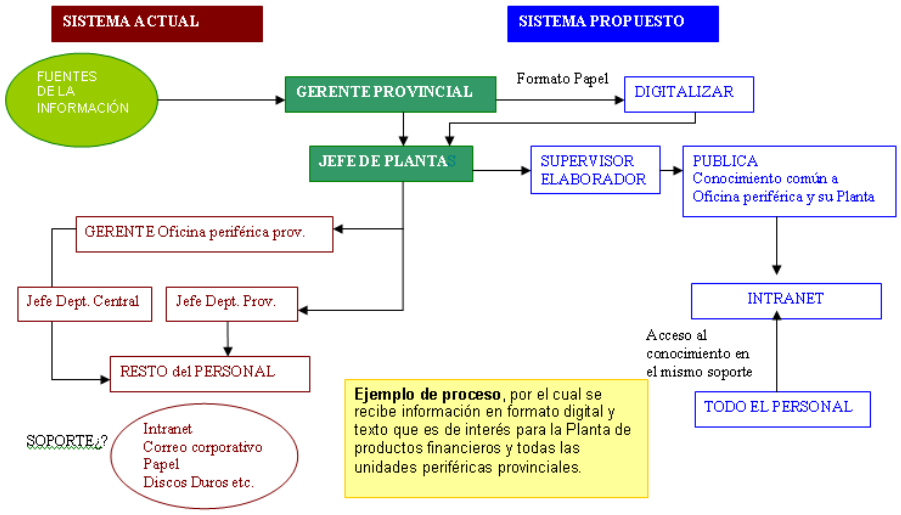


Fig. 5. Diagrama que muestra las mejoras del proceso digitalizar.

5 Viabilidad del proceso

En base a un exhaustivo análisis de costes y recursos humanos, se descarta la creación de un grupo de trabajo específico y común. A tenor de las encuestas realizados y una vez estudiados los datos, se concluye que la creación de perfiles Elaborador y Supervisor en cada Planta y Unidad Periférica es la óptima, únicamente los puestos asignados de cada departamento tienen los conocimientos necesarios para catalogar, resumir, descartar, mantener, priorizar y transformar la información entrante.

6 Conclusiones

El proceso presentado evita engrosar la lista de productos 'VaporWare'. Para ello los distintos análisis realizados en la elaboración inciden en respetar el estándar de facto 'Hyper Curve' [15] de la consultora Garther.

A raíz de las deficiencias de gestión que son susceptibles de mejora, se ha creado un proceso que recopila el conocimiento disperso y facilita el acceso al mismo, resaltando la gestión del conocimiento como base en la comunicación interna y externa cuya incidencia es primordial en la organización. A tal fin se logra gestionar dos procesos fundamentales inherentes al conocimiento: la creación y la transmisión de dicho conocimiento.

Se establecerán las garantías suficientes para asegurar el correcto tratamiento de los datos, según lo dispuesto en la LOPD, en lo que se refiere a su contexto, seguridad física y lógica, facilitando la auditoria de los datos tratados por la entidad, así como su acceso.

La culminación de este proyecto produce un proceso que transforma la información en conocimiento, captando el conocimiento implícito y explícito. Ofrece una vía única donde se almacena para su posterior acceso, y mejora la comunicación interna y externa de la organización, en definitiva el bien no tangible más preciado "el conocimiento".

Referencias

1. Grupo de Mejora de la Calidad Gestión del Conocimiento y Propuesta de Mejora de la Calidad Intranet desarrolladas en la empresa.
2. Real Decreto 951/2005 por el que se establece el marco general para la mejora de la calidad en la AGE.
3. Ley 30/1992 de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común (Principios Generales).
4. Sistemas de Gestión de la calidad (UNE-EN ISO 9001:2000)
5. Documento sobre directrices del Plan de Calidad, desarrollo y normas de implantación.
6. Ley 6/1997 de Organización y funcionamiento de la Administración General del Estado (Mejora de los servicios públicos atendiendo a las demandas de los ciudadanos).
7. La gestión del coneixement. Ed. FUOC, 2003. Agustí Canals Parera.
8. Universidad Carlos III de Madrid (Grupo de Mejora de Calidad – Atención e Incidencias Telefónicas).
9. Empresa de Málaga (Primera Auto evaluación EFQM).
10. Modelo EFQM de Excelencia. Informe Final (Auto evaluación por cuestionario)
11. Guía de mejoras para la Administración Pública. Modelo EFQM de Excelencia. Cuarta Edición.
12. El Esquema Lógico REDER
13. El libro blanco para la mejora de los servicios públicos. Ministerio de Administraciones Públicas.
14. Proyecto Intranet Provincial presentado por Antonio Jesús Pérez Reina como propuesta de mejora dentro del entorno EFQM en 2008.
15. Material diverso OPENCOURSEVARE “Material educativo y docente libre y gratuito”.

16. Tim O'Reilly «What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software» en el Portal de la Sociedad de la Información de Telefónica.

bd^{3}p. Base de Datos Documental para Direcciones Provinciales

Ángel Punzano Martínez

Dirección Provincial de la T.G.S.S. en Barcelona

Resumen. Realización de una aplicación web para el mantenimiento y utilización de una base de datos documental para las direcciones provinciales de las distintas Entidades Gestoras de la Seguridad Social. Los diferentes usuarios, en función de su perfil y mediante un navegador web podrán introducir y consultar la localización de los documentos a los que tienen acceso, de aquellos que circulan por su dirección provincial.

1. Introducción

Actualmente, en las direcciones provinciales de las Entidades Gestoras de la Seguridad Social se recibe, genera y envía una gran cantidad de documentos, principalmente escritos aunque en función del trabajo desarrollado en cada departamento podemos encontrar otros tipos: libros, planos, revistas, recortes de prensa, fotografías, etc. Esto representa un gran volumen de información que debe ser localizable de forma inmediata y en cualquier momento para que resulte realmente útil.
La herramienta que aquí se presenta pretende facilitar el control de toda esta documentación de forma sencilla. Permitirá a los trabajadores autorizados en cada dirección provincial introducir documentos en el sistema así como la posterior consulta de su ubicación, ya sea física (almacén, archivo, caja, etc.) o lógica, si se trata de documentos que se encuentran en la propia red informática (archivos de texto, fotografías digitalizadas, etc.)

2. Objetivo

El objetivo final de este proyecto es realizar una aplicación que permita localizar cualquier documento que haya circulado por la dirección provincial (y que haya sido introducido en el sistema) de forma rápida y eficaz. La aplicación permitirá realizar tanto búsquedas exactas por cualquiera de los campos introducidos al dar de alta un documento en el sistema como, y este es el objetivo primordial, búsquedas aproximadas sobre el texto incluido en el título o la descripción del documento.

3. Descripción de la aplicación

Para conseguir el objetivo fijado, la aplicación permitirá introducir en la base de datos documentos, buscarlos y editarlos. Además proporcionará tareas de mantenimiento de tablas auxiliares: altas, consultas, edición y bajas de usuarios, secciones, subdirecciones, tipos de documentos disponibles y palabras vacías.

3.1 Visibilidad

Para el desarrollo del sistema consideraremos una dirección provincial dividida en subdirecciones y éstas a su vez, en secciones. Un usuario sólo podrá pertenecer a una sección y por tanto a una subdirección.

Figura 1. Estructura simplificada de una Dirección Provincial.

En cuanto a la visibilidad, los documentos almacenados en nuestra base de datos podrán ser de dos tipos:

- Documentos públicos. Serán accesibles para cualquier usuario autorizado del sistema.
- Documentos privados. Serán accesibles únicamente a los usuarios pertenecientes a la misma sección de aquél que introdujo el documento en la base de datos, con dos excepciones: un usuario de tipo supervisor (subdirector) podrá acceder a todos los documentos de las secciones que dependan de su

subdirección, además tanto el director provincial como el administrador del sistema tendrán acceso a todos los documentos.

3.2 Descriptores y palabras vacías

En el ámbito de nuestra aplicación, consideraremos una palabra vacía (stop word) cualquier término incluido en el título o la descripción de un documento que no aporta información útil sobre el contenido de dicho documento. Son ejemplos habituales de palabras vacías los artículos, conjunciones, preposiciones, etc. Por el contrario, llamaremos descriptor a aquellas palabras incluidas en el título o la descripción de un documento que guardan relación con su contenido y que, por tanto, nos permitirán a posteriori localizarlo aunque no conozcamos exactamente ni su título ni la descripción que almacenamos inicialmente.

Figura 2. Ejemplo de selección de descriptores y palabras vacías en un título.

El sistema mantendrá una tabla de palabras vacías y una de descriptores. El mantenimiento de la primera será manual y a cargo del administrador. La segunda estará inicialmente vacía e irá creciendo a medida que se introducen documentos en la base de datos.

3.3 Usuarios y funcionalidades

En el sistema existen cuatro tipos de usuarios: empleado, supervisor, director y administrador. Sin embargo, sólo habrá dos tipos de funcionalidades: gestor y administrador.

Los usuarios de tipo empleado, supervisor y director tienen acceso a las funcionalidades de gestor, lo que les permitirá añadir, consultar, modificar y eliminar documentos de la base de datos. Por tanto, la diferencia entre estos tres tipos está en la visibilidad, no en la funcionalidad.

Los usuarios de tipo administrador también tienen acceso a las funcionalidades de gestor pero además se ocuparán del mantenimiento de las tablas de tipo de documento, palabras vacías, usuarios, secciones y subdirecciones.

3.4 Funcionamiento de las búsquedas aproximadas

Como hemos mencionado anteriormente, la funcionalidad principal de la aplicación es permitir realizar búsquedas aproximadas de documentos. Siguiendo con el ejemplo de la figura 1, imaginemos que después de un tiempo queremos recuperar "aquél escrito que recibimos que hablaba sobre horarios pero que no recordamos exactamente como se titulaba". En este caso lo que haremos es buscar, por ejemplo "instrucciones sobre horarios" entonces el sistema buscará los términos "instrucciones", "sobre" y "horarios" y nos ofrecerá un listado de todos los escritos que contienen como mínimo uno de los términos (en el título o la descripción del mismo) ordenados según el porcentaje de coincidencia, es decir consideramos que a más términos coincidentes más probabilidad existe de que se trate del documento buscado.

4. Implementación del sistema

Decidimos implementar el sistema en forma de aplicación web ya que esto nos proporcionaba diferentes ventajas:

- No es necesario instalar la aplicación en cada ordenador que vaya a acceder al sistema.
- Podemos diseñar una interfaz "tipo internet" a la que la inmensa mayoría de personas está habituada hoy en día.
- Diferentes tipos de ordenadores con diferentes configuraciones podrán acceder a la aplicación sin problemas.

4.1 Descomposición en subsistemas

A partir de las dos funcionalidades detectadas: gestor y administrador, podemos descomponer el sistema en tres subsistemas:

- Subsistema de validación. Será el encargado del control de acceso a la aplicación.
- Subsistema de gestión. Incluye todo lo referente al propósito de la aplicación, es decir, al trabajo con documentos: altas, edición y búsquedas.
- Subsistema de administración. Incluye todas las tareas auxiliares para el correcto funcionamiento del sistema: mantenimiento de tablas de usuarios, secciones, etc.

Aunque los subsistemas de gestión y administración son independientes entre sí, como los usuarios de tipo administrador también poseen funcionalidades de gestor, accederán a ambos desde un único punto de la aplicación.

4.2 Modelo de estructura de objetos

Las clases de negocio que intervienen en el sistema son: Usuario, TipoUsuario, Subdireccion, Seccion, Documento, TipoDocumento, Descriptor, Indice y Palabra-Vacia.

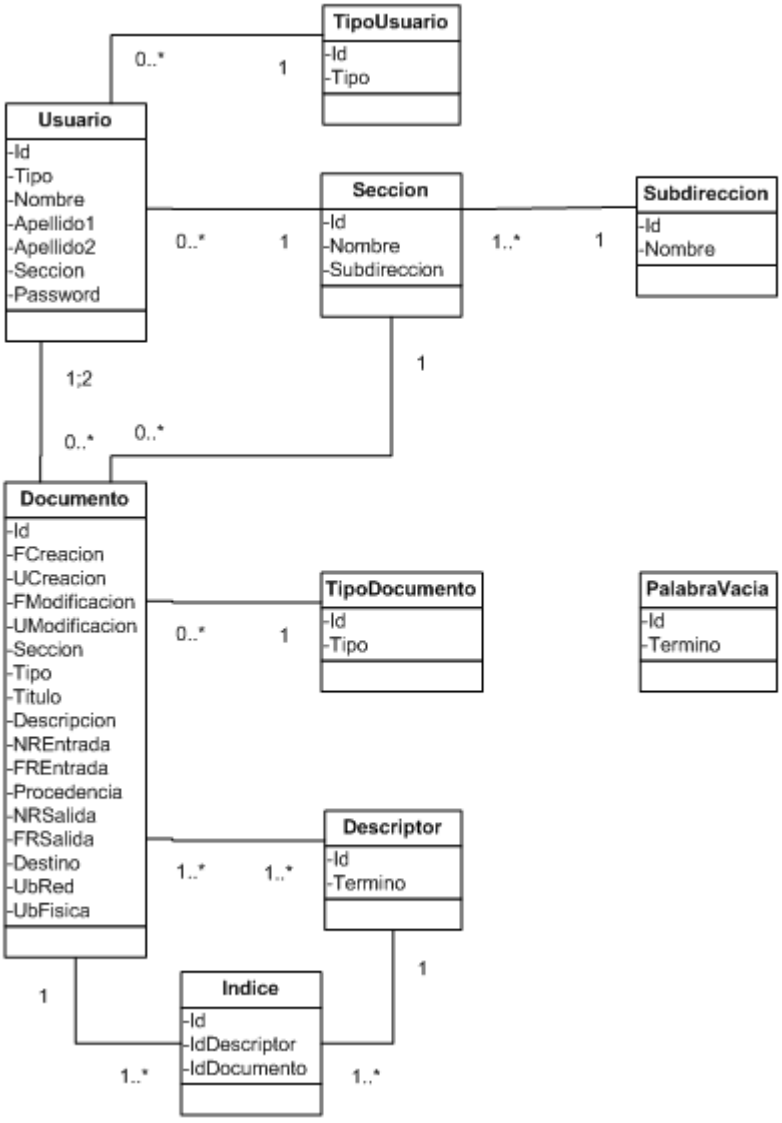


Figura 3. Diagrama de clases de negocio.

4.3 Interfaz del sistema

La aplicación está dotada de una interfaz intuitiva y de fácil uso. El diseño de las pantallas en forma de página web facilitará la aceptación por parte del usuario, ya que con toda probabilidad éste estará habituado a la navegación por internet.

De forma general, las pantallas estarán divididas en cuatro zonas principales siguiendo el esquema de la figura.

Figura 4. Distribución general de una pantalla estándar de la aplicación.

En el área de *Título de la aplicación* aparece el logotipo de la misma y su nombre.

En el *Área de información* se muestra el nombre del usuario que está conectado y el lugar del sitio dónde nos encontramos (migas).

En el *Área de menús* se ofrecen al usuario las diferentes opciones disponibles para su perfil.

Finalmente, en el *Área de trabajo* se van mostrando en cada momento los formularios necesarios para realizar las diferentes tareas disponibles en la aplicación.

Además, en la parte inferior de la pantalla se muestran siempre un enlace que permite cerrar la sesión en curso y volver a la pantalla de acceso y otro que muestra la pantalla *Acerca de* con información sobre la aplicación.

En las figuras siguientes podemos ver dos ejemplos de pantallas reales del sistema.

Figura 5. Pantalla *Consulta de documentos por campos*.

Figura 6. Pantalla *Listado de documentos encontrados*.

4.4 Arquitectura del sistema

La aplicación se ha desarrollado siguiendo un modelo de arquitectura de tres capas: presentación, lógica de negocio y acceso a datos. Así, los componentes del sistema quedan divididos en las diferentes capas según pertenezcan a la interfaz de usuario, aporten lógica de negocio o faciliten el acceso a los datos.

5. Referencias

5.1 Bibliografía

- Enciclopedia de Microsoft Visual C#. Fco. Javier Ceballos. Editorial Ra-ma.
- Introducción a la OOP. Francisco Morero. Grupo Eidos.
- Desarrollo de aplicaciones para internet con ASP.NET. Ángel Esteban. Grupo Eidos.
- Introducción a CSS. Javier Eguiluz Pérez. LibrosWeb.es.

5.2 Documentación

- Apuntes y material del curso Desarrollo de aplicaciones Web con .NET. Plan ATICA. Universidad de Alcalá de Henares.
- Metodología METRICA versión 3. Ministerio de Administraciones Públicas.

5.3 Páginas web consultadas

- www.netveloper.com
- http://blogdediegoramirez.blogspot.com (The DAR Side of IT)
- http://trytocatch.wordpress.com
- http://elguille.info
- http://eloyparedes.wordpress.com (El lado oscuro de la fuerza)
- http://www.asp.net
- http://msdn.microsoft.com

Gestión de los documentos archivados

Esteban Sebastián Marco

Unidad Provincial de Informática de Zaragoza
Gerencia de Informática de la Seguridad Social

Resumen.- En tanto no se llega a la oficina sin papeles los diferentes departamentos de cualquier centro generan importante cantidad de documentación que primero queda en el puesto de trabajo, luego en pequeños almacenes de planta, posteriormente en el almacén del edificio y finalmente en almacenes situados en edificios ajenos.
En el caso que se presenta, la dirección de una entidad administrativa y el personal encargado del control del archivo presentaron la iniciativa de construir una aplicación informática que les diera acceso al mismo desde cualquier puesto de trabajo sin que esta actuación supusiera una alteración en el proceso diario.
En este artículo se detallan los pasos realizados para dotar a esa entidad de un sistema informático que le permita el control de todo su material archivado, teniendo en cuenta que la citada entidad cuenta con diferentes centros ubicados en distintos edificios.
En primer lugar se describe la situación actual, solución que se propone, entorno tecnológico, codificación y tipos de usuarios. Posteriormente se analizan los requisitos, las especificaciones de la interfaz, los modelos de casos de uso, el modelo de datos y el análisis de consistencia, para el posterior diseño del modelo de clases, el físico de datos y el desarrollo de casos de uso reales, para finalizar con las pruebas del sistema y manuales de explotación y usuarios[1].

1. Introducción

En la definición del sistema se hace un breve análisis de la situación actual, se propone un solución en la que se detalla el entorno tecnológico en el que se va a desarrollar la aplicación del tipo cliente servidor sobre una red corporativa que comprende diversas redes locales interconectadas. Se continúa con la codificación requerida y se determina el tipo de usuarios.
Se exponen los requisitos que la aplicación debe cumplir para satisfacer las necesidades de los usuarios, como los datos a almacenar, procesos a realizar, como se debe presentar a los interesados y que medidas de seguridad debe reunir. Se detallan las especificaciones de la interfaz para que sea lo más amigable posible. Se diseñan casos de uso que corresponden con los procesos enumerados en los requisitos funcionales y de seguridad y un modelo de datos correspondiente a los requisitos de datos. El análisis de consistencia debe verificar esa correspondencia y que los requisitos de la interfaz cumplan las especificaciones de la misma.
A continuación en el proceso de diseño, se muestran todas las clases persistentes de la base de datos y sus relaciones y se desarrollan los casos de uso en diagramas de secuencia para permitir la planificación de la programación de los mismos. Un diagrama entidad-relación de los datos a partir del modelo de análisis, permite la creación de la base de datos. Se expone la necesidad de la fase de pruebas tanto por parte de los usuarios como por los desarrolladores de la aplicación, así como la elaboración de manuales para el personal encargado de su explotación y para el personal al que va destinada, para concluir con las ventajas e inconvenientes que presenta la ejecución del proyecto.

2. Análisis del Sistema.

2.1 Definición.

2.1.1. Situación actual y solución propuesta

En el caso que se presenta, no existe un registro informático del contenido del archivo, por lo que se produce una falta de conocimiento de éste en su conjunto pero no de forma particular ya que siempre hay alguien que conoce la ubicación exacta o aproximada del documento buscado en base a la documentación escrita y a la experiencia.
Los elementos que componen este archivo son documentos de carácter interno, documentos enviados a otras unidades administrativas propias o ajenas que pasan por el registro de salida y finalmente documentos recibidos del exterior a través del registro de entrada. Todos ellos una vez tramitados se envían dentro de sus correspondientes archivadores a los diferentes almacenes de la entidad.
La situación es favorable a su informatización, pues sólo se necesita la creación de un registro de todos los documentos almacenados, recogiendo los datos de los registros de entrada y salida e introduciendo los datos de los documentos de carácter interno y la creación de otro registro de los archivadores que contendrán esos documentos así como su ubicación en un almacén. La grabación de los datos de los documentos deberá efectuarse en el momento de su inclusión en un archivador, la de los archivadores en el de su apertura.
A esta información habrá que añadir las tablas que definirán la clasificación del archivo (centros, departamentos, temas, tipo de archivador, tipo de documento, almacenes y usuarios) estando la mayor parte de estos datos en los ficheros de la entidad.
El resultado debe ser que en todo momento un usuario pueda saber, de acuerdo con el grado de confidencialidad que se le asigne, en qué archivador está un documento, en que almacén está ese archivador, que archivadores hay en cada almacén, que documentos contienen esos archivadores y que documentos han sido retirados para consulta por un usuario, así como la disponibilidad de herramientas que le permitan la búsqueda de documentos con datos parciales.

2.1.2. Entorno tecnológico.

Ahora debe resolverse como hacer que la información esté accesible a todos los centros de la entidad, dado que si bien dichos centros forman parte de una red corporativa, están en diferentes edificios y utilizan diferentes servidores.

Fig. 1. Esquema de la red existente para nuestra entidad. En la oficina central se incluye el servidor de aplicaciones y de base de datos que atenderá las peticiones de los usuarios de los diferentes centros que componen la misma.

Una aplicación del tipo escritorio multiusuario que es la que habitualmente se usa en cada departamento no parece útil para este caso por lo que se pensó en una aplicación cliente servidor instalada en un PC con las suficientes prestaciones para admitir un servidor de aplicaciones y un servidor de base de datos (figura 1).

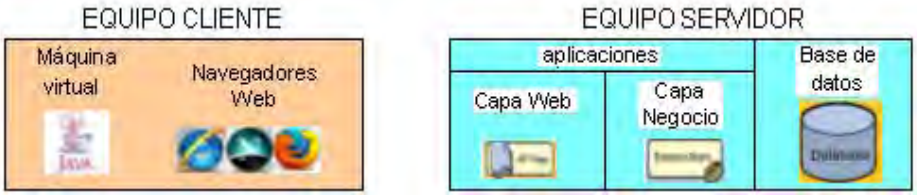


Fig. 2. Se simplifica el software del equipo cliente a máquina virtual java y navegador Web que en este caso será Internet Explorer. En el equipo servidor, una aplicación Java EE y una base de datos derby, deberán satisfacer las peticiones del cliente.

En esta aplicación se deja del lado cliente solamente el navegador y la máquina java, mientas que la carga de trabajo se le adjudica al servidor de aplicaciones [3] que por una parte atenderá las peticiones y devolución de información y de otra atenderá las actuaciones sobre la base de datos (figura 2). Esto permitirá al personal informático que mantiene la aplicación efectuar las actualizaciones oportunas de forma centralizada sin necesidad de interferir en los puestos de trabajo.

2.1.3. Codificación.

La codificación permitirá la identificación de cada elemento de las tablas impidiendo su duplicidad y facilitando la búsqueda de los mismos.

- Tablas: número de orden de creación (excepto en la de usuarios). Será transparente para el usuario que sólo debe ver el nombre del elemento de la tabla. Una vez creado un elemento de la tabla no se borrará ni se cambiará su significado sólo se le dará de baja o nuevamente de alta.
- Documentos: Propios: ejercicio - número de orden de grabación - P; Entrada: ejercicio - número del registro de entrada - E; Salida: ejercicio - número del registro de salida - S. (figura 3)
- Archivadores: ejercicio - número de orden grabación archivadores independientes - número anexo. (figura 3)

Fig. 3. Para facilitar el manejo de los códigos al usuario, se eliminan los ceros a la izquierda y se separan ejercicio, número, clase de documento o anexo archivador por barra inclinada.

2.1.4. Perfiles.

La aplicación tendrá tres perfiles de usuarios: usuario común que realizará todas las tareas de grabación, consulta modificación de los elementos del archivo, usuario que además de las tareas anteriores podrá actualizar las tablas codificadas (tipos de documentos, material, temas, departamentos, centros, almacenes, usuarios) y finalmente usuario del departamento de informática que se encargará de las tareas de instalación, mantenimiento y explotación de la aplicación informática y tendrá acceso a todas las opciones.

2.2. Requisitos.

Se establecen los requisitos que ha de cumplir la aplicación para satisfacer las necesidades de los usuarios en base a los datos que debe almacenar, las funciones que debe realizar, los criterios que deben regular la interfaz y las medidas de seguridad.

2.2.1. De datos.

Requisitos de datos del sistema definidos de forma que puedan utilizarse como modelo de la base de datos. Estos datos aparecerán en las entidades de la aplicación.
Los elementos físicos de nuestro archivo son los documentos y los objetos que los contienen que son los archivadores, almacenes y opcionalmente usuarios (figura 4).

Fig.4. Los tipos de documentos y material archivador se definen en la tabla tipo de documentos y tipo material, los almacenes y usuarios en las tablas del mismo nombre.

En el caso de los documentos se guardará el código del documento, su fecha, breve descripción del contenido, tipo de documento, código del archivador que lo contiene (ocasionalmente puede ser el código de un usuario), nombre de la unidad que lo emite (documento propio o de entrada) o de la unidad de destino (documento de salida), código del usuario que realiza la anotación y estado del documento, válido o anulado.
En el caso de los archivadores se registrará el código del archivador, breve descripción de su contenido, tipo físico del archivador, código del contenedor de este archivador que puede ser otro archivador mayor, un almacén u ocasionalmente un usuario, tema al que corresponde el contenido del archivador, código del departamento y centro al que pertenece, código usuario que realiza la anotación y estado del archivador, válido o anulado.
Especial importancia es la definición del archivo, es decir, los criterios que determinarán la clasificación y búsqueda de los elementos del archivo (figura 5). Para ello se deben establecer una serie de tablas que definan que tipo de documentos va a tener el archivo, que clase de archivadores, los temas que tratará, los centros y departamentos que abarca, así como los almacenes de que se dispone y los usuarios autorizados.

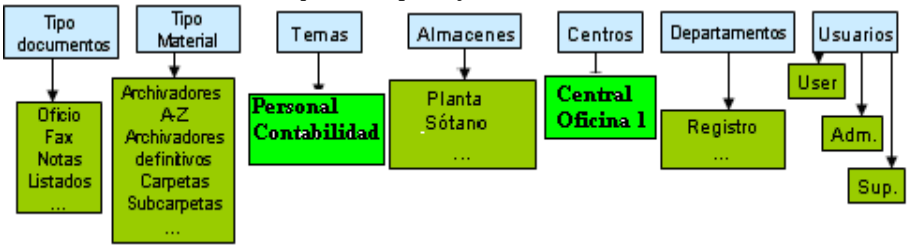


Fig. 5. Tablas que definen el archivo tipo documentos, material, temas, almacenes, centros, departamentos y usuarios.

Todas estas tablas dispondrán de los campos código, nombre (ubicación en el caso de almacenes, dni, nombre y apellidos en el caso de usuarios), fecha de alta, fecha de baja y código usuario que realiza la transacción. La tabla de usuarios tendrá además el nivel de acceso que se le designe.

2.2.2 Funcionales.

La aplicación deberá permitir al usuario la ejecución de todos los procesos necesarios para controlar el archivo que serán altas, consultas, modificaciones y bajas en todas las tablas que componen este archivo. Todos estos procesos se reflejaran en casos de uso y se detallarán en diagramas de secuencia.
En los documentos se distinguirá el proceso de alta entre documentos propios y los de entrada o salida (figura 6) por la diferencia de datos a introducir ya que en los primeros no hay datos grabados, mientras que en los segundos se deben recuperar los datos del registro de entrada o salida

Fig.6. En el caso de documentos propios se graban todos los datos, en el resto se recuperan los datos de los registros de entrada y salida. En todos los casos se graba el archivador que lo almacena y se emite el nro. de documento.

En el caso de alta de los archivadores, la aplicación distingue entre archivadores independientes y aquellos que son continuación de otros, puesto que en el segundo caso obtendrán la mayor parte de los datos del archivador inicial (figura 7). Esto se hace para mantener juntos archivadores que encierran información que deba permanecer unida, como en el caso de documentación que se reparte en diferentes meses.

Fig.7. Cuando una documentación que debiera estar en un archivador por su tamaño, tiene que almacenarse en varios, se consideran todos los archivadores anexos al primero y mantienen el código de éste.

En el momento de entrada en la aplicación se capturará el código de usuario y la fecha del día por lo que los campos fecha alta, fecha baja y código del usuario que realiza la modificación serán automáticamente rellenados por el sistema.
En todo momento se podrá cambiar la ubicación de un documento del archivador o usuario que lo contenga (figura 8). Igualmente se podrá cambiar la ubicación de un archivador del usuario, archivador o almacén que lo contenga.

Fig.8. Los documentos y archivadores tienen un campo reservado para el elemento que los contiene que puede ser un archivador, un almacén o un usuario. Cuando se produce un cambio en su ubicación debe reflejarse en ese campo.

El acceso a un elemento determinado del archivo se realizará introduciendo su código. Los informes permitirán la consulta de varios, seleccionando las condiciones a cumplir por los campos del elemento buscado. Tener en cuenta que los documentos tienen los criterios de clasificación del archivador que los contienen (tema, centro, etc.)

Una vez que los archivadores están preparados para ser traslados fuera del alcance de los usuarios, se les adherirá una etiqueta con el contenido de los mismos Esta etiqueta podrá incluir un código de barras que almacene esa información.

2.2.3 De la interfaz.

Se facilita un icono con la dirección de acceso a través del explorador (figura 9). Puesto que los usuarios deben acceder a su ordenador a través de la Intranet corporativa, es posible la captura de su código de red con la correspondiente validación. El sistema comprobará que ese código corresponda a un usuario registrado para acceder a la aplicación y su nivel de seguridad determinará las opciones que puede manipular. Durante toda la sesión la aplicación le permitirá mantenerse conectado y no tener que volver a introducir código y clave cada vez que acceda a una pantalla [4].
La navegación dentro de la aplicación podrá realizarla a través de un menú de barra horizontal con submenús verticales. Las opciones del menú visibles estarán de acuerdo con su nivel de acceso.

Fig.9. Al pulsar el icono, la aplicación leerá el código de red del usuario y comprobará que está dado de alta en la tabla de usuarios de la aplicación. Si es así, aparecerá el menú general de la misma.

Cuando se produzca un error o una acción imprevista en la aplicación se presentará una página de error con el motivo de la misma. Se facilitará la entrada de datos poniendo listas para seleccionar el elemento deseado en el caso de las tablas de clasificación del archivo. En el caso de grabación de documentos y archivos, la aplicación devolverá una página con los datos correspondientes al registro grabado.
Especial atención deberá dedicarse al código de documento propio, de entrada o salida y al de los archivadores tanto nuevos como anexos, pues el mismo deberá anotarse en los documentos o en los archivadores.

2.2.4 De seguridad.

Al hacer la entrada en la aplicación se capturará el código de usuario en la Intranet corporativa y se comprobará que está dado de alta en la tabla de usuarios de la aplicación. Este código se guardará en el alta o modificación que realice en cualquiera de las entidades de la base de datos.
Se efectuarán copias de seguridad de los datos grabados (figura 10) así como de las modificaciones en el software. Estas copias de seguridad se realizarán según el software y hardware disponible [5], volcado a cinta o la red corporativa.

Fig.10. Si el sistema dispone del software y hardware adecuado, se volcará la información a cinta o dvd, en caso contrario se guardará la información en el servidor de la red local en la que se encuentra el servidor de esta aplicación.

Aparte de los ordenadores de los usuarios, la aplicación requiere de un servidor de aplicaciones y de un servidor de base de datos (ambos en la misma máquina). El acceso a éstos también estará protegido por la seguridad de la red corporativa (primero hay que entrar en la red) y posteriormente por contraseñas propias de los servidores que serán sólo conocidas, además de por el personal informático, por el personal de la unidad administrativa que designe la dirección de la misma.
Se efectuarán comprobaciones periódicas de la base de datos con objeto de ver si hay códigos con diferentes significados, que todos los registros de entrada y salida han sido grabados en el registro de documentos de la aplicación, y verificaciones del contenido físico de un archivador con los datos grabados.

2.3. Especificaciones de la interfaz.

Se facilita el acceso a la aplicación mediante un acceso directo a la página de inicio con un icono elegido por el usuario. Se procura crear una interfaz con el usuario que le sea amigable, con un menú con todas las opciones de la aplicación y control de la navegación que permita acceder a las diferentes pantallas de la misma con la mayor facilidad posible. En el caso de grabación de documentos y archivadores se emitirá una pantalla con los datos grabados con especial atención al código asignado por la aplicación (figura 11) y se intentará controlar todos los errores, explicando al usuario la causa del mismo.

Fig.11. Dibujo de alguna de las pantallas que servirán de base para la creación de las que se usarán en la aplicación.

El control de pantallas se hará fundamentalmente con el ratón, pero para facilitar la cumplimentación de los datos de los formularios, se podrá utilizar la tecla [tab] para desplazarse a través del mismo, así como las flechas izda-dcha, backscpace y sup. En los campos en los que deba introducirse un dato ya grabado, se facilitará una lista en la que podrá seleccionar el elemento que corresponda. En los formularios con diversas opciones se dispondrán checkboxs que le permitan seleccionar las opciones que desee considerar.
Se distinguen en la aplicación diferentes tipos de pantallas a través de las cuales se interactúa con la misma.

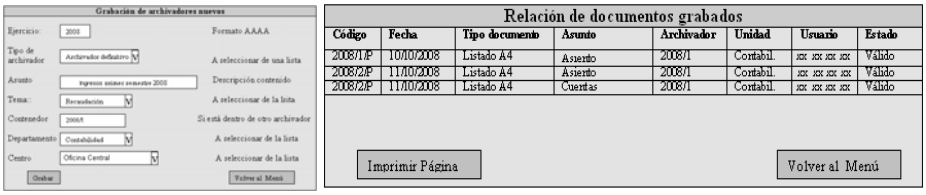


Fig.12. Formulario para la grabación de documentos y listado de de documentos grabados.

Formularios: para la entrada de datos, listados para emisión de los datos seleccionados (figura 12), pantalla mixta: primero presentan los datos grabados y luego esperan la grabación de nuevos datos y pantallas con mensajes presentando los datos grabados o el error producido.

2.4. Modelo de casos de uso.

Los procesos enumerados en el apartado de requisitos se representan en casos de uso [6] (figura 13). En esta fase se corresponden con los apartados del menú general de la aplicación, nos indican los actores implicados y facilitan su posterior desarrollo en la fase de diseño con diagramas de secuencia.

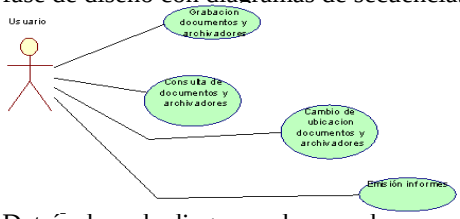


Fig.13. Modelo de caso de uso general de las opciones de las que dispone un usuario: grabación de documentos y archivadores, consulta y modificación de documentos y archivadores, obtención de informes del archivo.

Detrás de cada diagrama de caso de uso, se emitirá una ficha con nombre, autor, fecha y descripción; actores, condiciones que se han de cumplir, flujo de acciones en caso de ejecución normal y en condiciones anómalas.

2.5. Modelo de datos.

Los requisitos establecidos en el apartado de datos nos indican las entidades (figura 14) y sus atributos de los elementos que compondrán las clases persistentes del sistema.

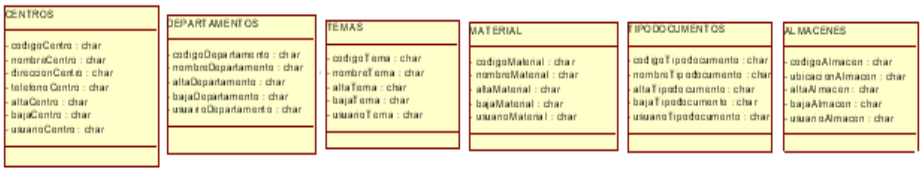


Fig.14. Entidades que definen la clasificación del archivo con sus atributos.

2.6. Análisis de consistencia.

En esta fase se comprueba que los requisitos funcionales y de seguridad tienen su correspondiente caso de uso, que los de datos tienen la clase correspondiente y que los de interfaz responden a una especificación de la misma.

3. Diseño.

3.1. Modelo de clases.

En el diagrama de clases se muestran las clases persistentes de la base de datos (figura 15), con sus atributos, métodos y relaciones entre ellas a partir del modelo de datos definido en la fase anterior.

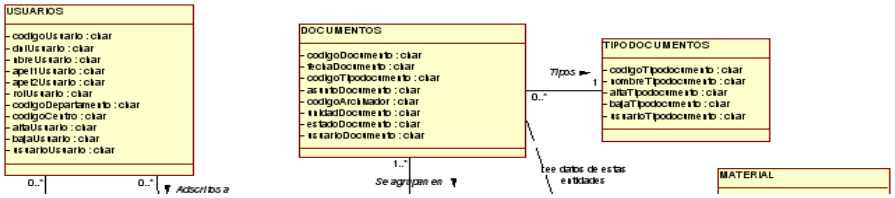


Fig.15. Parte del modelo de clases.

También incluye la descripción de las clases no persistentes que son las clases del servidor de aplicaciones que reciben las peticiones del cliente y devuelven la información recibida de la base de datos.

3.2. Diseño físico de datos.

El diagrama de entidad-relación construido a partir de las clases persistentes del sistema genera una tabla por cada entidad. En dichas tablas se han incluido los atributos de las entidades originales asignándoles un tipo de datos acorde con la clase de información que almacenarán. A partir de este diagrama se emite un script (figura 16) de creación de las tablas que componen la base de datos [7].

```
CREATE TABLE ALMACENES(
          codigoAlmacen                          CHARACTER(2) NOT NULL,
          ubicacionAlmacen                       CHARACTER(40) NOT NULL,
          altaAlmacen                            CHARACTER(10) NOT NULL,
          bajaAlmacen                            CHARACTER(10) NOT NULL,
          usuarioAlmacen                         CHARACTER(8) NOT NULL,
          PRIMARY KEY (codigoAlmacen));
```

Fig.16. Script de creación de la tabla de almacenes de la base de datos.

3.3. Diseño de casos de uso reales.

Para definir las tareas de programación de los jsp, servlets, clases no persistentes, etc. que se instalarán en el servidor de aplicaciones se diseñan diagramas de secuencia (figura 17) que se corresponden con los casos de uso de la fase de análisis [8].

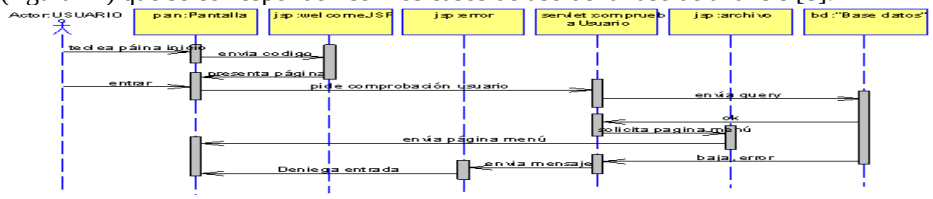


Fig.17. Diagrama de secuencia que detalla el proceso de validación del usuario.

4. Pruebas del sistema.

Antes de empezar la explotación de la aplicación debe realizarse una fase de pruebas que en este caso no tiene problemas de disponibilidad de elementos para la misma, puesto que se tiene todo el material archivado de ejercicios anteriores.
Además de comprobar los procesos del sistema, altas, bajas, consultas y modificaciones, también se tendrá que verificar conexiones, mantenimientos de sesiones, capacidad de carga del servidor, etc.
Aunque parte de las pruebas pueden realizarse con el software adecuado (JUNIT, DBUNIT, JMETER) [9] es necesario la colaboración del personal al que va destinado la aplicación, sobre todo en lo referente a la interfaz.

5. Manuales.

Se deben emitir manuales de explotación para el personal encargado de mantener la aplicación, con instrucciones de instalación, arranque y parada de los servidores de aplicaciones y de base de datos así como del mantenimiento de la aplicación y de la

integridad su base de datos. Igualmente debe recoger la planificación de las copias de seguridad.
También es necesaria la edición de un manual de usuario con todas las pantallas que se le pueden presentar y con indicación de las instrucciones necesarias para la cumplimentación de las mismas.

6. Conclusión.

La realización de este proyecto es francamente viable pues se dispone de los medios materiales de software y hardware necesarios para su implementación, así como del personal informático y de gestión capaz de llevar adelante el proyecto. Esto supondría dotar a la entidad del conocimiento en tiempo real del archivador, usuario o almacén en que se encuentra cada documento en un momento determinado y facilitarle la localización de aquellos de los que se dispone una información incompleta.
Como único problema a destacar es el que los registros de entrada y salida de documentos están fuera del alcance de la aplicación, lo que hace que para evitar la repetición de los datos ya grabados, éstos deban volcarse de los listados diarios de dichos registros a nuestra tabla de documentos lo que lleva a un proceso batch diario. Este inconveniente puede ser subsanado si se incluye esta aplicación dentro del entorno del registro de entrada/salida, lo que permitiría el acceso a la base de datos corporativa

Referencias

1. http://www.csi.map.es/csi/metrica3. Ministerio de Administraciones Públicas. “MÉTRICA. VERSIÓN 3. Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información”. Se toma como base de los pasos a realizar para el desarrollo de de esta aplicación.
3. Sun Microsystems, Inc “The Java EE 5Tutorial For Sun Java System Application Server 9.1” Capítulo I. 2007. Define la distribución del software en una aplicación Java Enterprise Edition 5
4. http://dis.um.es/ ~lopezquesada/documentos/ IES_0607/DFSI/ curso/ UT11/ cookiesJSP.pdf. “Manejo de sesiones con JSP”
5. http://www.symantec.com ”Backup Exec”. Aplicación para planificar los backups del sistema
6 http://alistair.cockburn.us/Use+cases,+ten+years+later. Alistair Cockburn. “Use cases, ten years later”. Grados de detalle escribiendo casos de uso.
7. Herramientas Case como System Archytect de Popkin Software and Systems Inc. permiten la emisión de scripts de creación de bases de datos a partir de diagramas de entidad-relación.
8. http://plugins.netbeans.org/PluginPortal “Netbeans UML”. Este plugin de Netbeans permite la generación de código a partir del modelo UML.
9. http://jakarta.apache.org/jmeter/ Apache JMeter permite la ejecución de pruebas de carga basada en diferentes protocolos: Web, SOAP, DataBase vía JDBC, LDAP, JMS.

Aplicación Web (Java EE) para la gestión integral de un almacén de artículos de una Entidad

Luís Miguel Soria Duarte

Dirección Provincial de la Tesorería General de la Seguridad Social
en Illes Balears

Resumen. El sistema gestiona las entradas y salidas de productos del Almacén y facilita a todos los usuarios de la Entidad un catálogo de productos para que puedan hacer sus peticiones de material. Al final de cada ejercicio se obtienen los datos estadísticos que permiten establecer la estimación de las unidades previstas de cada artículo para el siguiente ejercicio.

1. Introducción

Respondiendo a una petición de la Secretaría Provincial de la Dirección Provincial de la Tesorería General de la Seguridad Social en Illes Balears, para la elaboración de una aplicación para la gestión del Almacén, el 17 de Febrero de 2006 se puso en explotación mi aplicación “Almacén”. Desarrollada en Visual Basic 6 sobre una base de datos Access. Actualmente se sigue utilizando y ha contribuido a la mejora de la gestión del Almacén.

La tecnología Java EE permite implementar una aplicación Web que integre, por medio de perfilado de usuarios, la parte de administración, gestión y usuarios o clientes.

2. Descripción

La aplicación “Almacén” utiliza la tecnología Java Plataform, Enterprise Edition (Java EE) utilizando un modelo de tres capas.

El sistema gestiona las entradas y salidas de productos del Almacén y facilita a todos los usuarios de la Dirección Provincial un catálogo de productos que les permite buscar un artículo para localizar el código necesario para realizar las peticiones de material.

Para que el sistema de gestión del almacén funcione se han creado dos roles de usuarios: El administrador y el editor.

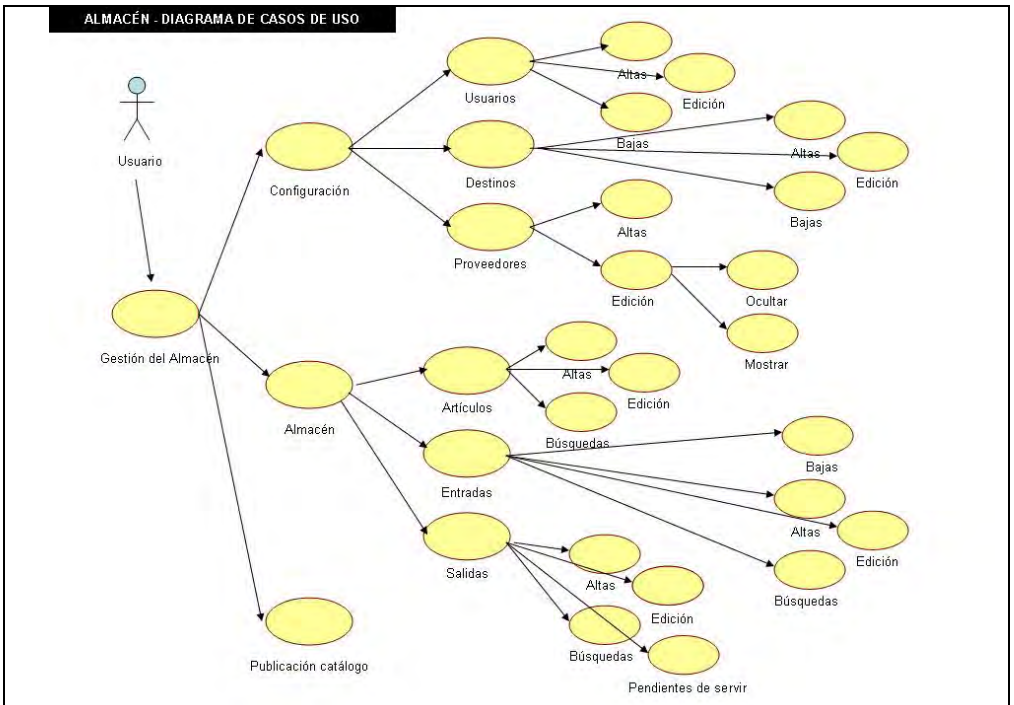


Imagen nº 1. Diagráma de casos de uso

El Administrador puede crear y modificar usuarios, cambiar la configuración de la aplicación, regularizar el stock y publicar los catálogos de productos.

El usuario Almacén puede dar entradas y salidas de material y consultar el Stock de los productos.

Por último estarían los usuarios que pueden acceder al catálogo de productos, que por su puesto estará permanentemente actualizado, para realizar sus pedidos.

Los pedidos se hacen cumplimentando un formulario que se remite por correo electrónico al Jefe del Almacén.

Nota: En el futuro, y gracias a la tecnología Java EE, el objetivo será la creación de un sistema de pedidos que sustituiría las peticiones por correo electrónico corporativo (Lotus Notes).

3. Características técnicas

El sistema aquí descrito usa la tecnología Java EE. Se ha implementado una aplicación web que permite al usuario interactuar con la base de datos del Almacén desde su navegador. Otro requisito que debe cumplir el PC del usuario, es tener instalada la maquina virtual java. En cuanto al servidor donde va alojada la aplicación debe reunir las características que se describen a continuación.

3.1. Servidor web

El sistema de gestión de Almacén se apoya en el Servidor de Aplicaciones web GlassFish V2.

Las pruebas las he realizado montando el servidor de aplicaciones en un PC con Windows XP. No obstante también se puede montar GlassFish V2 en Linux, siendo este sistema operativo en el que finalmente correrá la aplicación cuando se pase a explotación. O al menos esta es mi intención.

GlassFish V2 es un servidor de aplicaciones Java Enterprise Edition 5, es open source y sirve para desarrollar aplicaciones empresariales (transaccionalidad, aplicaciones distribuidas, arquitecturas MOM, desarrollo portales WEB, desarrollo SOA, etc.)

3.2. Servidor de bases de datos

El motor de bases de datos es MySQL (MySQL Server 5.0). Sistema de gestión de base de datos relacional, multihilo y multiusuario.
El software MySQL® proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multi-threaded, multi usuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido. MySQL es una marca registrada de MySQL AB.

El software MySQL tiene una doble licencia. Para este proyecto se utilizaría la licencia GNU General Public License (http://www.fsf.org/licenses/).

MySQL dispone de una API (Application Programming Interface, o Interfaz de Programación de Aplicaciones) para Java que es necesario instalar.

JDBC es un API de Java para acceder a sistemas de bases de datos, y prácticamente a cualquier tipo de dato tabular. El API JDBC consiste de un conjunto de clases e interfaces que permiten a cualquier programa Java acceder a sistemas de bases de datos de forma homogénea. Con esta API, se puede crear un sólo programa en Java que sea capaz de enviar sentencias SQL a la base de datos apropiada, en nuestro caso a MySQL.

La aplicación de Java debe tener acceso a un controlador (driver) JDBC adecuado. Este controlador es el que implementa la funcionalidad de todas las clases de acceso a datos y proporciona la comunicación entre el API JDBC y la base de datos real. JDBC proporciona una interfaz de alto nivel y evita tener que tratar con detalles de bajo nivel para acceder a bases de datos.

En el caso del manejador de bases de datos MySQL, Connector/J es el controlador JDBC oficial. Puede descargarse de la página web de MySQL AB

http://www.mysql.com/products/connector/

3.3. Java EE

Arquitectura Cliente/Servidor basado en la Web en un modelo físico de 3 capas: Cliente, Lógica de Negocio y Datos.

Para la Capa de Presentación se dibujan las páginas web dinámicas que muestran los datos al usuario mediante Servlets y JSPs.

La Lógica de Negocio está implementada en las Clases de Java. Hay que decir que gracias al IDE NetBeans 6.1 en el que se ha elaborado el proyecto, se ha simplificado muchísimo la escritura de código.

En cuanto a la Capa de Acceso a Datos se toma como referencia el modelo de base de datos de la aplicación que ha inspirado este proyecto, y que como se explica en este artículo se basaba en Access. Se trata simplemente de adaptar dicha base de datos a MySQL. El acceso a esta base de datos desde Java se hace, como se explicó anteriormente, con JDBC gracias al driver Connector/J.

3.4. Componentes JavaScript

Se utiliza JavaScript para la validación de datos. También se emplea el control de código libre JCalendar para facilitar al usuario la selección de fechas desde un calendario.

4. Características funcionales

El sistema de gestión de Almacén tiene 4 áreas principales, tal y como se muestra en la imagen nº 2 (pantalla principal de la aplicación):

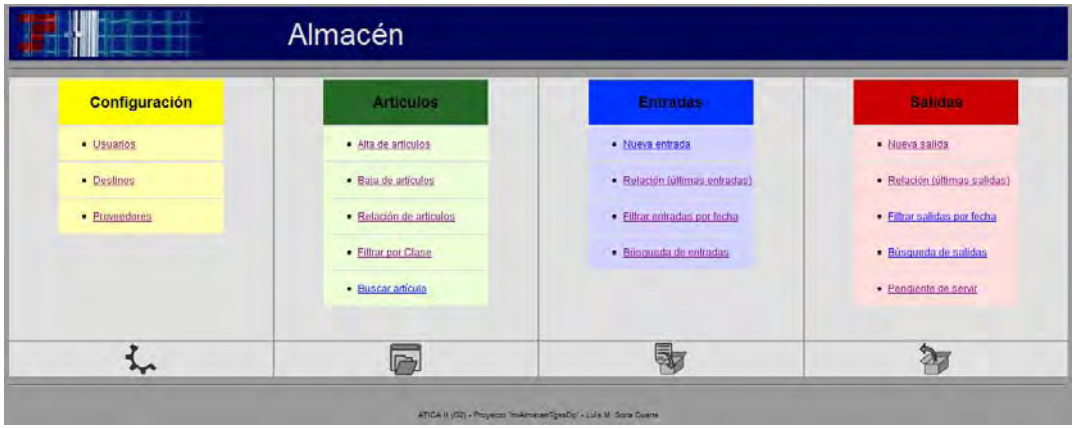


Imagen nº 2. Pantalla principal de la aplicación

Gestión del Almacén:

- Configuración
- Artículos
- Entradas
- Salidas

4.1. Configuración

4.2. Usuarios

Gestión de usuarios (Altas, bajas y modificaciones) asignándoles el rol de administrador o de lector.

Administrador:

- Alta, baja y edición de usuarios.
- Cambiar parámetros de la aplicación.
- Agregar, eliminar y editar productos; regularizar Stock.

- Edición de los parámetros para publicación del catálogo.
- Publicación del catálogo.

Lector:

- Registro de entradas de almacén.
- Registro de salidas de almacén.
- Registro de proveedores.
- Registro de clientes.
- Búsqueda y consulta de artículos.

4.3. Destinos

Los destinatarios son todas las Oficinas y Áreas de la Dirección Provincial a las que se sirve material.

4.4. Proveedores

La gestión de Proveedores es similar a la de destinatarios. No es posible eliminar proveedores, sin embargo se pueden mostrar y ocultar en las listas de los formularios de entradas de artículos.

4.5. Artículos

Relación de artículos: En la relación de artículos está toda la información de cada artículo: Código, nombre, clase a la que pertenece, existencias, si está en el catálogo o no, si es un artículo [1]**asociado**, y el nombre con el que aparece en el catálogo.

Alta de artículos: El "Código" de artículo es un campo numérico que está sujeto al valor del campo "Clase". Por ejemplo, todos los artículos de la clase "Papel impresoras" están en el rango de 5.000 a 5.999.

El campo "Visible", ocultar o mostrar en el catálogo permite que el artículo aparezca en el catálogo al que tienen acceso los usuarios para poder realizar sus pedidos de material. Por ejemplo no está visible ninguno de los artículos de limpieza ya que los usuarios no pueden realizar pedidos de este tipo de productos.

[1] Hay algunos artículos que necesitan un tratamiento especial. Por ejemplo [8022 CARPETA "CAPITAL COSTE" # 100xpaquete] se confecciona en "Reproducciones" a partir del artículo [8014 CARTULINA AZUL PARA CARPETAS ASUNTO, ARCHIVO, APLAZAMIENTO DPTGSS (440 x 310 mm)]. Por lo tanto el artículo 8022 tiene como principal el 8014. De esta forma se controlan las existencias del producto principal que es el que se recibe en el Almacén.

4.5.1. Baja de artículos

Aunque está implementada la baja de artículos, solo debe utilizarse en casos muy concretos y en todo caso nunca dentro de un mismo ejercicio presupuestario pues se perderían las referencias con los movimientos de entradas y salidas del artículo eliminado.

Filtrar por clase de artículo: Es posible obtener un listado de artículos de una clase concreta mediante el filtro de clase de artículo.

Buscar artículo: Para encontrar rápidamente un artículo conociendo su código existe un formulario de búsqueda de artículo.

4.6. Entradas

Nueva entrada: La fecha es obligatoria. El Proveedor se selecciona de una lista, por lo tanto si no existe habrá que crearlo previamente. El artículo se selecciona de la lista desplegable de artículos. Por último es obligatorio escribir una cantidad que será el número de unidades que se sumarán al stock del artículo.

Relación (últimas entradas): La columna Número de registro es un código compuesto por el año y un número secuencial. La numeración se reinicia para cada nuevo ejercicio. Además de los campos de la tabla entradas se muestra el Stock real del artículo (última columna). Es posible acceder a la ficha del artículo pulsando sobre el icono “Lupa”.

Edición de una entrada: Desde el listado de entradas se accede al formulario con los datos de la entrada para su edición. Es posible corregir el número de unidades, incluso pueden introducirse números negativos. Si no se varía el número de unidades no se produce ningún recálculo del stock. Otra posibilidad es rectificar el número de unidades sin que se actualice el stock. Esta última operación solo puede hacerla el administrador para el caso de regularizaciones en las existencias de artículos. No es posible eliminar ningún asiento de entrada.

Filtrar entradas por fecha: Aunque normalmente solo se visualizarán las entradas del ejercicio actual (El resto pasan a un histórico), es posible filtrar las entradas entre dos fechas.

Búsqueda de entradas: Un formulario permite hacer una búsqueda de una entrada concreta si conocemos el número.

4.7. Salidas

Nueva salida: La fecha es obligatoria, y aunque se ofrece la fecha del sistema es posible modificarla. El Destinatario se selecciona de una lista, por lo tanto si no existe habrá que crearlo previamente. El artículo se selecciona de la lista desplegable de artículos. Por último es obligatorio escribir una cantidad que será el número de unidades que se restarán del stock del artículo. En el caso de que no exista suficiente stock del artículo, se ofrece la posibilidad de crear un nuevo registro "Pendiente de Servir".

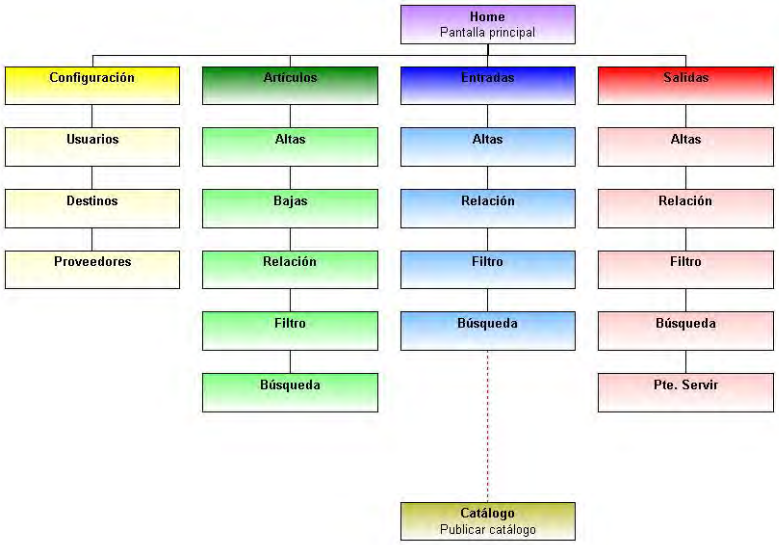
Relación (últimas salidas): La relación está ordenada por número de registro en descendente para ver al principio las últimas salidas. La columna Número de registro es un código compuesto por el año y un número secuencial. La numeración se reinicia para cada nuevo ejercicio. Además de los campos de la tabla salidas se muestra el Stock real del artículo (última columna). Es posible acceder a la ficha del artículo pulsando sobre el icono "Lupa".

Filtrar salidas por fecha: Aunque normalmente solo se visualizarán las salidas del ejercicio actual (El resto pasan a un histórico), es posible filtrar las salidas entre dos fechas.

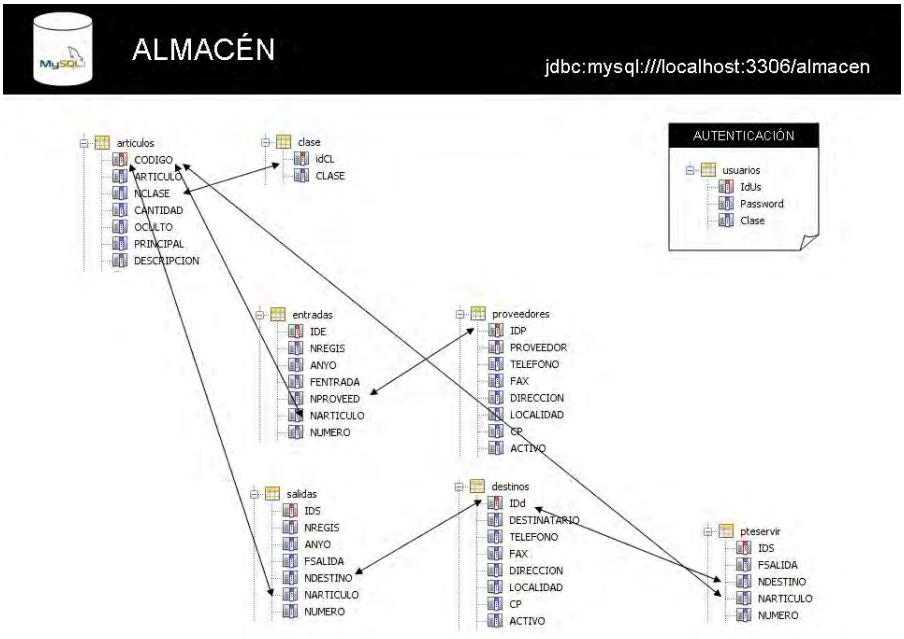
Búsqueda de salidas: Un formulario permite hacer una búsqueda de una salida concreta si conocemos el número.

Pendiente de servir: Cuando se crea una nueva salida y no existe suficiente stock del artículo, se crea un nuevo registro en la tabla de pedidos pendientes de servir. Los campos: Fecha, destinatario, número de unidades y artículo, guardan la información del pedido que no pudo servirse por falta de existencias. El número de unidades son las que han quedado pendientes de servir.

5. Modelo de navegación Web



6. Estructura de la base de datos



En Palma, a 8 de Julio de 2009
Luís M. Soria Duarte
Teléfono: 971 218 432
e-mail: luis-miguel.soria@tgss.seg-social.es

Diseño de una base de datos para gestionar el departamento de salud laboral y seguridad e higiene en el trabajo (GETSALA)

María Antonia Trejo Rodríguez

Unidad Provincial de Informática de Vizcaya

Gerencia de Informática de la Seguridad Social,

Resumen. Realización del diseño de una base de datos que integre toda la gestión para el control tanto de la vigilancia de la salud de los trabajadores como del control y seguimiento de la seguridad e higiene en el trabajo de forma que a través del gestor de base de datos se establezcan las restricciones y dependencias tanto a nivel de usuario como de datos.

1. Introducción

Con el fin de dar contenido al derecho de los trabajadores a una protección eficaz en materia de seguridad y salud en el trabajo, la Ley 31/95 reguló una serie de instrumentos entre los que se encuentra la obligación de los empresarios a establecer:

- Vigilancia de la salud
- Vigilar el cumplimiento de dicha normativa en sus propios centros de trabajo de obras o servicios correspondientes a su actividad
- Organizar la constitución de un servicio de prevención o del recurso a un servicio de prevención ajeno a la empresa.

Bien, pues una vez establecida la normativa, y los protocolos a seguir necesitamos una herramienta que nos ayude a conseguir estas funciones con la mayor eficacia y rapidez, debemos plantearnos qué medios podemos utilizar para el control de todos los datos obtenidos en esta labor.

El uso de los medios electrónicos, informáticos y telemáticos suponen unos beneficios evidentes pero también incrementa el riesgo de vulnerabilidad de los datos almacenados, los cuales deben minimizarse con la adopción de medidas de seguridad, generando confianza en las personas afectadas y en los propios usuarios de las aplicaciones.

Existen diversas normativas que establecen unas pautas de cómo recopilar, procesar o utilizar la información obtenida de los usuarios.

2. Descripción

En cuanto al estudio realizado sobre un departamento de salud laboral de una empresa mediana del sector servicios, a la organización del trabajo que realizan y las herramientas informáticas que han empleado para su gestión debemos hacer las siguientes observaciones:

- Existen dos áreas funcionales: Salud Laboral y Prevención de Riesgos.
- Las herramientas informáticas son: hojas de cálculo Excel, documentos de Word.
- Cada persona guarda sus archivos.
- Los documentos finales obtenidos (informes, procedimientos y normas), sí son compartidos.
- Falta de control de cambios, dificultad para localizar los documentos que contienen los datos relativos a una actividad concreta.
- Cada vez disponen de más información y aumenta la dificultad de organizar y mantener los datos actuales.

3. Solución aportada:

Del análisis de la legislación vigente y de las normas establecidas sobre la materia que nos ocupa he deducido el siguiente modelo de datos, teniendo en cuenta que la información que debe recogerse está relacionada con dos ámbitos, en principio, bien diferenciados pero que, en el ámbito laboral, tienen una relación directa.

- Empleados
- Departamentos
- Datos médicos
- Datos sobre indicadores ambientales y de acceso a los locales
- Datos de las tareas a realizar

3.1 DEFINICIÓN DE ENTIDADES

Tenemos dos unidades básicas de información: empleados y departamentos, por lo tanto estas son las tablas creadas inicialmente:

Ilustración 1.- Diseño de tablas: Empleados, Departamentos, Empleados_departamento

ENTIDADES RELACIONADAS CON DATOS MÉDICOS (SALUD)

Los datos médicos se basan principalmente en fechas de toma de datos médicos, que podemos definir como citas porque habitualmente están programadas por el servicio médico o son solicitadas por los trabajadores al servicio médico pero deben de estar controlada su simultaneidad y duración, por lo tanto tenemos las siguientes entidades:

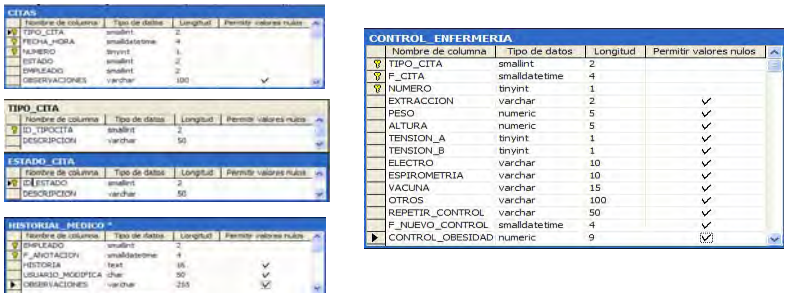


Ilustración 2.- Diseño de tablas Citas, Tipo_cita, Estado_cita, Historial_medico y Control_enfermería

Periódicamente se realizan campañas de prevención y detección de riesgos de salud, aquí he diseñado una única tabla de recogida de datos de esas encuestas (ENCUESTA_SALUD), con un campo clave que identifica el tipo de encuesta que es y las preguntas de cada una de las campañas se recogen en la tabla TIPO_ENCUESTA.

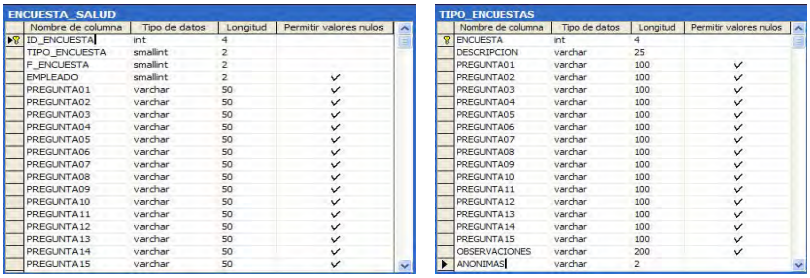


Ilustración 3.- Diseño de tablas de Encuesta_salud y Tipo_encuestas

El control de las peticiones de pruebas sobre empleados realizadas a empresas externas (análisis, radiografías, estudios concretos, etc) se realizan a través de las tablas de: PETICION_SERVICIO, TIPO_SERVICIO y PROVEEDORES

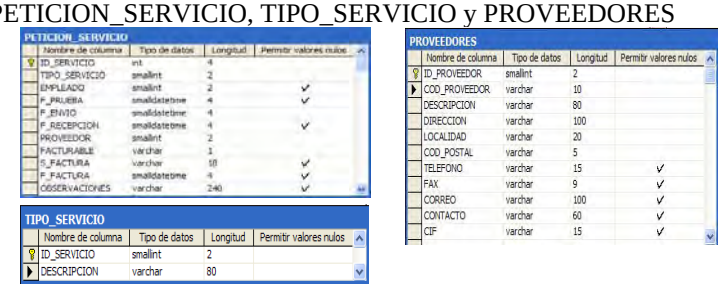


Ilustración 4.- Diseño de tablas de Peticion_servicio, tipo_servicio y Proveedores.

También deben supervisar las bajas de los empleados y para ello creo las siguientes tablas: BAJA_LABORAL, CONTINGENCIA y DIAGNOSTICO.

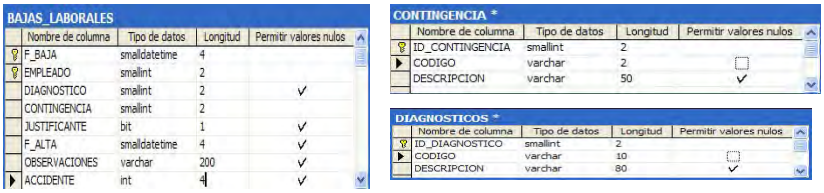


Ilustración 5.- Diseño de tablas de Bajas Laborales, Contingencia y Diagnósticos

Hay que llevar un registro especial para los accidentes de trabajo, bien ocurridos en las instalaciones de la empresa o bien ocurridos en un lugar público durante los desplazamientos de los empleados. Para ello creamos las tablas de: ACCIDENTE_INCIDENCIAS, TIPO_LUGAR_ACCIDENTE, TIPO_ATENCION_ACCIDENTE.

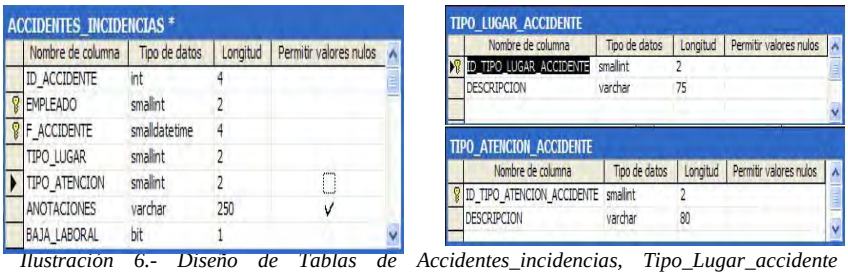


Ilustración 6.- Diseño de Tablas de Accidentes_incidencias, Tipo_Lugar_accidente y Tipo_atención_accidente

ENTIDADES RELACIONADAS CON DATOS DE SEGURIDAD LABORAL

En cuanto a la información que hay que recoger sobre seguridad e higiene en el trabajo, se basa fundamentalmente en condiciones específicas (medioambientales, físicas, etc.) de los departamentos, y también hay que recoger información de las condiciones de trabajo de los empleados, por lo tanto he creado la siguiente estructura de tablas:

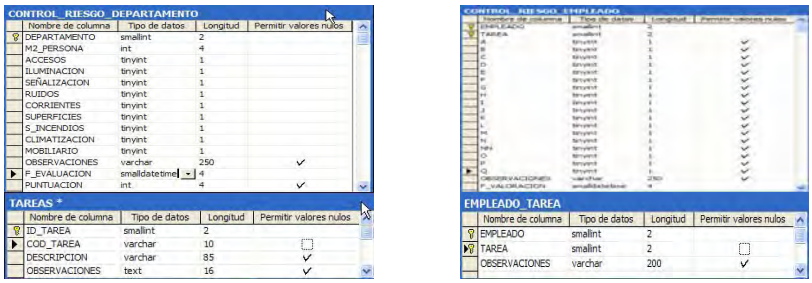


Ilustración 7.- Diseño de tablas de Control riesgo departamento, Control riesgo empleado, Tareas y Empleado tarea

También incluyo aquí las tablas de mantenimiento de empleados en cuanto a su situación, y tipo.

SITUACION_EMPLEADO

	Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
🔑	ID_SITUACION	smallint	2	
	DESCRIPCION	varchar	25	

TIPO_EMPLEADO

	Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
🔑	TIPO_EMPLEADO	smallint	2	
	DESCRIPCION	varchar	50	

Ilustración 8.- Diseño de Tablas de Situacion y tipo empleado

Las relaciones entre estos grupos de tablas quedan de la siguiente manera:

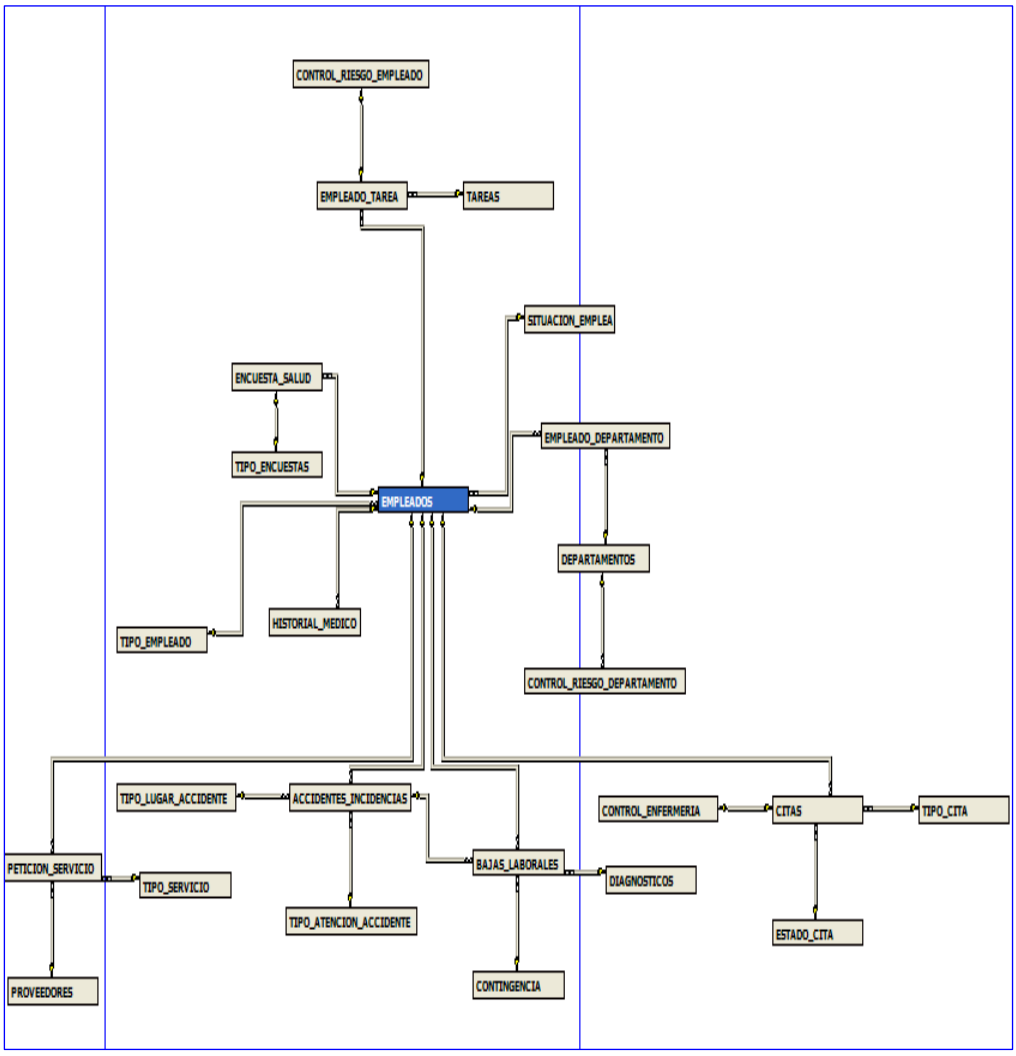


Ilustración 9.- Relaciones entre tablas de Salud y Seguridad Laboral

ENTIDADES RELACIONADAS CON CONTROL DE ACCESOS (LOPD)

Como esta base de datos contiene datos que se consideran según la LOPD de "nivel de seguridad alto" debemos llevar un registro con los datos accedidos que recoja: usuario, fecha y hora del acceso y dato accedido, por lo que debemos de hacer una gestión de usuarios más específica que la de los propios inicios de sesión de sql y los derechos de usuario a nivel de sql, estos también estarán implementados. Se establecen cuatro tablas independientes del resto de tablas de datos, la tabla de usuarios contiene además del usuario que coincide con el inicio de sesión de sql, el nivel o perfil de acceso y una situación Por lo tanto defino las tablas de Usuarios, Situación_usuario, Conexiones y Acceso, quedando el diseño como se ilustra a continuación.

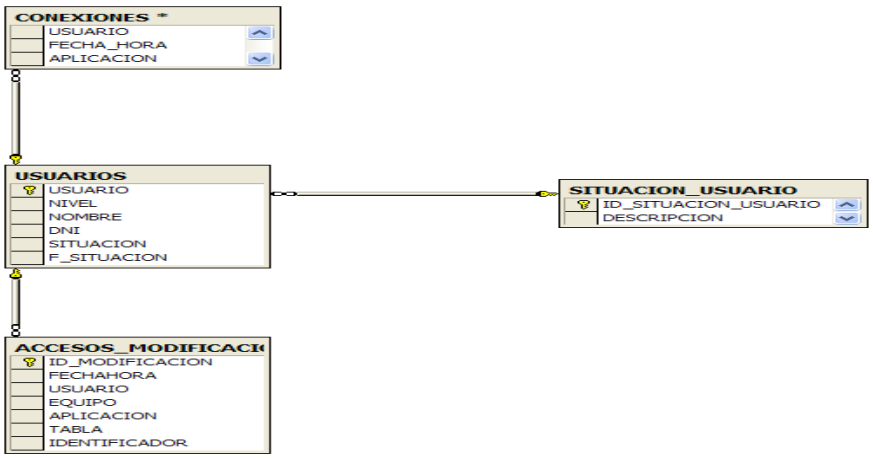


Ilustración 10.-Diseño y Relaciones entre tablas de Accesos

3.2 DEFINICIÓN DE USUARIOS Y ROLES

Dentro del gestor de bases de datos establezco la seguridad a través de usuarios y roles (funciones).

Los roles que se crean en el servidor se corresponden con el nivel de acceso de la tabla de usuarios, puesto que tanto la aplicación como el servidor sql deben controlar los permisos de acceso por usuario a través de estas funciones incluyendo la denegación de servicio para aquellas entidades o columnas no permitidas.

Los roles o funciones serán los siguientes:

.- EDITOR_MEDICO
.- EDITOR_ATS
.- USUARIO_ADMISTRATIVO

El gestor de bases de datos de encarga de grabar las conexiones de los usuarios a través de un sistema de vistas y un procedimiento almacenado.

También se utilizan vistas y desencadenadores para grabar tanto las consultas como las modificaciones de datos en la tabla de accesos.

3.3 DEFINICIÓN DE DESENCADENADORES Y VISTA

He implementado una serie de controles sobre las tablas de la base de datos, se pueden establecer muchos más controles a la vista de las necesidades de los usuarios, aquí expongo un ejemplo de los miles de desencadenadores que se pueden definir y que consiste en actualizar un campo con el usuario que modifica cierta información:

```
/****** Objeto:  desencadenador dbo.TK_ACTUALIZA_ ******/
REATE TRIGGER [TK_ACTUALIZA_HISTORIAL] ON
dbo.HISTORIAL_MEDICO
FOR INSERT, UPDATE, DELETE
AS
DECLARE @CLAVE AS smalliNT
declare @fecha as smalldatetime
 SELECT @CLAVE= empleado,
      @FECHA= f_anotacion  FROM INSERTED
     IF UPDATE(historia)
        BEGIN  UPDATE historial_clinico SET USUARIO_modifica = su-
ser_Sname()
                 WHERE empleado=@CLAVE and f_anotacion=@fECHA
 END
```

También se pueden establecer vistas con valores de campos que nos sirvan para establecer funcionalidades dentro de la aplicación. Pongo por ejemplo una vista creada sobre la tabla CONTROL_ENFERMERIA que contiene un campo con una relación de futuro: F_NUEVO_CONTROL, donde se refleja la fecha a partir de la cual el usuario debe realizar una acción. La vista sería la siguiente:

```
CREATE VIEW dbo.vw_Control_Avisos_enfermeria
AS
SELECT     TOP 100 PERCENT dbo.CONTROL_ENFERMERIA.TIPO_CITA,
dbo.CONTROL_ENFERMERIA.F_CITA,
dbo.CONTROL_ENFERMERIA.NUMERO,
                 dbo.CONTROL_ENFERMERIA.REPETIR_CONTROL,
dbo.CONTROL_ENFERMERIA.F_NUEVO_CONTROL,
dbo.EMPLEADOS.ID_EMPLEADO,
                 dbo.EMPLEADOS.CODIGO_EMPLEADO,
dbo.EMPLEADOS.NOMBRE, dbo.EMPLEADOS.APELLIDO1,
dbo.EMPLEADOS.APELLIDO2, dbo.EMPLEADOS.DIRECCION_CORREO,
dbo.EMPLEADOS.TELEFONO
FROM       dbo.CONTROL_ENFERMERIA INNER JOIN
                 dbo.CITAS ON dbo.CONTROL_ENFERMERIA.TIPO_CITA =
dbo.CITAS.TIPO_CITA AND
                 dbo.CONTROL_ENFERMERIA.F_CITA =
dbo.CITAS.FECHA_HORA AND dbo.CONTROL_ENFERMERIA.NUMERO =
dbo.CITAS.NUMERO INNER JOIN
                 dbo.EMPLEADOS ON dbo.CITAS.EMPLEADO =
dbo.EMPLEADOS.ID_EMPLEADO
```

```
WHERE     (dbo.CONTROL_ENFERMERIA.REPETIR_CONTROL IS NOT
NULL) AND (dbo.CONTROL_ENFERMERIA.F_NUEVO_CONTROL >
GETDATE())
ORDER BY dbo.CONTROL_ENFERMERIA.F_NUEVO_CONTROL
```

3.4 EJEMPLO DE INTERFACE DE USUARIO

Me he planteado el diseño de una aplicación Visual Basic como interface de usuario, de forma que obtenga un ejecutable que englobe todas las funcionalidades tanto de mantenimiento de tablas auxiliares como de tablas de datos, que pueda interactuar fácilmente con un gestor de base de datos como es sql Server pero que también pueda interactuar con la ofimática estándar de Microsoft como es Microsoft office, incluso también con herramientas de correo o de diseño de informes como pueda ser cristal reports, para poder obtener el máximo rendimiento en un entorno de usuario que se base en estaciones Windows XP. Una aplicación que se ejecute desde un único punto común a todos los usuarios (una unidad de red) pero en modo de sólo lectura para ellos, y así poder controlar la versión que utilizan, de forma que se instale en el puesto los elementos mínimos requeridos para la carga del programa, como puedan ser los controles, dll, etc., radicando en la unidad de red el programa, los reports, o plantillas de Excel, Word, o cualquier otro elemento que pueda utilizarse para componer las salidas de información que requiera el usuario. Por lo tanto la aplicación debe de tener un formulario inicial que identifique al usuario que va a acceder y por lo tanto que también identifique los derechos de ejecución a partir del momento de la conexión. Este formulario inicial debe realizar las siguientes operaciones:

1) Conexión con el servidor para validar usuario y contraseña
2) Validar el tipo de usuario y obtener su perfil
3) Controlar el estado del usuario como activo
4) Registrar los intentos fallidos de conexión por contraseña incorrecta
5) Permitir el cambio de contraseña.

Una vez establecida la conexión se accederá a un formulario MDI que controlará el acceso al resto de formularios y la vuelta de estos. En este momento deben presentarse tan solo el acceso a las entidades autorizadas para cada uno de los perfiles, por lo que el programa debe de establecer como visibles o no aquellas partes que correspondan a cada usuario.
También debe implementarse desde este formulario el acceso a las opciones de impresión y a las ayudas de usuario establecidas en cada parte del programa.

4. Conclusiones

I. *La primera conclusión a la que he llegado es que las soluciones informáticas para este departamento deben de llegar avaladas por la dirección de la empresa, puesto que en las reglas de negocio de este departamento están intrínsecamente ligadas la discreción, la independencia y la privacidad de su trabajo.*
II. *La segunda conclusión es que este trabajo no pasa de ser un mero ensayo sobre el rendimiento que puede darse a un sistema gestor de bases de datos para evitar*

controles específicos de las herramientas de programación de interface de usuario. Si los controles tanto de precedencia de datos como de valores de campos o de acceso se hacen a nivel del gestor, da igual si se utiliza programación web, orientada a objetos o el usuario se cuela con la ofimática estándar a través de ODBC, el usuario tan solo podrá hacer lo que el gestor le permita y además siempre quedará registro de sus accesos.

III. *La Herramienta utilizada como interface de usuario debe implementar además de la presentación de datos, utilidades de ofimática que no puedan realizarse desde el gestor de base de datos, como por ejemplo el envío de citaciones utilizando el sistema de correo electrónico corporativo, o la elaboración de gráficos de estadísticas y demás.*

Sistema de Gestión Inmobiliaria: "Inmobiliaria Alcalá"

Juan Villa Martínez
Jesús Resina Hernández

Unidad Provincial de Informática. Granada

Resumen: Se trata de una aplicación web desarrollada como proyecto de los cursos del plan ATICA, "Especialización en Desarrollo de Aplicaciones WEB/JAVA" impartidos por la Universidad de Alcalá de Henares. Dicha aplicación se ha realizado utilizando los conocimientos adquiridos.

Los usuarios (empleados y clientes) de la Inmobiliaria Alcalá, ubicados en distintos puntos de la red o de Internet pueden, en función de su perfil (administrador, empleado o cliente) visualizar los inmuebles disponibles, así como el alta y baja de los mismos y la propia gestión de usuarios.

1. Introducción.

La aplicación Inmobiliaria Alcalá (IA) se ha realizado en **Java EE** (***Java Platform, Enterprise Edition***), plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java, que presenta arquitectura de N niveles distribuida, se basa componentes de software modulares y se ejecuta sobre un servidor de aplicaciones.

Por las características del proyecto, se ha optado por la programación orientada a objetos (POO), puesto que se pueden establecer relaciones entre entidades bien definidas, realizar restricciones a usuarios, así como adaptabilidad y reusabilidad. También se ha adoptado una estructura cliente-servidor por su facilidad para la implementación.

Las herramientas de trabajo para el análisis e implementación del software han sido:

- NetBeans IDE 6.5 y 6.7 como entorno de desarrollo, distribuido por Sun.
- Diagram Designer v.1.21.2 de MeeSoft para diseño de diagramas.
- Apache Derby v.10.4.2.0 de Sun como sistema gestor de base de datos relacional escrito en Java.

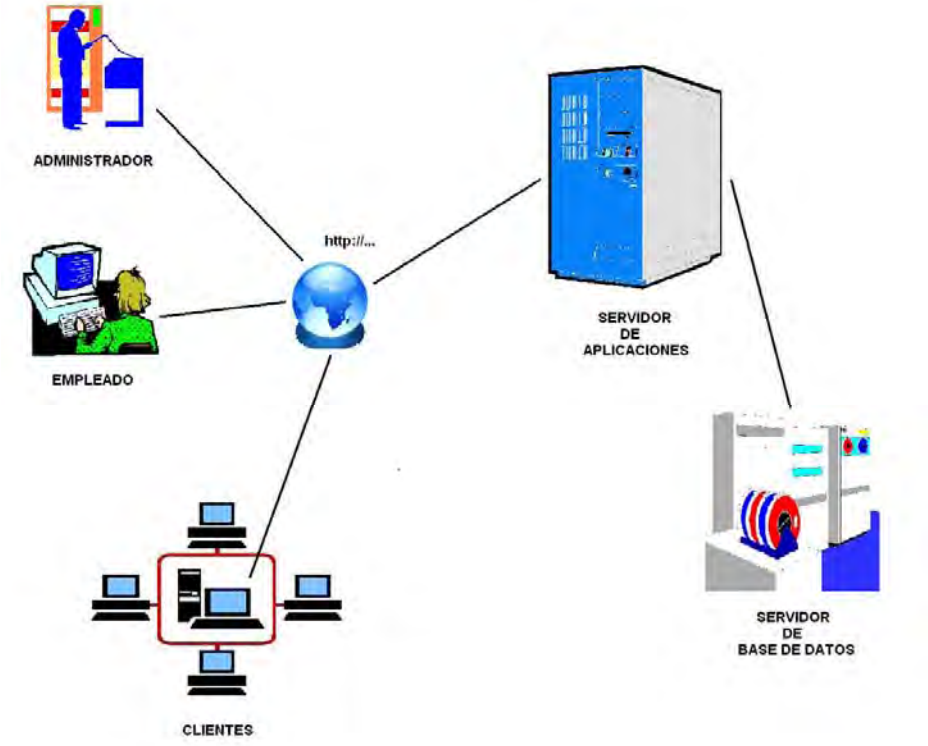
Una vez alojada en un servidor web de aplicaciones con soporte de **EJB** (***Enterprise Java Beans*****),** se tendrá acceso, con identificación para empleados y con registro previo e identificación para clientes.

El enlace con una base de datos relacional permitirá dar alta y baja de empleados, clientes e inmuebles al administrador, alta y baja de inmuebles a los empleados y compra y alquiler de inmuebles a los clientes. Los empleados y administradores tienen la posibilidad de agregar fotos de los inmuebles.

Hay que destacar que en la aplicación se ha tenido en cuenta aspectos de seguridad como validación de usuarios, cifrado de la contraseña, así como gestión de la sesión evitando que un usuario pueda adquirir los privilegios de otro.

2. Sistema de Gestión Inmobiliaria.

La estructura de esta aplicación se compone de una base de datos relacional, una aplicación web alojada en un servidor de aplicaciones, sobre el que interactúan los distintos usuarios. El entorno tecnológico del modelo se compone de tres capas: capa de datos, capa de negocio y capa de presentación. En el diagrama siguiente se muestra el despliegue del sistema.



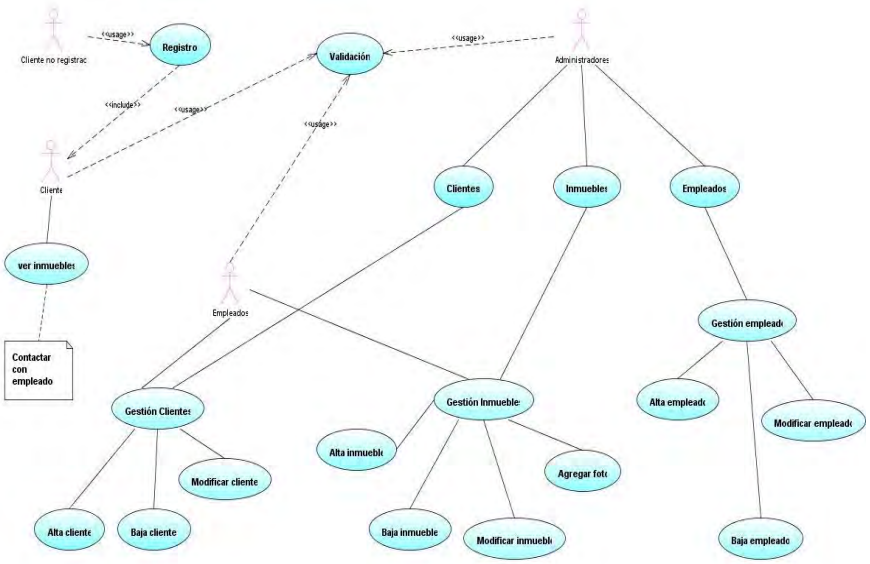
Hay que identificar dos tipos de usuarios intervinientes en el sistema, por un lado los empleados: administrador y genérico, y por otro los clientes: usuarios registrados.

El sistema presenta una serie de condiciones:

- Funcionales: operaciones que pueden realizar los distintos usuarios de la aplicación IA.

- Datos: entidades de datos que se van a utilizar.
- Seguridad: control de acceso y registro que presenta el sistema.
- Interface: diseño personalizado de página Web.

A continuación se muestra el diagrama de casos de uso definido para el sistema.



En el diagrama se pueden ver cuatro roles definidos:

- **Cliente no registrado**.

  Es aquel usuario que visita la página de bienvenida y que sólo va a tener acceso si se registra correctamente.

- **Cliente**.

  Usuario que una vez validado puede tener acceso a la visualización de inmuebles.

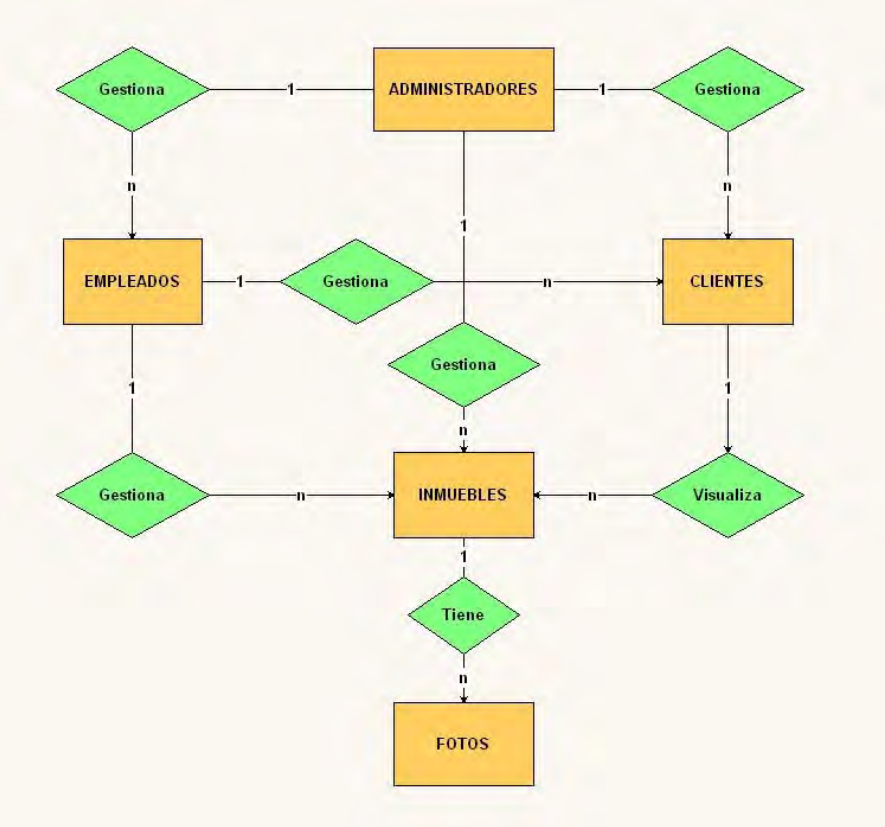
- **Empleados**: se encarga de la gestión de los inmuebles, y de los clientes.

- **Administradores**.

  Es el rol más completo con todas las opciones posibles. Se diferencia del empleado en que puede gestionar a los propios empleados.

2.1 Base de datos.

Como ya se ha indicado, se trata de una base de datos relacional compuesta por cuatro tablas, tal y como se puede ver en el siguiente diagrama conceptual (Entidad-Relación).

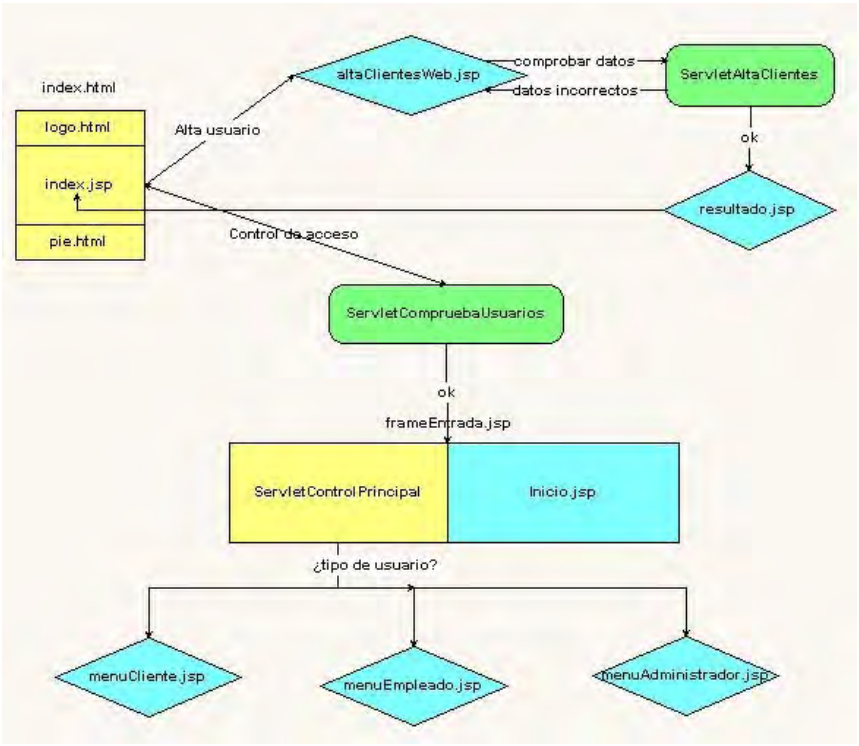


3. Funcionamiento de la aplicación.

Para comprender el funcionamiento de esta aplicación se muestra por un lado los diagramas de flujo de los distintos objetos del sistema y a continuación de forma más gráfica las pantallas de la capa de presentación.

3.1 Diagrama de flujo. Menú principal.

Página de bienvenida que permite al usuario registrado (cliente, empleado y administrador) entrar en la aplicación y al cliente no registrado darse de alta.



3.2 Capa de presentación. Menú principal.

La aplicación se ha ejecutado con distintos navegadores y versiones, funcionando perfectamente con Mozilla Firefox 3.0.6 e Internet Explorer 7 y 8.

4. Conclusión.

Se trata de una aplicación de gestión donde intervienen distintos usuarios con distintos perfiles y en función de el realizar sus funciones. Se ha tenido en cuenta como se dijo al principio el control de acceso y seguridad de la página. También se han depurado los errores que pueda cometer el usuario en la utilización de los formularios.

5. Bibliografía y software.

Documentación impresa:

- √ Manuales del curso *"Especialización en Desarrollo de Aplicaciones WEB/JAVA"* proporcionados por la Universidad de Alcalá.
- √ Diseño de páginas Web con XHTML, JavaScript y CSS. Juan Carlos Orós. 2006. Ed. RA-MA.
- √ Piensa en Java. Bruce Eckel. Ed. Pearson/Prentice Hall
- √

Documentación Web:

- √ ***SUN.com***: La página de los creadores de Java, con documentación, manuales,… http://java.sun.com
- √ ***IBM.com***: IBM proporciona numerosos manuales y artículos sobre Java, entre otros. http://www.ibm.com/developerworks/views/java/library.jsp

- √ ***JavaHispano***: documentación y foros de ayuda sobre java en castellano. http://www.javahispano.org/
- √ ***Enterprise JavaBeans, second edition***, de Richard Monson-Haefel. Ed. O'Reilly. Disponible en HTML gratuitamente en: http://docstore.mik.ua/orelly/java-ent/ebeans/index.htm
- √ ***Java Servlet Programming***, de Jason Hunter con William Crawford. Ed. O'Reilly. Disponible gratuitamente en formato HTML en: http://docstore.mik.ua/orelly/java-ent/servlet/index.htm
- √ Kodo™ 4.1.4 Developers Guide for JPA/JDO, que posee numerosos ejemplos de JPQL. http://edocs.bea.com/kodo/docs41/full/html/
- √ The Java EE 5 Tutorial. http://java.sun.com/javaee/5/docs/tutorial/backup/update3/doc/QueryLanguage.html
- √ Oracle: Queriyng JPA Entities with JPQL and Native SQL http://www.oracle.com/technology/pub/articles/vasiliev-jpql.html
- √ ***Foros del Web***: Foro en castellano dedicado al diseño y a la programación de sitios web. http://www.forosdelweb.com/

Software Utilizado:

- √ **NetBeans:** http://www.netbeans.org/
- √ **Diagram Designer** v.1.21.2 de MeeSoft.
- √ **Apache Derby:** http://db.apache.org/derby/
- √ **Internet Explorer v 6, 7 y 8.**
- √ **Mozilla Firefox v 2.x y 3.x:** http://www.mozilla-europe.org/es/
- **JSCalendar:** ***The DHTML Calendar:*** http://dynarch.com/mishoo/
- **TinyMCE - *Javascript WYSIWYG Editor:*** http://tinymce.moxiecode.com
- **Lightbox JS - Fullsize Image Overlays:** http://www.huddletogether.com
- **SortTable:** http://www.kryogenix.org/code/browser/sorttable/
- **GIMP - *The GNU Image Manipulation Program:*** http://www.gimp.org
- **Apache Jakarta Commons fileUpload:** http://commons.apache.org/fileupload
- **COOL TEXT Graphics Generator:** http://www.cooltext.com

Notas