

Complu – Meeting

Autores

David García Higuera

Pablo Langa Blanco

David Toro Córdoba

Profesor director

Fernando Sáenz Pérez

*(Proyecto de Sistemas Informáticos 2005 - 2006, Facultad de Informática,
Universidad Complutense de Madrid).*

Palabras clave para búsqueda bibliográfica

- Reunión.
- Tarea.
- Agenda.
- Periodicidad.
- Gestor.
- Grupo.
- Mensaje.
- Restricción.
- Horario.
- Calendario.

Resumen del proyecto:

Complu – Meeting es un sistema que gestiona agendas y permite crear reuniones o citas entre sus usuarios de una manera sencilla e intuitiva. Además permite gestionar usuarios en grupos y enviar mensajes entre ellos. La aplicación otorga la posibilidad de configurar avisos sobre la agenda a la cuenta de correo electrónico del usuario.

Ofrece facilidades para asignar fechas libres entre usuarios convocados a reuniones con distintas restricciones en sus respectivas agendas. Estas características hacen a esta aplicación útil como herramienta de gestión de reuniones entre grupos de trabajo u organizaciones.

Su acceso es a través de Internet con lo cual no es necesaria su instalación en los clientes permitiéndoles acceder desde cualquier PC y plataforma conectados a la red.

Project's summary:

Complu – Meeting is a system that manages agendas and allows to create meetings or appointments between users of a simple and intuitive way. So allows managing users in groups and sending messages between them. The application grants the possibility of setting notifications about the agenda to the user's e-mail.

It offers facilities to assign free dates between summoned users for meeting with different restrictions in its respective agendas. These characteristics make this application usefull as management tool of meeting between work's groups or organizations.

It access is across Internet with which it is not necessary it installation on the clients allowing them to access from any PC and platform connected to the net.

Índice:

1. Tecnologías	5
2. Manual de instalación	12
2.1. Instalación de Visual Studio 2005 con ASP.NET	
2.2. Instalación y configuración de iis	
2.3. Instalación, configuración y sincronización de la base de datos	
3. Manual de usuario	29
3.1. Inicio	
3.2. Alta	
3.3. Principal	
3.4. Nueva tarea	
3.5. Lista de tareas	
3.6. Detalles de tarea	
3.7. Nueva reunión	
3.8. Lista de reuniones	
3.9. Detalles de reunión	
3.10. Usuarios y grupos	
3.11. Nuevo grupo	
3.12. Nuevo mensaje	
3.13. Lista de mensajes	
3.14. Periodicidad	
3.15. Archivos adjuntos	
3.16. Usuarios convocados	
3.17. Preferencias	
3.18. Probar compatibilidades	
4. Requisitos	80
4.1. Introducción	
4.1.1. Propósito	
4.1.2. Ámbito del sistema	
4.1.3. Definiciones, acrónimos y abreviaturas	
4.1.4. Referencias	
4.1.5. Visión general del documento	
4.2. Descripción general	
4.2.1. Perspectiva del producto	
4.2.2. Funciones del sistema	
4.2.2.1. Alta en el sistema	
4.2.2.2. Baja en el sistema	
4.2.2.3. Vistas del calendario	
4.2.2.3.1. Vista del mes	
4.2.2.3.2. Vista de semana	
4.2.2.3.3. Vista de día	
4.2.2.4. Tareas	
4.2.2.5. Lista de tareas	
4.2.2.6. Reunión	
4.2.2.7. Lista de reuniones	
4.2.2.8. Usuarios y grupos	
4.2.3. Características del usuario	

4.2.4.	Restricciones	
4.2.5.	Suposiciones y dependencias	
4.2.5.1.	Suposiciones	
4.2.5.2.	Dependencias	
4.3.	Requisitos específicos	
4.3.1.	Interfaces externas	
4.3.1.1.	Interfaces de usuario	
4.3.1.2.	Interfaces hardware	
4.3.1.3.	Interfaces software	
4.3.1.4.	Interfaces de comunicación	
4.3.2.	Funciones	
4.3.2.1.	Alta en el sistema	
4.3.2.2.	Baja en el sistema	
4.3.2.3.	Vistas del calendario	
4.3.2.3.1.	Vista de mes	
4.3.2.3.2.	Vista de semana	
4.3.2.3.3.	Vista de día	
4.3.2.4.	Tareas	
4.3.2.5.	Lista de tareas	
4.3.2.6.	Reunión	
4.3.2.7.	Lista de reuniones	
4.3.2.8.	Usuarios y grupos	
5.	Diseño y arquitectura de la aplicación	93
5.1.	Introducción	
5.2.	Arquitectura	
5.3.	Diseño	
5.3.1.	Modelo de desarrollo	
5.3.2.	Diseño de alto nivel	
5.3.3.	Diseño a nivel de código fuente	
6.	Diseño de la base de datos	132
7.	Problemas resueltos	144
8.	Bibliografía	146

1. Tecnologías

Las principales tecnologías utilizadas en el desarrollo de nuestra aplicación, se basan en la plataforma .net de Microsoft. La base de datos utilizada es MYSQL 5.0 y el conector entre ambas tecnologías es ODBC, que es un estándar de acceso a bases de datos desarrollado por Microsoft.

A continuación, se pasan a detallar las principales características de las tecnologías utilizadas.

Plataforma .NET

La plataforma Microsoft .NET proporciona todas las herramientas y tecnologías necesarias para desarrollar aplicaciones Web distribuidas. Expone un modelo de programación para aplicaciones multinivel consistente e independiente del lenguaje de programación, a la vez que proporciona una interoperabilidad y una fácil migración desde las tecnologías actualmente existentes.

El .NET Framework es un entorno de desarrollo y ejecución multilenguaje de componentes que proporciona los bloques básicos para desarrollar Aplicaciones Web y Servicios Web. A grandes rasgos, está formado por un runtime universal (Common Language Runtime, o CLR), una librería de clases unificada común (Base Class Library) y ASP.NET, con Formularios Web, Servicios Web y Servicios de Acceso a Datos (ver figura 1).

El Common Language Runtime (CLR) es un motor de ejecución de código de alto rendimiento. El código destinado a ejecutarse sobre el runtime se denomina código gestionado. Cuando un componente se ejecuta, el runtime es el responsable de crear objetos, invocar llamadas a métodos, gestionar la reserva de memoria (incluyendo la recolección de basura), iniciar y detener hilos y procesos, garantizar la política de seguridad y satisfacer cualquier dependencia que el componente tenga sobre otros componentes, con independencia del lenguaje de programación en el que estén escritos.

Principales características:

ASP.NET es un entorno de programación sobre el CLR y las librerías de clases básicas para desarrollar potentes aplicaciones Web. Los modelos de programación basados en Formularios Web proporcionan un modo fácil y potente de crear interfaces de usuario dinámicas, y los Servicios Web de ASP.NET ofrecen los bloques básicos para construir aplicaciones Web distribuidas que utilizan estándares como XML.

Soporte de múltiples lenguajes compilados.

ASP.NET, aunque no descarta totalmente la idea de los lenguajes de scripting, introduce soporte para lenguajes completamente compilados,

ofreciendo al desarrollador escribir el código en Visual Basic, C++ o C#. De hecho, gracias a la incorporación del conjunto de tipos comunes y el funcionamiento global del CLR, ofrece un verdadero entorno neutral independiente del lenguaje para las aplicaciones Web.

Mejor rendimiento.

Cuando solicitamos una página ASP.NET por primera vez, ASP.NET genera dinámicamente una clase que representa esa página. El parseador verifica e interpreta los contenidos de la clase derivada y la pasa al compilador, que es el responsable de compilar el código a un lenguaje intermedio (Microsoft Intermediate Language - MSIL), un conjunto de instrucciones independientes de la CPU. Tras compilar, la clase derivada se almacena en un caché de salida, de modo que las subsiguientes peticiones de la misma página se leen del caché y se ejecutan mucho más rápido; esto permite incrementar el rendimiento tras la primera ejecución ya que se utiliza la versión compilada y el contenido de la página no tiene que ser parseado de nuevo. Incluso los lenguajes de scripting son compilados antes de su ejecución, de modo que páginas JScript también tendrán una mejora en su rendimiento. Al final, ASP.NET instancia la clase y el CLR la ejecuta para devolver HTML al cliente. El lenguaje HTML incluido en el flujo de salida puede provenir de código HTML que esté presente en la página, de los controles de la página o cualquier script que genere HTML por sí mismo.

Mayor escalabilidad.

La Web es fundamentalmente un modelo sin estado y sin correlación entre peticiones HTTP. Esto puede dificultar el proceso de desarrollo de aplicaciones Web, ya que las aplicaciones generalmente necesitan mantener el estado entre varias peticiones. ASP.NET mejora y extiende los servicios de gestión de estado introducidos por ASP al proporcionar tres tipos de estado a las aplicaciones Web: aplicación, sesión y usuario. El estado de aplicación, como en ASP, es específico de la instancia de una aplicación y no es persistente. El estado de sesión es específico de una sesión de usuario con la aplicación. El estado de una sesión ASP.NET se almacena en un proceso separado que incluso puede configurarse para ser almacenado en una base de datos, en un proceso o en una máquina separada. Esto hace que podamos utilizar el estado de sesión cuando la aplicación se implanta en una granja de servidores. El estado de usuario es similar al estado de sesión, pero generalmente no expira y es persistente. El estado de usuario es útil para almacenar preferencias de usuario y demás información de personalización. Todos los servicios de gestión de estado se implementan como módulos HTTP, de modo que pueden ser fácilmente añadidos o eliminados del flujo de una aplicación. Un módulo de terceros puede proporcionar servicios de gestión de estado con independencia de los proporcionados por ASP.NET si hubiese necesidad de ello.

Además, ASP.NET utiliza un conjunto de mecanismos de caché que mejoran el rendimiento de las aplicaciones Web. Como hemos visto, todas las

páginas ASP.NET son compiladas en el servidor, y cacheadas. Se actualizan en el caché únicamente cuando la página cambia o cuando expira el periodo de caché predeterminado.

ASP.NET nos permite utilizar el estado de sesión incluso con navegadores que tienen deshabilitado el soporte de cookies. Las sesiones sin cookies utilizan URLs para pasar el identificador de sesión a una página ASP.NET. Una sesión sin cookies exige codificar datos en una URL, proceso que el navegador realiza automáticamente. Todos estos mecanismos nos permiten añadir más servidores Web a medida que vayamos teniendo más tráfico.

Soporte para la actualización "en vivo" de las aplicaciones

Un administrador no necesita detener el servidor Web ni la aplicación para actualizar la configuración. Los archivos de la aplicación nunca se bloquean, de modo que pueden sobrescribirse incluso cuando la aplicación está ejecutándose. El runtime recogerá y propagará automáticamente cualquier cambio que hayamos hecho. Cuando se actualizan los archivos, el sistema conmuta transparentemente a la nueva versión. El sistema detecta los cambios de los archivos, lanza una nueva instancia de la aplicación utilizando el nuevo código y empieza a rutear las nuevas peticiones entrantes a esa aplicación. Cuando todas las peticiones pendientes procesadas por la instancia de la aplicación existente han sido gestionadas, esa instancia se elimina.

Modelo de programación:

Formularios web.

Los Formularios Web (Web Forms) proporcionan un rico conjunto de servicios para ayudar a los desarrolladores a escribir páginas Web interactivas; son páginas Web que nos permiten escribir código de modo parecido a como lo hacemos hoy con ASP. Están diseñados sobre un modelo de programación orientado a objetos, permitiendo reutilizar el código y separar el código de la aplicación del contenido del interfaz de usuario de la página. Por ejemplo, con Visual Basic.NET arrastramos gráficamente los controles a un Formulario Web y posteriormente codificamos los procedimientos de los eventos o controles asociados. En ASP tradicional eso no es posible, porque no hay ningún enlace entre los controles del interfaz de usuario y su código de servidor asociado; en ASP.NET sí que existe ese enlace, y de este modo promueve una aproximación más estructurada. Esto permite aprovechar mejor las características heterogéneas de nuestro equipo de desarrollo: los creativos utilizarán las herramientas de diseño y se encargarán del atractivo visual del sitio y los programadores se responsabilizarán de la codificación. Otra de las ventajas es que la localización resulta mucho más simple, dado que la página del interfaz de usuario tiene elementos exclusivamente de interfaz, en lugar de una mezcla de interfaz y código.

Controles de servidor

La librería de clases del .NET Framework, que proporciona las clases comunes a todas las aplicaciones .NET, contiene varias clases que representan controles de servidor. Estas clases están divididas en dos categorías: controles HTML y controles Web. ASP.NET proporciona un modelo de aplicación Web que aprovecha los controles de servidor y una infraestructura que simplifica el trabajo de crear las páginas. Existen controles que encapsulan elementos comunes del interfaz de usuario HTML como cajas de texto, listas, menús desplegables, etc., y otro conjunto adicional de controles Web más complejos que realizan tareas avanzadas como visualizar calendarios, tablas que validan entradas de usuario o gestionar la rotación de publicidad de la página. Estos controles se ejecutan en el servidor Web y simplemente proyectan su interfaz como HTML a un navegador. En el servidor, los controles automáticamente mantienen sus estados de selección, y exponen un modelo de programación orientado a objetos al desarrollador mediante propiedades, métodos y eventos definidos en las clases para generar el código de servidor, ofreciendo una mayor claridad en el modelo de programación. Por otra parte, si el conjunto de controles que proporciona el sistema no nos es suficiente, podemos escribir nuestros propios controles, heredando parte de su comportamiento de algunos de ellos.

ADO.NET

ADO.NET es una evolución del modelo de acceso a datos de ADO que controla directamente los requisitos del usuario para programar aplicaciones escalables. Se diseñó específicamente para el Web, teniendo en cuenta la escalabilidad, la independencia y el estándar XML.

ADO.NET utiliza algunos objetos ADO, como Connection y Command, y también agrega objetos nuevos. Algunos de los nuevos objetos clave de ADO.NET son DataSet, DataReader y DataAdapter.

La diferencia más importante entre esta fase evolucionada de ADO.NET y las arquitecturas de datos anteriores es que existe un objeto, DataSet, que es independiente y diferente de los almacenes de datos. Por ello, DataSet funciona como una entidad independiente. Se puede considerar el objeto DataSet como un conjunto de registros que siempre está desconectado y que no sabe nada sobre el origen y el destino de los datos que contiene. Dentro de un objeto DataSet, de la misma manera que dentro de una base de datos, hay tablas, columnas, relaciones, restricciones, vistas, etc.

El objeto DataAdapter es el objeto que se conecta a la base de datos para llenar el objeto DataSet. A continuación, se vuelve a conectar a la base de datos para actualizar los datos de dicha base de datos a partir de las operaciones realizadas en los datos contenidos en el objeto DataSet. En el pasado, el procesamiento de datos se basaba principalmente en la conexión. Ahora, con el fin de proporcionar a las aplicaciones multinivel mayor eficacia, se

está adoptando para el procesamiento de datos un enfoque basado en mensajes que manipulan fragmentos de información. En el centro de este enfoque se sitúa el objeto `DataAdapter`, que proporciona un puente entre un objeto `DataSet` y un almacén de datos de origen para recuperar y guardar datos. Para ello, envía solicitudes a los comandos SQL apropiados que se ejecutan en el almacén de datos.

Los principales objetos son los siguientes:

Objetos Connection. Para conectar con una base de datos y administrar las transacciones en una base de datos.

Objetos Command. Para emitir comandos SQL a una base de datos.

Objetos DataReader. Proporcionan una forma de leer una secuencia de registros de datos sólo hacia delante desde un origen de datos SQL Server.

Objetos DataSet. Para almacenar datos sin formato, datos XML y datos relacionales, así como para configurar el acceso remoto y programar sobre datos de este tipo.

Objetos DataAdapter. Para insertar datos en un objeto `DataSet` y reconciliar datos de la base de datos.

ODBC(Open DataBase Connectivity)

ODBC es una interfaz de programación de aplicaciones estándar (API) que permite acceder a datos contenidos y manejados por sistemas de gestión de bases de datos (DBMSs). Utilizando ODBC, las aplicaciones pueden acceder a datos almacenados en una gran variedad de ordenadores personales, mini ordenadores y grandes ordenadores, incluso aunque cada DBMS utilice un formato diferente para guardar la información.

Entre sus características, destacan:

ODBC es una interfaz de programación de aplicaciones estándar que utiliza SQL (Structured Query Language).

Oculto al programador la complejidad a la hora de conectarse a un origen de datos: por ejemplo, el acceso a los datos a través de redes de comunicación es transparente.

Permite a múltiples aplicaciones acceder a múltiples orígenes de datos.

Proporciona un modelo de programación homogéneo, es decir, bases de datos muy diferentes se manejan, vía ODBC, como si fueran idénticas, siendo ODBC el encargado de realizar las adaptaciones necesarias.

Se basa en el modelo cliente / servidor.

MYSQL 5.0

Hemos escogido este sistema gestor de bases de datos ya que es un sistema de libre distribución y se adapta a las necesidades de la aplicación.

Algunas de las características más importantes de Mysql son:

MySQL es un sistema de gestión de bases de datos relacionales.

Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en una misma tabla. Esto añade velocidad y flexibilidad. La parte SQL de "MySQL" se refiere a "Structured Query Language". SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL. El estándar SQL ha evolucionado desde 1986 y existen varias versiones. En este manual, "SQL-92" se refiere al estándar del 1992, "SQL : 1999" se refiere a la versión del 1999, y "SQL:2003" se refiere a la versión actual del estándar. Usamos la frase "el estándar SQL" para referirnos a la versión actual de SQL.

El software de bases de datos MySQL es un sistema cliente / servidor que consiste en un servidor SQL multi-threaded, programas y bibliotecas cliente, herramientas administrativas y un amplio abanico de interfaces de programación para aplicaciones (APIs).

Escrito en C y en C++.

Probado con un amplio rango de compiladores diferentes.

Funciona en diferentes plataformas.

Uso completo de multi-threaded mediante threads del kernel. Pueden usarse fácilmente múltiples CPUs si están disponibles.

Proporciona sistemas de almacenamientos transaccionales y no transaccionales.

Un sistema de reserva de memoria muy rápido basado en threads.

SMTP(Simple Mail Transfer Protocol)

Protocolo simple de transferencia de correo electrónico. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras o distintos dispositivos (PDA's, Celulares, etc).

Con el tiempo se ha convertido en uno de los protocolos más usados en internet. Para adaptarse a las nuevas necesidades surgidas del crecimiento y

popularidad de internet se han hecho varias ampliaciones a este protocolo, como poder enviar texto con formato o archivos adjuntos.

SMTP se basa en el modelo cliente-servidor, donde un cliente envía un mensaje a uno o varios receptores.

En el conjunto de protocolos TCP/IP, el SMTP va por encima del TCP, usando normalmente el puerto 25 en el servidor para establecer la conexión.

2. Manual de instalación

La instalación consta de tres fases:

1. Instalación de Visual Studio 2005 con ASP.NET.
2. Instalación y configuración del iis.
3. Instalación, configuración y sincronización de la base de datos.

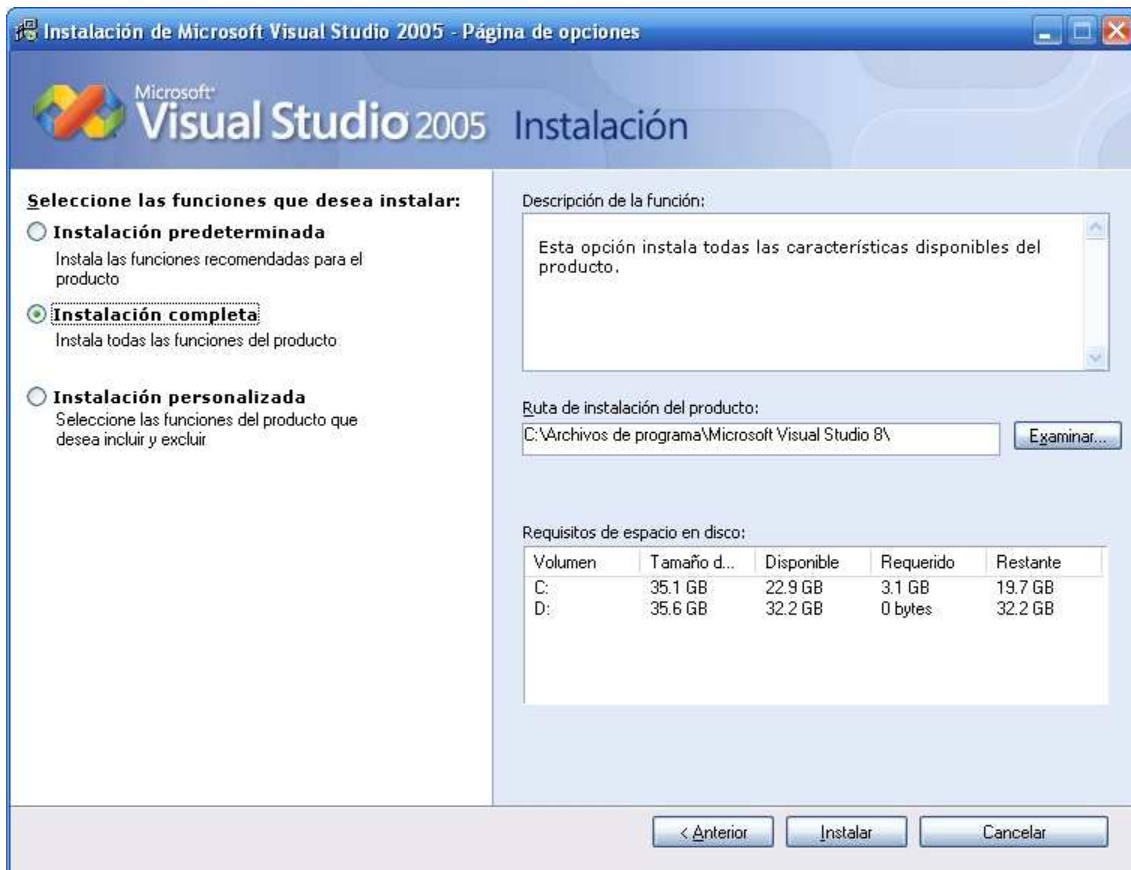
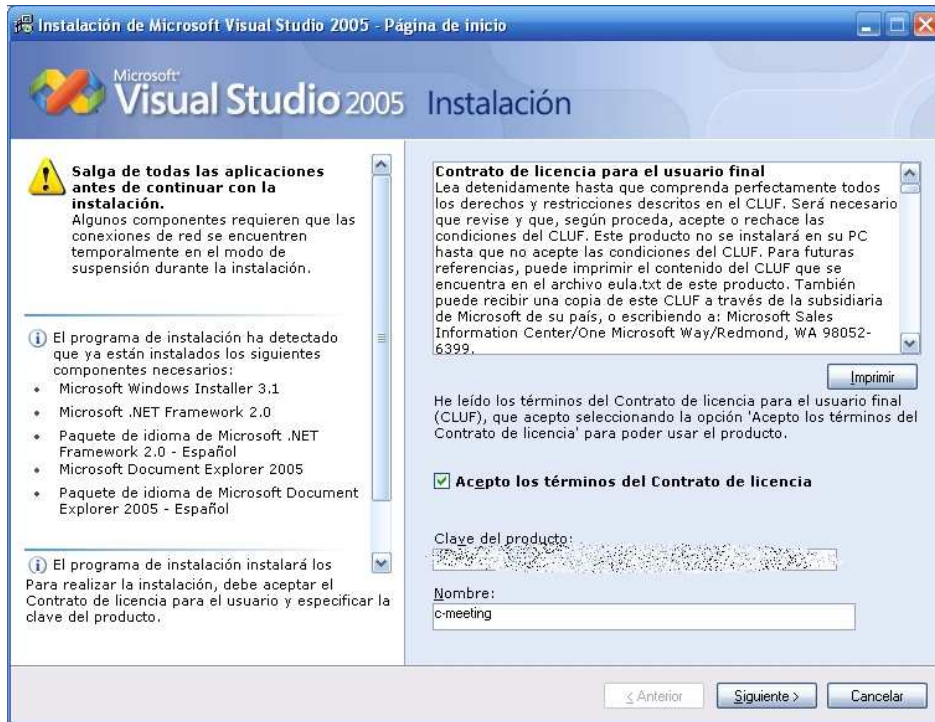
Los requisitos para el servidor, son los propios del Visual Studio 2005, puesto a que el resto de aplicaciones necesitan menor potencia de cálculo. Luego es recomendable que el servidor cumpla:

	Procesador	Sistema operativo	RAM	Disco duro	Pantalla	Otros
Mínimo	tipo Pentium a 600 MHz	Windows®003 Windows XP Windows 2000	128MB	1.3 GB	800x600,256 colores	Unidad DVD Mouse
Recomendado	Tipo Pentium a 1 GHz		256MB		1024x768, alta densidad	

2.1 Instalación de Visual Studio 2005 con ASP.NET.

Seguimos los pasos para instalar nuestra versión de Visual Studio 2005 como se muestra en las imágenes:



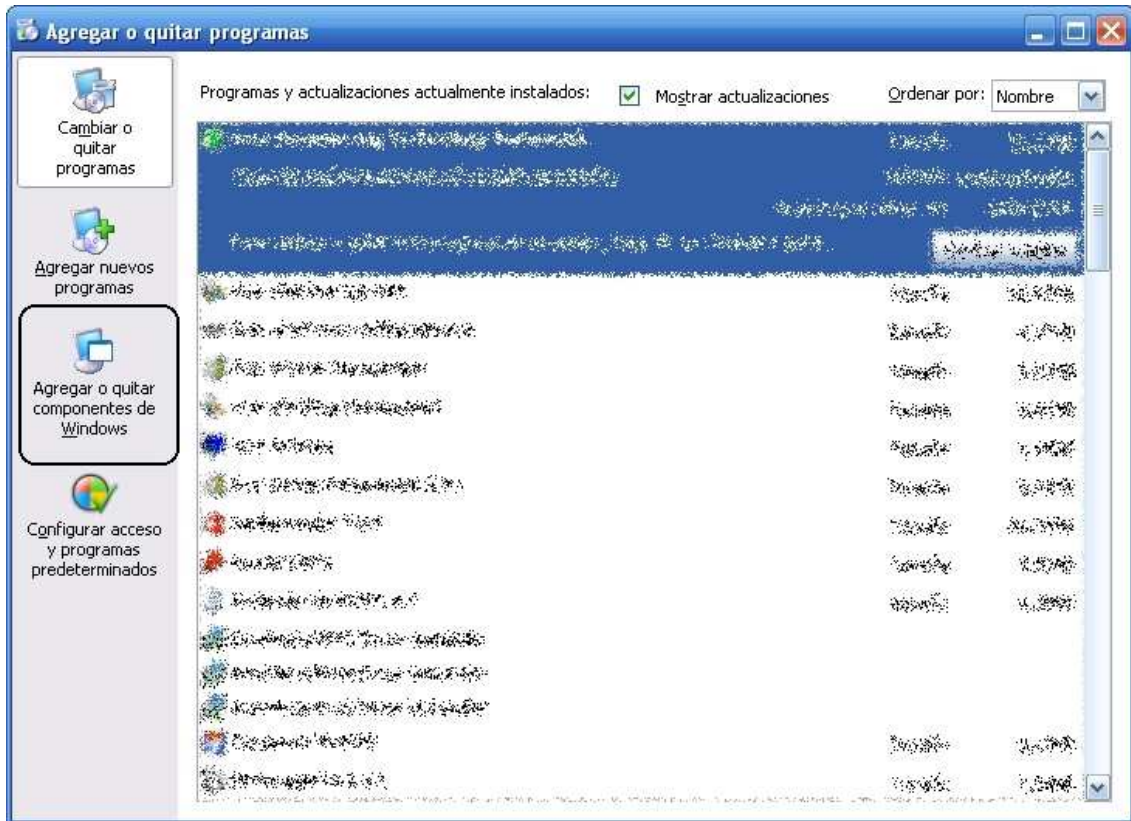


Con esto queda instalado el Visual Studio 2005, después hay que copiar nuestra aplicación a una carpeta dentro de la ruta:

C:\Inetpub\wwwroot

2.2 Instalación y configuración de iis.

Para que el ordenador pueda hacer de servidor es necesario tener el iis, que es un componente de Windows, para su instalación simplemente hay que añadirlo como otro componente de Windows, de la siguiente forma:

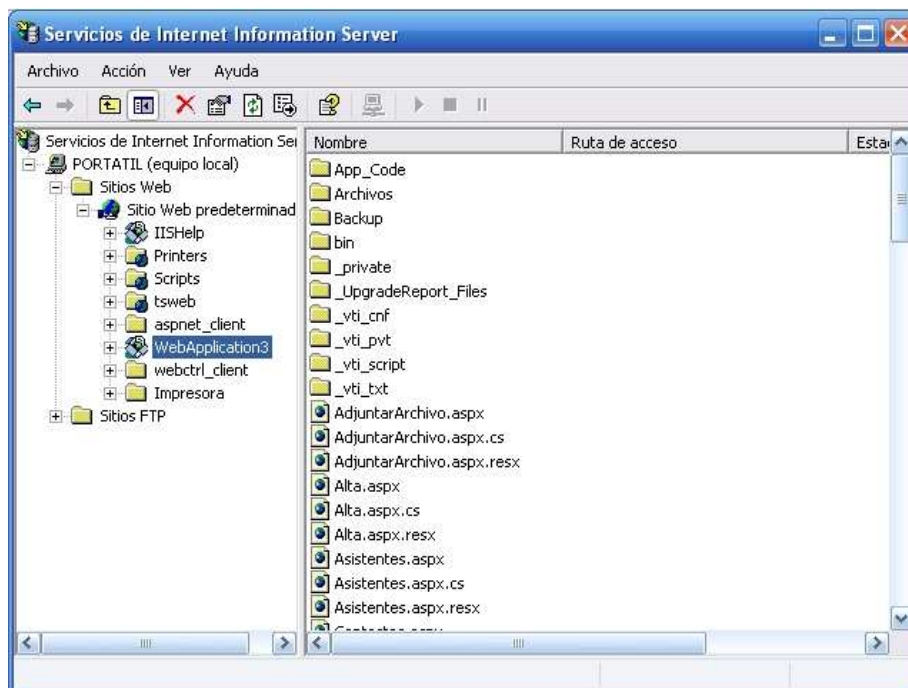




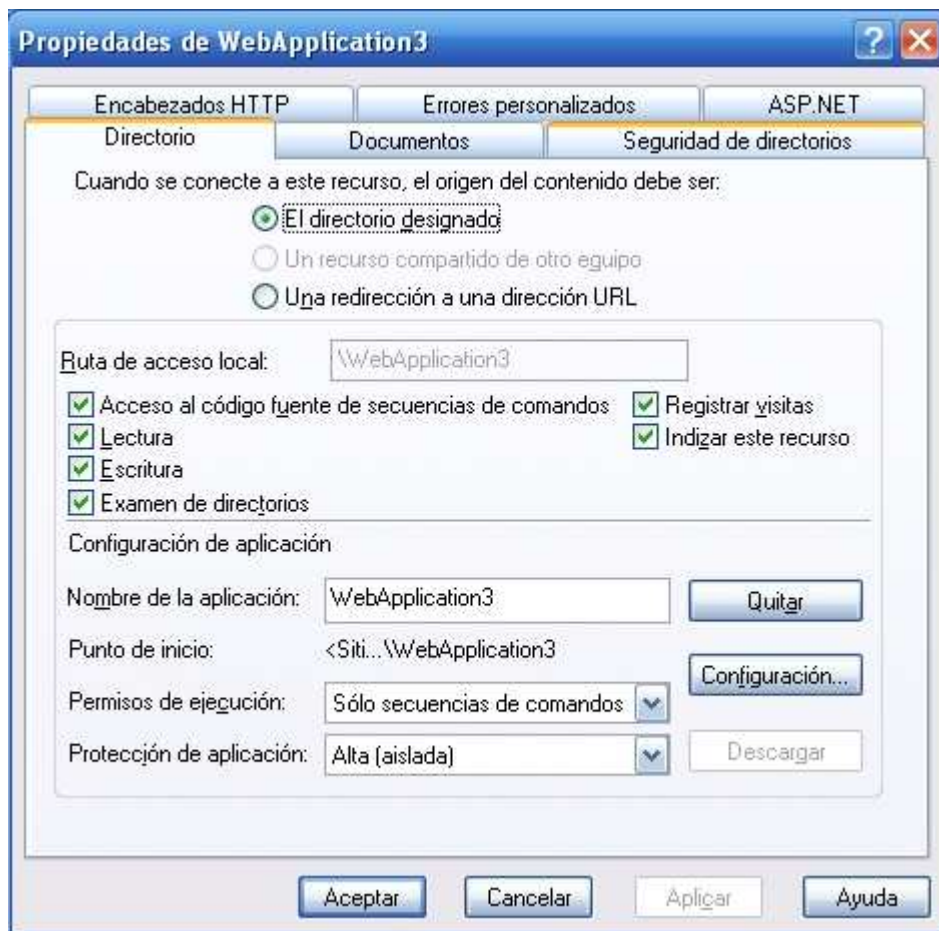
Una vez instalado se procede a configurarlo:

Ruta para llegar hasta el iis:

Panel de control → Herramientas administrativas → Servicios de Internet Information Server



Botón derecho→Propiedades: Debe de aparecer la siguiente información:





El resto de campos deben ser los valores por defecto.

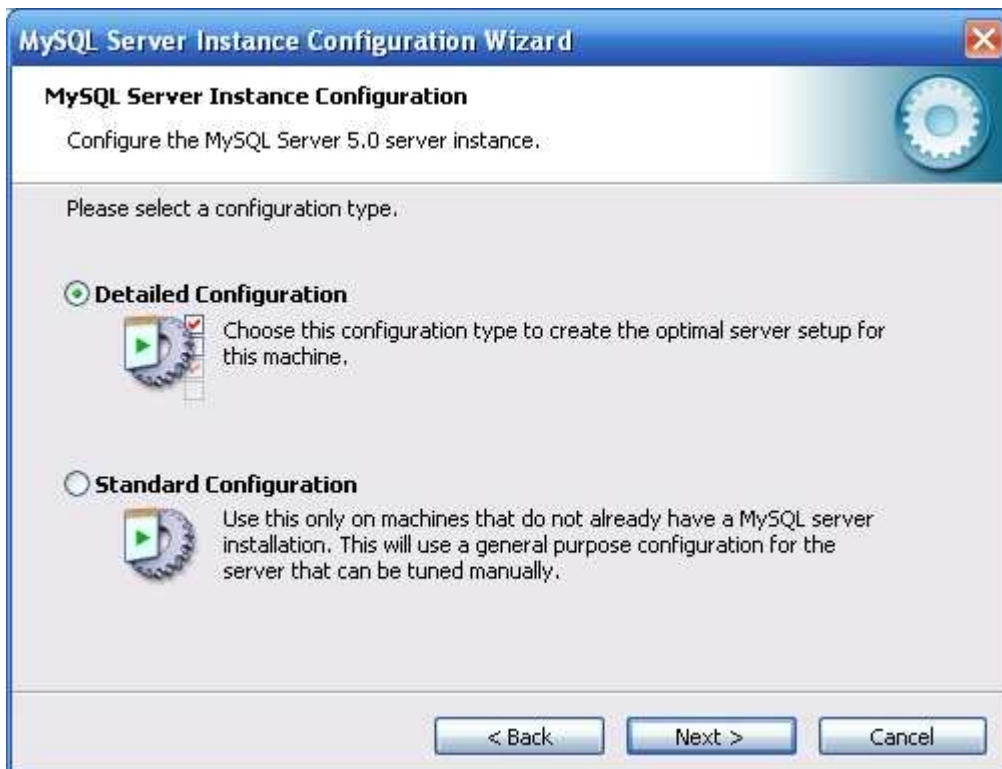
2.3 Instalación, configuración y sincronización de la base de datos.

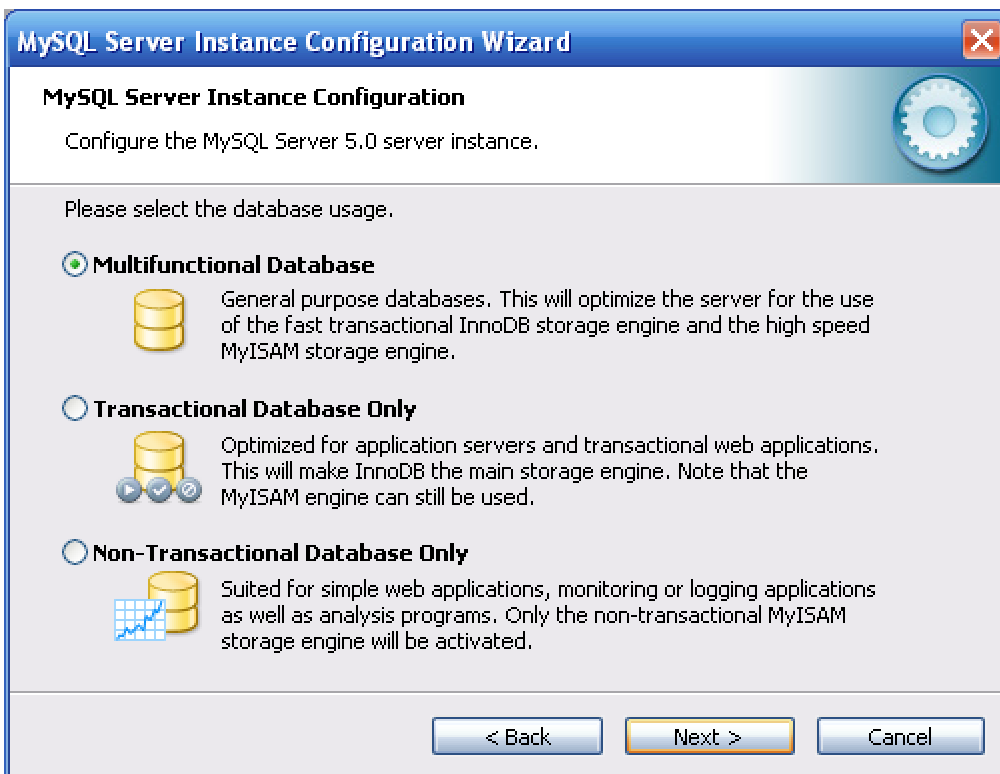
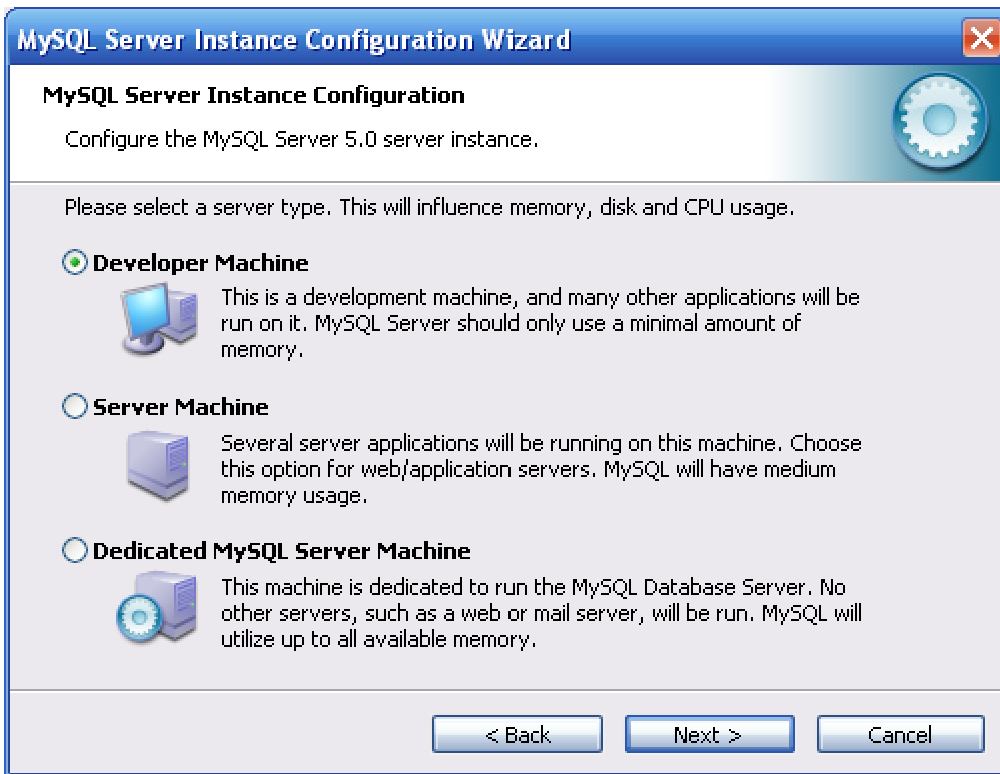
Para la aplicación se ha utilizado la base de datos mysql-5.0.20, con la instalación de la versión completa:

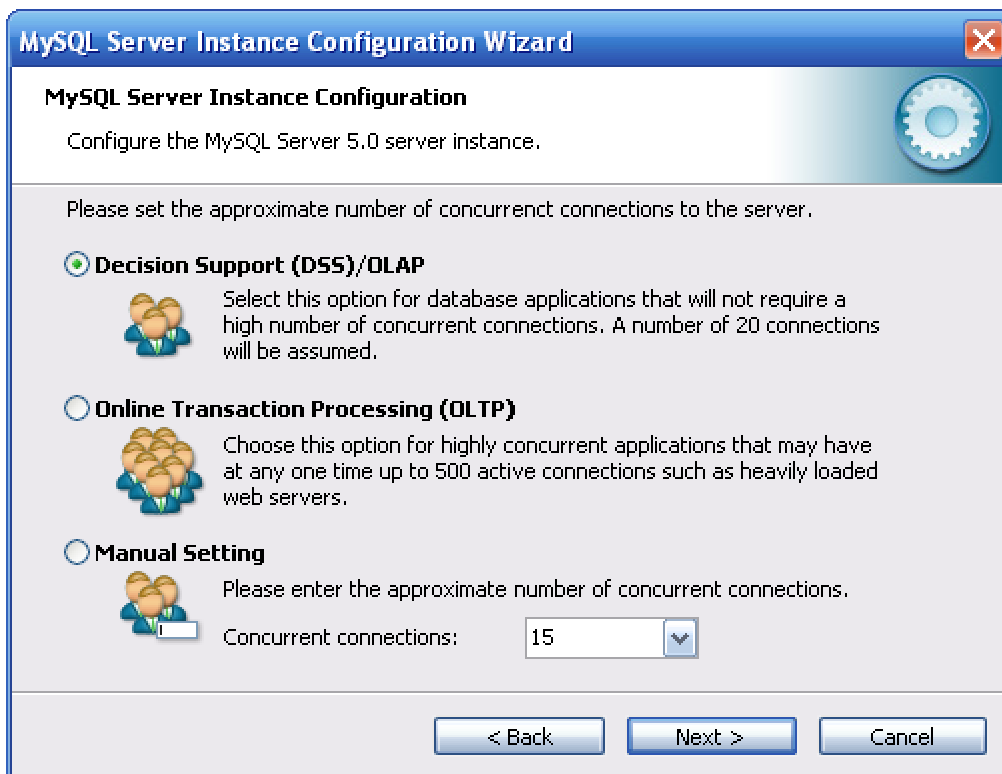
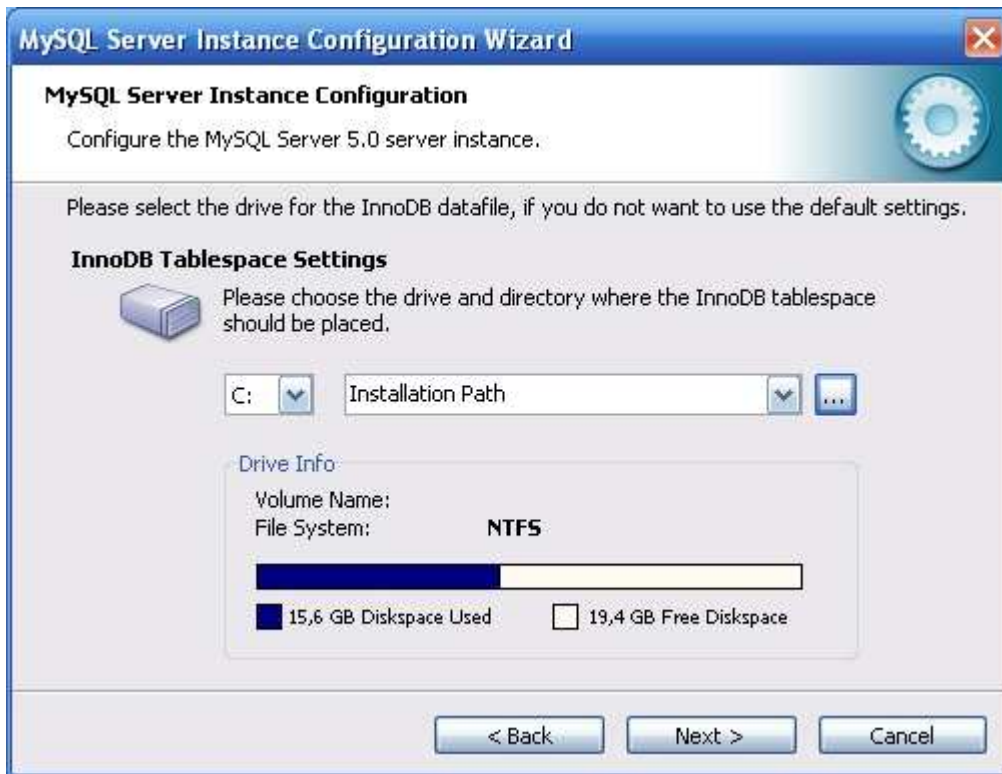


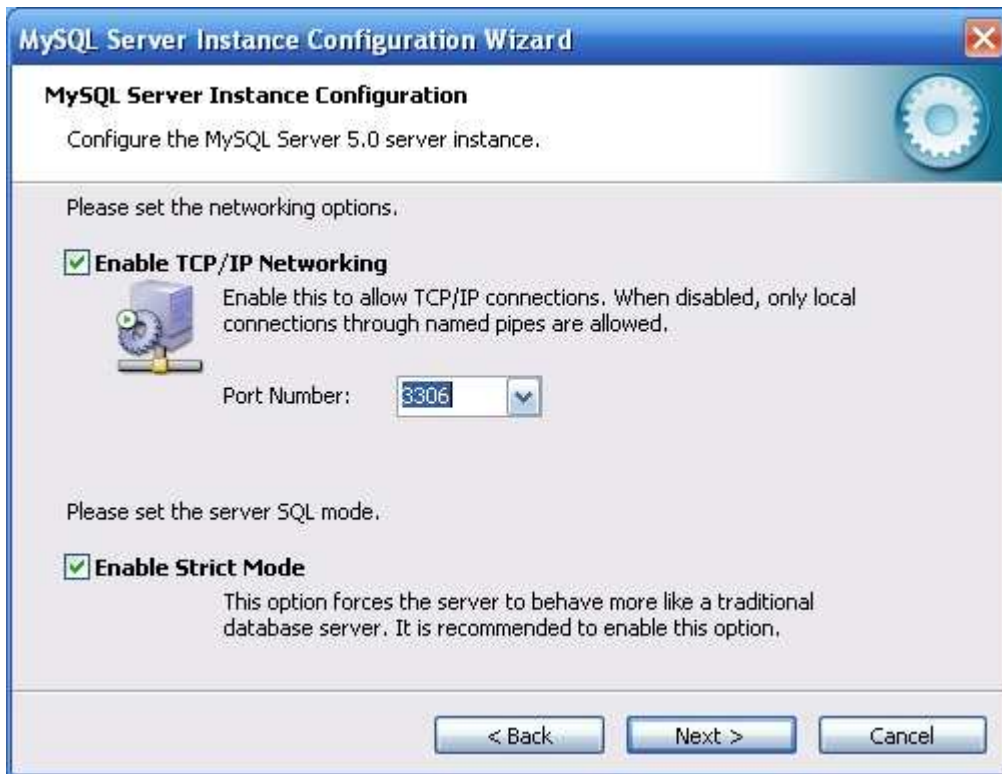


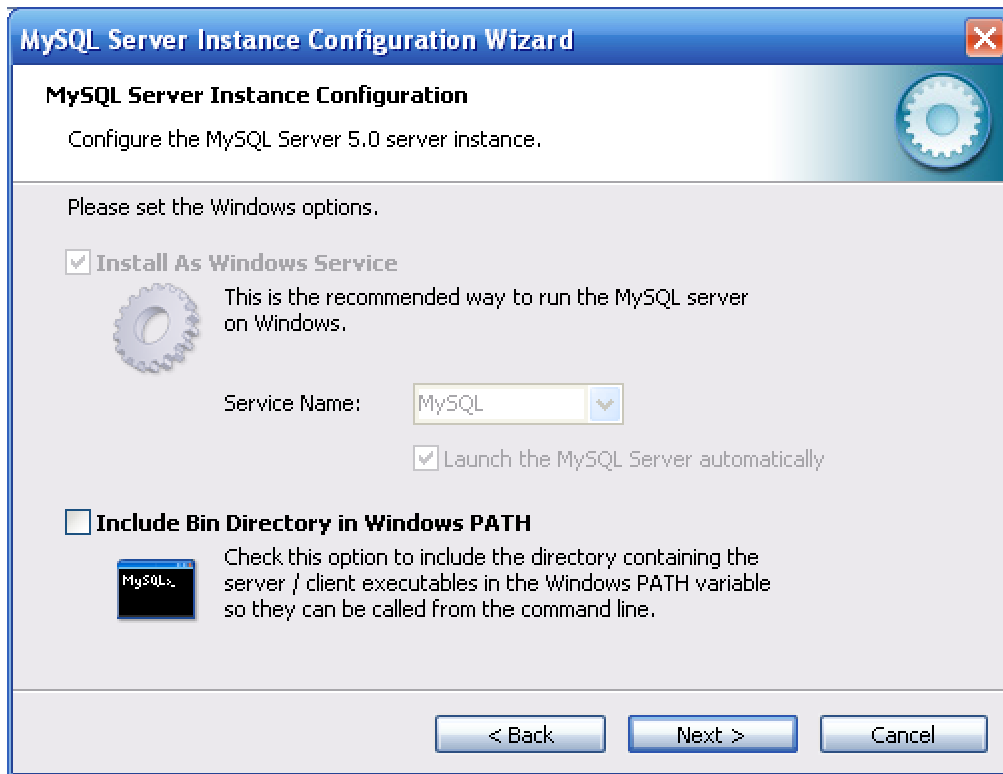
Posteriormente se procede a la configuración de la base de datos:







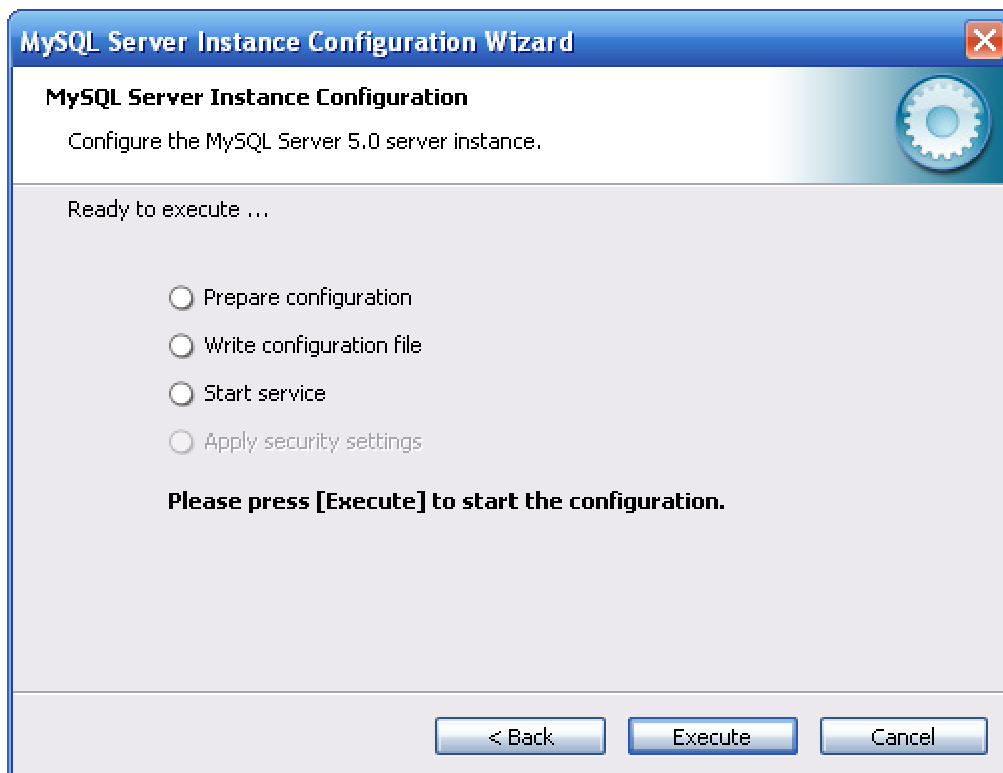




Ahora se introduce la contraseña de root:



Y el último paso de confirmación:



Ahora se ejecuta la aplicación de mysql y se crea la base de datos complumeeting:

```
MySQL Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.0.20-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

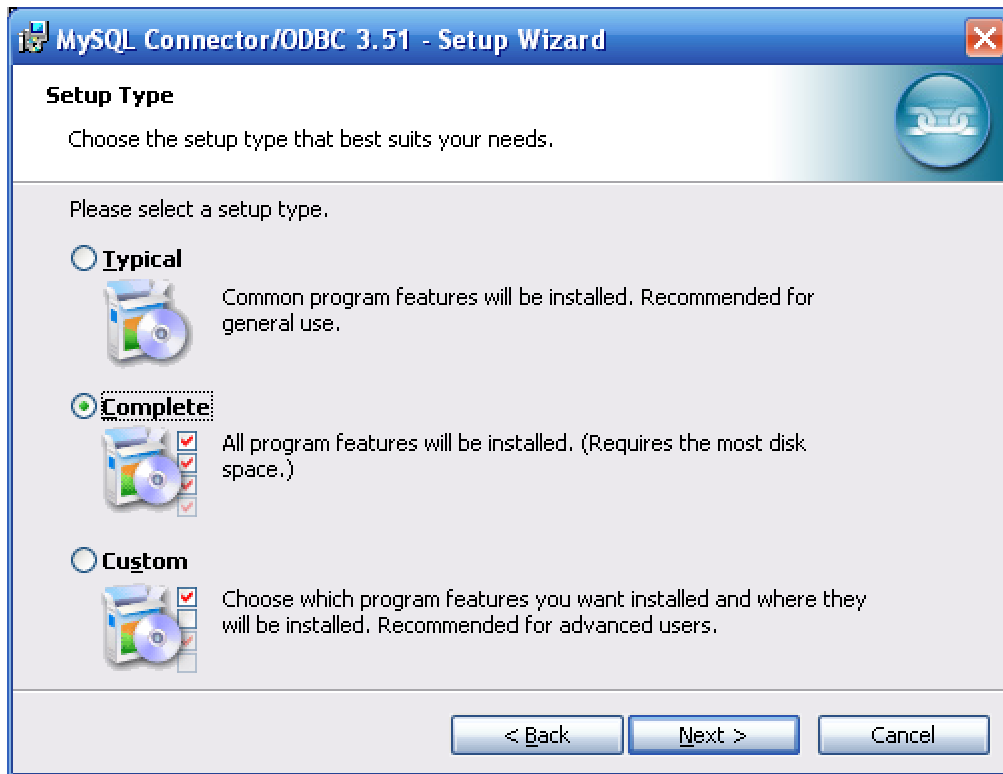
mysql> create database complumeeting;
Query OK, 1 row affected (0.00 sec)

mysql> use complumeeting
Database changed
mysql> _
```

Y a continuación se instala el código de nuestra base de datos.

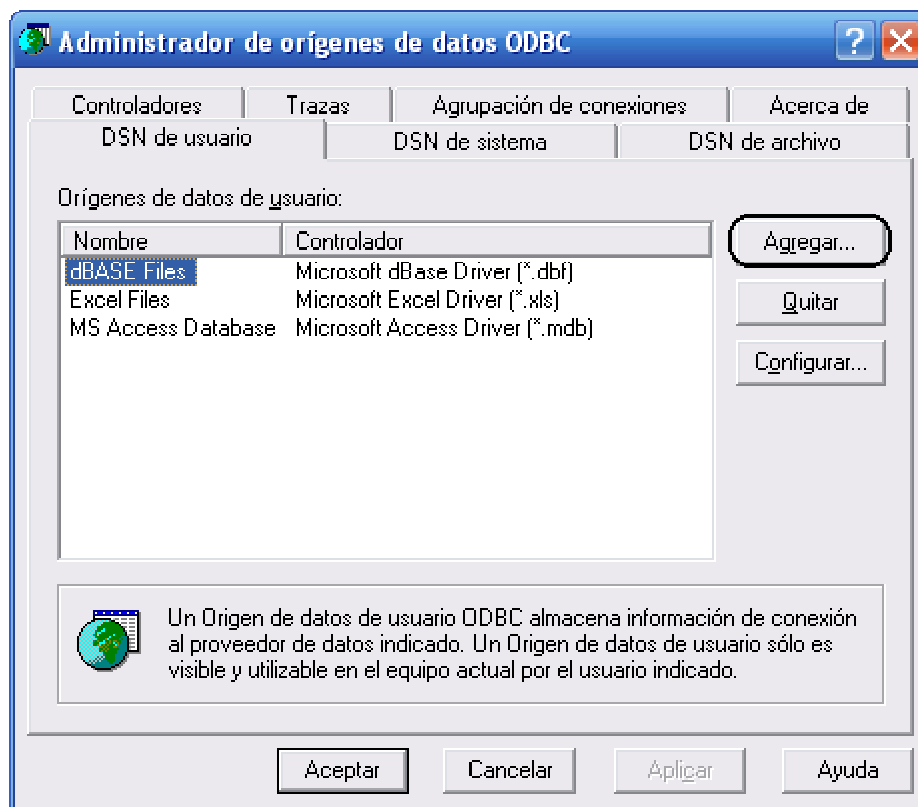
Una vez se ha realizado la configuración de la base de datos sólo falta instar un conector para sincronizar la aplicación con la base de datos recién creada, en este caso el conector utilizado es:

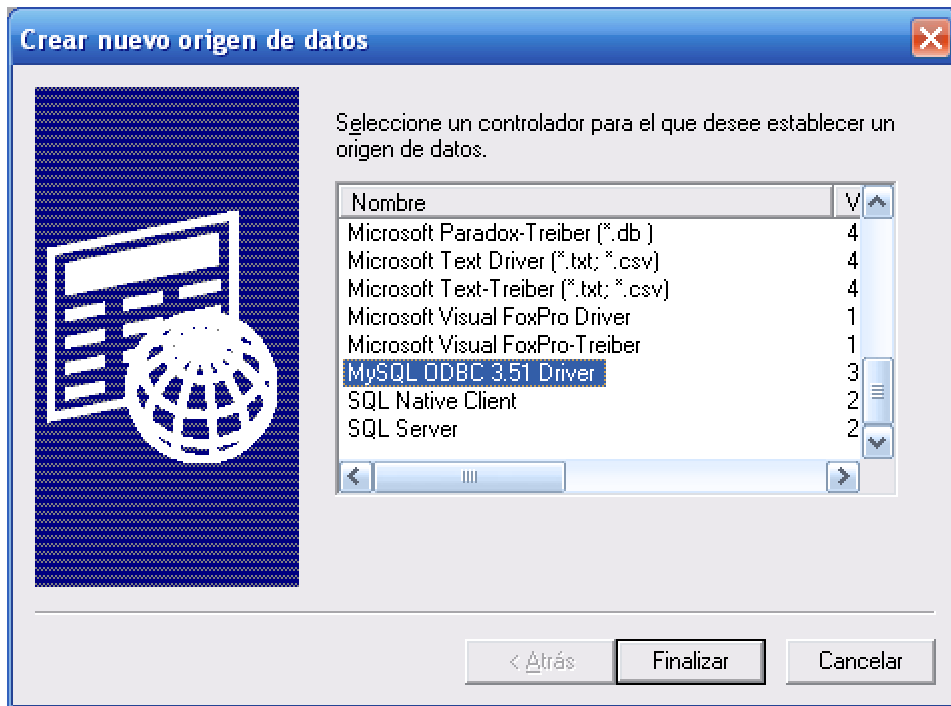
mysql-connector-odbc-3.51.12



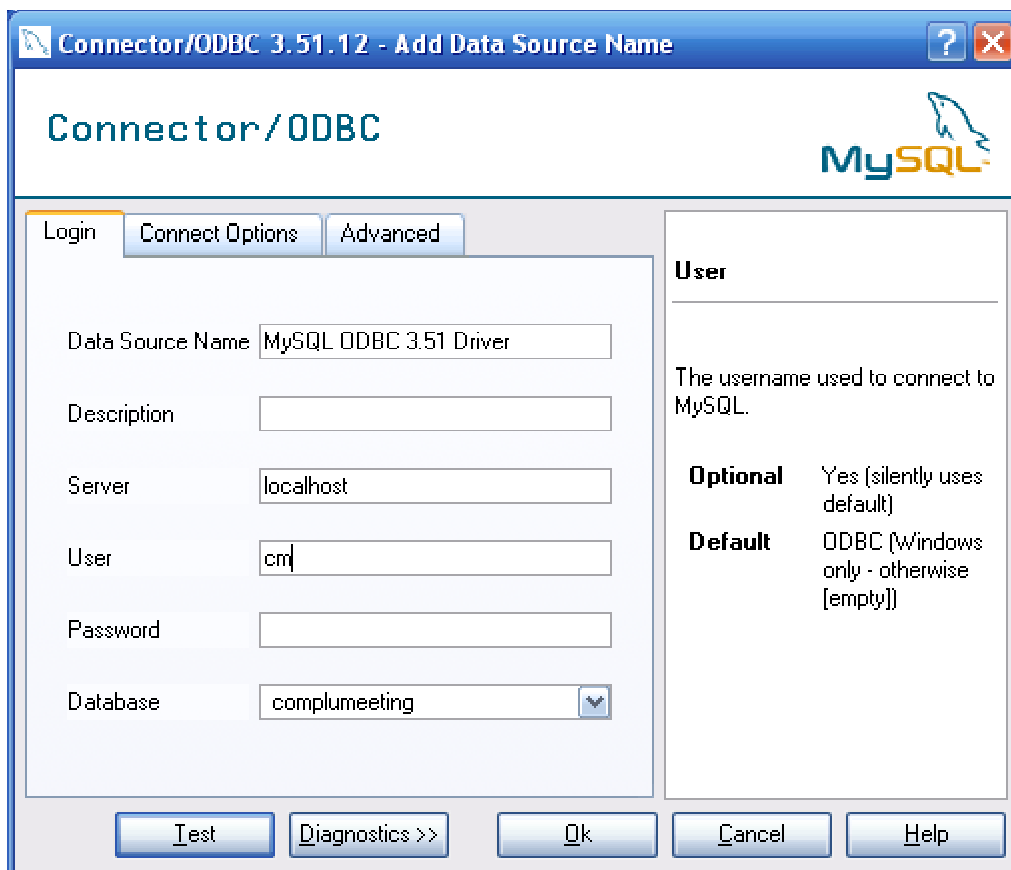
Una vez instalado se configura en: Panel de control→Herramientas administrativas→Orígenes de datos (ODBC)

Primero hay que añadir el conector:





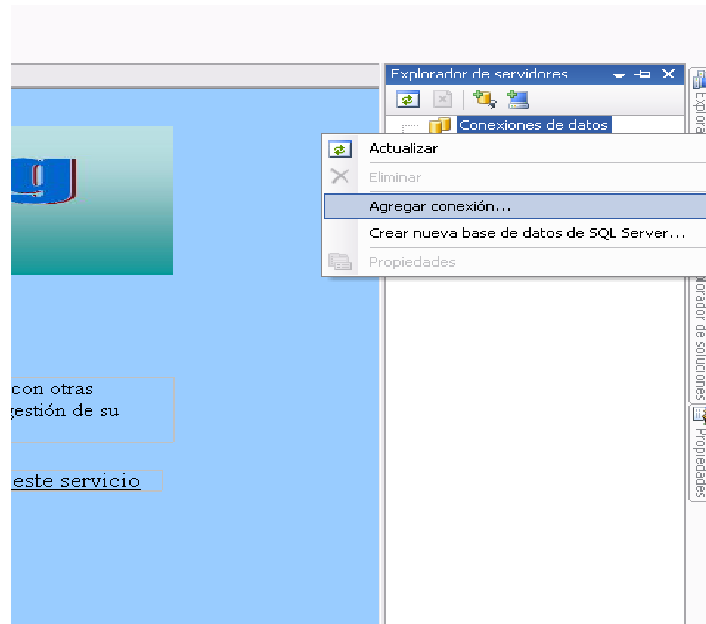
Y posteriormente se configura:



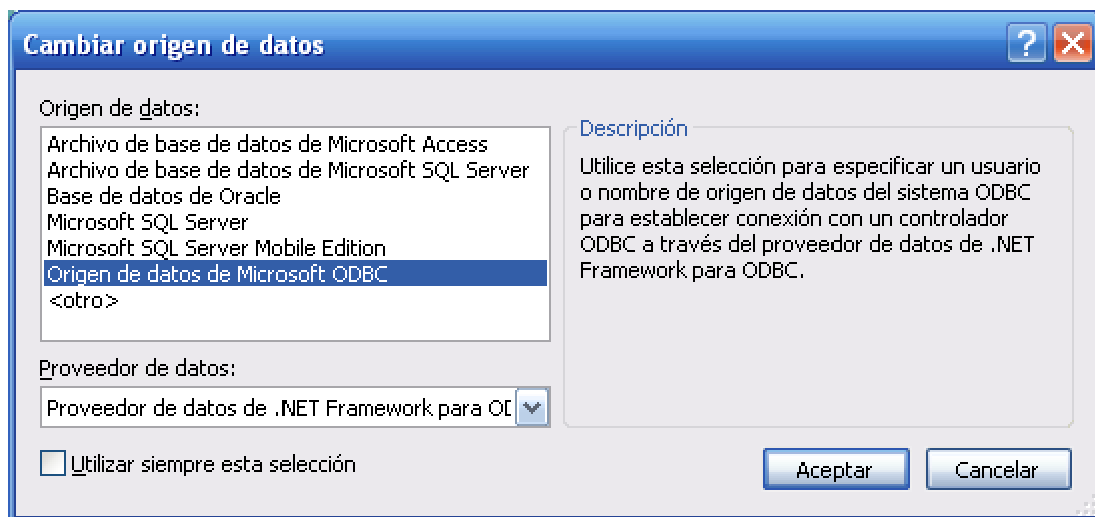
El resto de opciones se deben dejar como vacías.

Posteriormente en el Visual Studio y con la aplicación abierta se finaliza la configuración del conector:

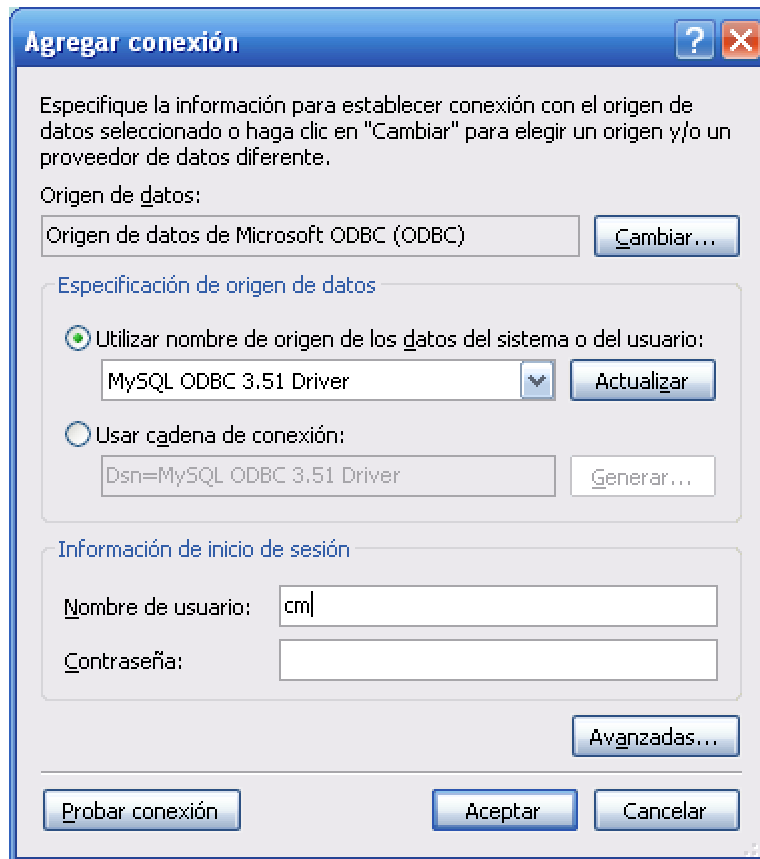
Se abre el explorador de servidores, y sobre conexiones de datos, con el botón derecho se selecciona agregar conexión.



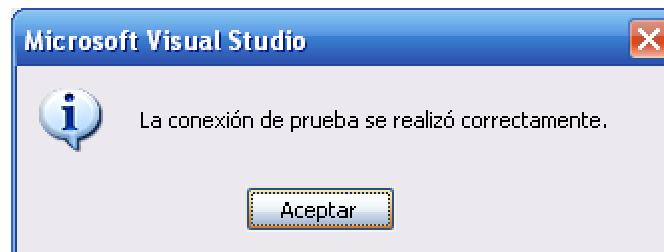
Y posteriormente se selecciona la opción ODBC y continuar:



Y se rellena como sigue:



Y probamos la conexión para finalizar:



3. Manual de usuario

En este punto se detallan todas las funcionalidades del sistema a nivel de usuario. Se detallan todas las páginas que forman la aplicación, describiendo su aspecto y su funcionalidad. En algunos casos, cuando se ha considerado conveniente, se han introducido ejemplos de uso, con el fin de facilitar el entendimiento de cada uno de los procesos.

3.1 Inicio

Esta es la página de acceso a la aplicación, desde aquí podrán acceder al sistema los usuarios registrados, o registrarse los nuevos usuarios.



La página se compone de diferentes apartados que se pasara a detallar a continuación.

Usuario: En este recuadro el usuario debe introducir el identificador de usuario que escogió al darse de alta en la aplicación, o en su defecto alguna de las direcciones de e-mail con las que se registro.

Contraseña: En este cuadro de texto el usuario debe introducir la contraseña con la que se dio de alta. En el caso en el que otro usuario nos haya dado de alta y todavía no hayamos configurado nuestra contraseña, no será necesario rellenar este recuadro.

Breve descripción: Nos ofrece una breve descripción de las funcionalidades que ofrece esta herramienta.

Botón iniciar sesión: La función de este botón es acceder a la pagina principal, tras comprobar que el nombre de usuario y la contraseña sean correctos. En caso de no serlo se mostrara un mensaje indicándolo.

Guardar mi nombre de usuario: El usuario debe marcar esta opción si desea que se recuerde su nombre de usuario la próxima vez que acceda a la aplicación.

Guardar mi nombre de usuario y contraseña: El usuario debe marcar esta opción si desea que se recuerde su nombre de usuario y su contraseña la próxima vez que acceda a la aplicación.

¿Olvidó la contraseña?: Este enlace sirve para recordar la contraseña a los usuarios que la hayan olvidado. Es necesario que se introduzca el identificador de usuario o la dirección de correo electrónico con la que el usuario se dio de alta, tras pulsar el enlace se enviará la contraseña a la dirección de correo asociada al identificador de usuario, o bien a la dirección que se ha introducido.

Botón alta en el servicio: Este enlace redirige a la página de alta, donde los nuevos usuarios podrán darse de alta en el sistema.

3.2 Alta

Desde esta página los nuevos usuarios pueden darse de alta en el sistema para poder empezar a disfrutar de los servicios que ofrece esta herramienta. A su vez esta página servirá para que usuarios ya registrados modifiquen sus datos de alta. En este segundo caso, la página aparecerá rellena con los datos del usuario.

Alta - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección [http://localhost/WebApplication3/\(S\(eyjngw55ern4k3n4vaiy145\)\)/Alta.aspx](http://localhost/WebApplication3/(S(eyjngw55ern4k3n4vaiy145))/Alta.aspx) Ir Vínculos >>

Complu-meeting

Gestor de reuniones

Introduzca los datos necesarios : [Volver](#)

Identificador usuario *

Nombre *

Apellidos *

Contraseña *

Vuelva a escribir la contraseña *

E-mail *

E-mail 2

E-mail 3

Organización

Una contraseña contribuye a evitar el acceso no autorizado a sus datos

Dirección de correo electrónico que le permitirá recibir avisos

Números de teléfono

Personal Trabajo

Movil Fax

Direcciones

Personal

Dirección Código postal

Ciudad País

Provincia

Trabajo

Dirección Código postal

Ciudad País

Provincia

Volver: Este enlace redirige a la página de inicio sin guardar los cambios realizados.

Identificador de usuario: Este identificador servirá como distintivo a cada uno de los usuarios. Se utilizará para poder acceder al sistema. Este campo es obligatorio, y único, es decir, si existen otros usuarios dados de alta con este mismo identificador, se mostrará un mensaje indicando que debe cambiarse.

Nombre: Campo obligatorio, debe indicarse el nombre real del usuario.

Apellidos: Campo obligatorio, debe indicarse los apellidos del usuario.

Contraseña: Campo obligatorio, en este campo debe indicarse la contraseña de seguridad que se utilizará para acceder al sistema.

Vuelva a escribir la contraseña: Campo obligatorio, procedimiento de seguridad para evitar que la contraseña se haya introducido incorrectamente en el cuadro contraseña.

E-mail / E-mail1 / E-mail2: Es obligatorio introducir una dirección de correo electrónico, las otras dos son opcionales. Esta dirección, también puede ser utilizada como identificador del usuario, no pudiendo darse de alta con la misma dirección.

Organización: Campo no obligatorio, en este campo se debe introducir la organización a la que pertenece el usuario.

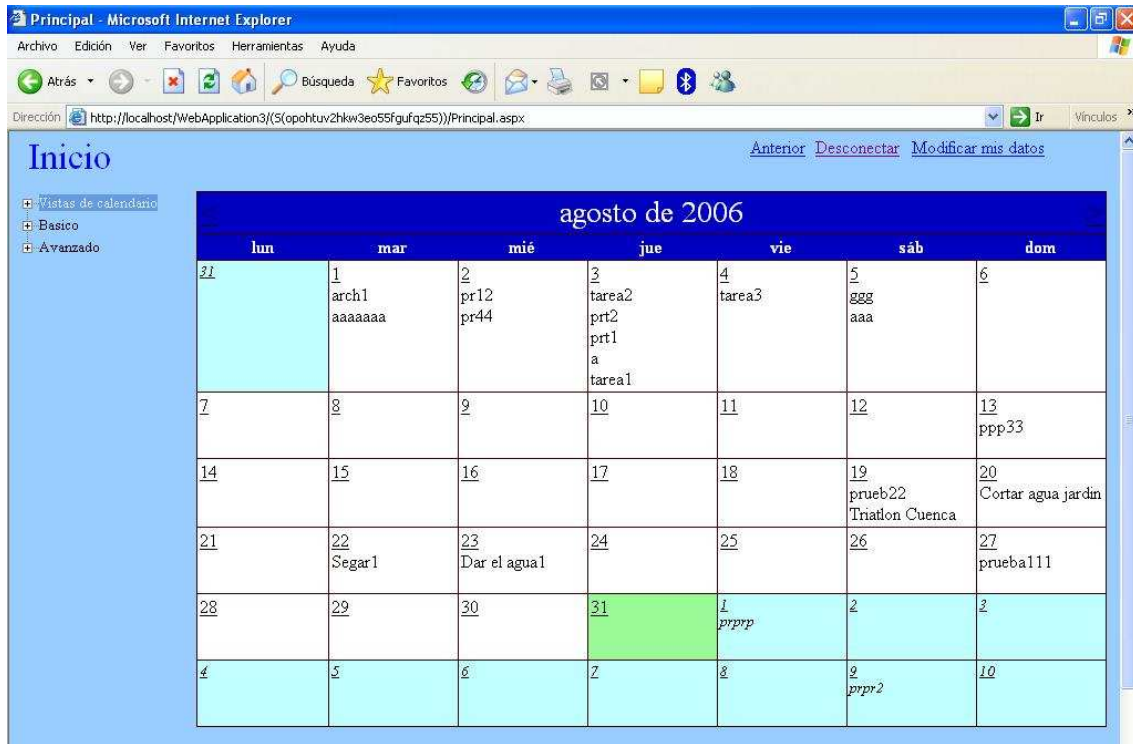
Números de teléfono: Los números de teléfono son opcionales, servirán de posible contacto para otros usuarios de la aplicación.

Direcciones: Las direcciones del usuario también son opcionales, servirán para posible contacto de otros usuarios.

Botón enviar: Este botón sirve para terminar el proceso de alta en el sistema y enviar todos los datos al servidor.

3.3 Principal

Ésta es la página principal de la aplicación, en esta página está centralizada la mayoría de las acciones que se pueden llevar a cabo.



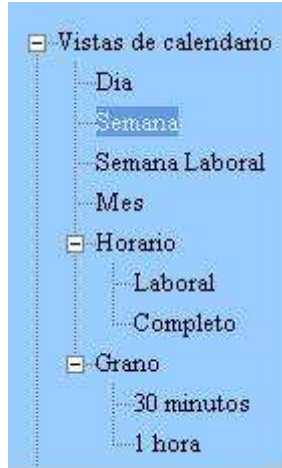
La página se compone de unas partes fijas y otras que van variando según se configure la página. Se va a comenzar por detallar los elementos fijos.

Título de la página: como en toda la aplicación se sitúa en la esquina superior izquierda, y en este caso indica al usuario que se encuentra en el inicio.

Botones de navegación: Se sitúan en la esquina superior derecha y proporcionan enlaces con distintos puntos de la aplicación. Anterior redirige a la página de la cual venimos. Desconectar lleva a la página de acceso a la aplicación. Modificar mis datos lleva a la página de alta donde se podrán modificar los datos.

Menú lateral: El menú lateral se compone de tres nodos principales.

Vistas del calendario: Al desplegar este menú se tiene acceso a diferentes configuraciones del calendario que se sitúa en la parte central. Al desplegar el menú nos encontramos con los siguiente.



Se pueden encontrar 4 tipos de vistas principales, los menús de horario y grano solo aparecen cuando no está seleccionada la vista de mes. Una vez desplegado el menú queda seleccionado en azul oscuro aquella vista que se está mostrando actualmente.

Vista de día: Cuando se selecciona la vista de día en el menú, la zona central quedará en una configuración similar a la siguiente.

<<	1/9/2006	>>
8:00		
9:00		
10:00	Prueba161	
11:00		
12:00		
13:00		
14:00		
15:00		
16:00		
17:00		
18:00		
19:00		
20:00		
21:00		

Cabecera: En esta vista se puede encontrar una cabecera, donde se indica el día que estamos viendo con un formato del estilo de día/mes/año.

Flechas de desplazamiento: Se sitúan a ambos lados de la cabecera en los extremos de la tabla, la flecha de la derecha enviará al siguiente día en el calendario, mientras que la de la izquierda enviará al día anterior.

Horas: Se sitúan en una columna vertical en la zona izquierda de la tabla, cada fila de la tabla estará identificada por una hora de esta columna. El rango de las horas puede variar, desde 8:00 – 21:00 en horario laboral, hasta las 24 horas en horario completo. A su vez se puede seleccionar ver una división de 30 minutos o de horas seleccionando el correspondiente menú de grano.

Zona central: En esta zona se pueden encontrar enlaces con las tareas y reuniones que se situarán en su fecha y hora determinadas, a partir de las cuales se puede tener enlace a sus detalles pulsando sobre ellas.

Vista de semana: Cuando seleccionamos la vista de semana en el menú, la zona central quedará en una configuración similar a la siguiente.

	28/8/2006<-->3/9/2006							
	Lunes 28	Martes 29	Miercoles 30	Jueves 31	Viernes 1	Sabado 2	Domingo 3	
8:00								
9:00								
10:00					Prueba161			
11:00								
12:00								
13:00								
14:00								
15:00								
16:00								
17:00								
18:00								
19:00								
20:00								
21:00								

Cabecera: En esta vista se puede encontrar una cabecera, donde se indica la semana que estamos viendo con un formato del estilo de día/mes/año <--> día/mes/año, donde la fecha de la izquierda indica el comienzo de la semana, mientras que la de la

derecha indica el último día de la semana. Bajo esta cabecera e identificando cada una de las columnas se puede encontrar cada uno de los días de la semana, comenzando en el lunes hasta el domingo. Cada uno de estos días va acompañado por el día del mes que es, con el fin de facilitar al usuario encontrar la fecha adecuada.

Flechas de desplazamiento: Se sitúan a ambos lados de la cabecera en los extremos de la tabla, la flecha de la derecha enviará a la semana siguiente en el calendario, mientras que la de la izquierda enviará a la anterior.

Horas: Se sitúan en una columna vertical en la zona izquierda de la tabla, cada fila de la tabla estará identificada por una hora de esta columna. El rango de las horas puede variar, desde 8:00 – 21:00 en horario laboral, hasta las 24 horas en horario completo. A su vez se puede seleccionar ver una división de 30 minutos o de horas seleccionando el correspondiente menú de grano.

Zona central: En esta zona se puede encontrar enlaces con las tareas y reuniones que se situarán en su fecha y hora determinadas, a partir de las cuales se tiene enlace a sus detalles pulsando sobre ellas.

Vista de semana laboral: Cuando se selecciona la vista de semana laboral en el menú, la zona central quedará en una configuración similar a la siguiente.

28/8/2006<-->3/9/2006					
	Lunes 28	Martes 29	Miércoles 30	Jueves 31	Viernes 1
8:00					
9:00					
10:00					Prueba161
11:00					
12:00					
13:00					
14:00					
15:00					
16:00					
17:00					
18:00					
19:00					
20:00					
21:00					

Cabecera: En esta vista se puede encontrar una cabecera, donde se indica la semana que se ve con un formato del estilo de día/mes/año <--> día/mes/año, donde la fecha de la izquierda indica el comienzo de la semana, mientras que la de la derecha indica el último día de la semana. Bajo esta cabecera e identificando cada una de las columnas podemos encontrar cada uno de los días de la semana, comenzando en el lunes hasta el viernes. Cada uno de estos días va acompañado por el día del mes que es, con el fin de facilitar al usuario encontrar la fecha adecuada.

Flechas de desplazamiento: Se sitúan a ambos lados de la cabecera en los extremos de la tabla, la flecha de la derecha enviará a la semana siguiente en el calendario, mientras que la de la izquierda enviará a la anterior.

Horas: Se sitúan en una columna vertical en la zona izquierda de la tabla, cada fila de la tabla estará identificada por una hora de esta columna. El rango de las horas puede variar, desde 8:00 – 21:00 en horario laboral, hasta las 24 horas en horario completo. A su vez se puede seleccionar ver una división de 30 minutos o de horas seleccionando el correspondiente menú de grano.

Zona central: En esta zona se pueden encontrar enlaces con las tareas y reuniones que se situarán en su fecha y hora determinadas, a partir de las cuales podremos tener enlace a sus detalles pulsando sobre ellas.

Vista de mes: Cuando se selecciona la vista de mes en el menú, la zona central quedará en una configuración similar a la siguiente.

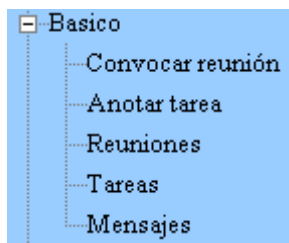
septiembre de 2006						
lun	mar	mié	jue	vie	sáb	dom
<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>	1 prpr Prueba161	<u>2</u>	<u>3</u>
<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u> prpr2	<u>10</u>
<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>
<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>
<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>	<u>1</u>
<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>

Cabecera: En la cabecera de este control se puede observar el mes y el año que se esta visualizando. Bajo esta cabecera e identificando cada una de las columnas podemos encontrar cada uno de los días de la semana, comenzando en el lunes hasta el domingo.

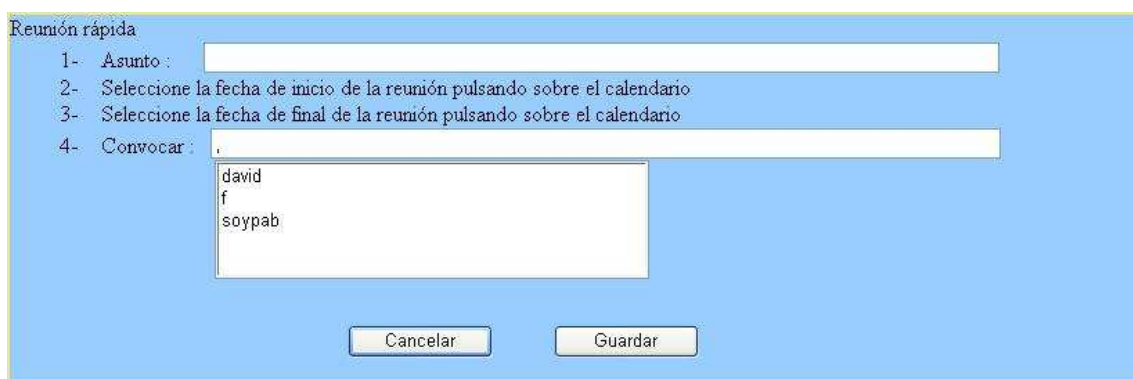
Flechas de desplazamiento: Se sitúan a ambos lados de la cabecera en los extremos de la tabla, la flecha de la derecha enviará al mes siguiente en el calendario, mientras que la de la izquierda enviará al anterior.

Zona central: En esta zona se puede encontrar una serie de celdas, cada una de ellas corresponde a un día del mes. Cada uno de los números que representan a los días da acceso a la vista de día pulsando sobre él. Estas celdas también contienen los asuntos de las tareas y las reuniones de cada uno de los días. En este caso no se puede acceder directamente a la tarea o reunión, tendremos que pulsar sobre el día y una vez en la vista de día acceder. Las celdas en color blanco corresponden al mes escrito en la cabecera, mientras que las celdas azul claro corresponden a los últimos días del mes anterior y a los primeros del siguiente. La celda en verde corresponde al día en el que estamos.

Básico: Permite realizar acciones básicas rápidamente sin la necesidad de desplazarnos a otras páginas de la aplicación. Este apartado está pensado para facilitar el uso de la aplicación y hacerla más accesible al usuario, se centralizan las principales tareas y las más utilizadas por los usuarios.



Convocar reunión: Pulsando sobre este apartado del menú se puede convocar una reunión de forma rápida y sencilla.

A screenshot of a dialog box titled "Reunión rápida". It contains four numbered steps: 1- "Asunto:" followed by a text input field; 2- "Seleccione la fecha de inicio de la reunión pulsando sobre el calendario"; 3- "Seleccione la fecha de final de la reunión pulsando sobre el calendario"; 4- "Convocar:" followed by a dropdown menu. The dropdown menu is open, showing three options: "david", "f", and "soypab". At the bottom of the dialog are two buttons: "Cancelar" and "Guardar".

Asunto: En este cuadro de texto el usuario tendrá que introducir un texto que describa la reunión que está creando. Este campo es necesario para poder crear la reunión, en caso de que este vacío, no se podrá crear la reunión. El asunto se mostrará en distintos puntos de la aplicación y junto con la fecha y la hora (si tiene) será el identificador de la reunión.

Seleccionar la fecha de inicio: Tras haber introducido el asunto se debe seleccionar la fecha y la hora que se desea como fecha de inicio de la reunión. Para ello se debe pulsar en el día y la hora que se desee en cualquiera de las distintas vistas del calendario. Una vez se haya pulsado aparecerá el asunto en esa posición y el recuadro se volverá de color amarillo. Si se desea cambiar la fecha de inicio antes de guardar se puede pulsar sobre otro día y hora tras haber introducido la fecha de fin de la reunión.

Seleccionar la fecha de fin: Tras haber introducido la fecha de inicio de la reunión, se debe seleccionar la fecha de final de la reunión, para ello se debe pulsar en el día y la hora que se desee en cualquiera de las distintas vistas del calendario. Tras haber seleccionado la fecha de final, todo el rango de horas que comprende la reunión quedará señalado en amarillo. Si lo desea puede cambiar la fecha de la reunión seleccionando una nueva fecha de inicio.

Convocar: En este cuadro de texto se indican los distintos usuarios que van a ser convocados para la reunión, separados por comas. Estos usuarios se pueden seleccionar del cuadro de usuarios, o bien escribirlos a mano. En el caso de que haya usuarios repetidos en la lista, estas repeticiones serán eliminadas en la siguiente recarga de la página.

Lista de usuarios: En esta lista se encuentran todos los usuarios del sistema que están dados de alta como contactos personales. Para convocarlos a la reunión solo es necesario pulsar sobre ellos y se añadirán automáticamente a la lista de usuarios convocados.

Botón cancelar: Este botón sirve para cancelar la creación de la reunión; la reunión no se guardará y el recuadro de convocar reunión desaparecerá.

Botón guardar: Este botón no se encuentra activo hasta que no se introduce el asunto y se seleccionan la fecha y la hora de inicio y de fin, una vez activo significa que se puede guardar la reunión. Una vez guardada la reunión, esta será notificada a los usuarios asistentes mediante un mensaje de la aplicación y un correo electrónico.

Anotar tarea: Pulsando sobre este apartado del menú se puede anotar una tarea de forma rápida y sencilla.

Asunto: En este cuadro de texto el usuario tendrá que introducir un texto que describa la tarea que está creando. Este campo es necesario para poder crear la tarea, en caso de que este vacío, no se podrá anotar la tarea. El asunto se mostrará en distintos puntos de la aplicación y junto con la fecha y la hora(si tiene) será el identificador de la tarea.

Seleccionar la fecha: Tras haber introducido el asunto se debe seleccionar la fecha y la hora que se desea como fecha de vencimiento. Para ello se debe pulsar en el día y la hora que se desee en cualquiera de las distintas vistas del calendario. Una vez se haya pulsado aparecerá el asunto en esa posición. Si se desea cambiar la fecha antes de guardar se puede pulsar sobre otro día y hora.

Botón cancelar: Este botón sirve para cancelar la creación de la tarea; la tarea no se guardará y el recuadro de tarea rápida desaparecerá. Para crear otra tarea se debe pulsar de nuevo sobre anotar tarea.

Botón guardar: Este botón no se encuentra activo hasta que no se introduce el asunto y se seleccionan la fecha y la hora, una vez activo significa que podemos guardar la tarea. Una vez guardada la tarea, el recuadro vuelve a su estado inicial para poder anotar otra tarea.

Reuniones: Pulsando sobre esta opción se puede acceder a un listado de las reuniones confirmadas, ordenadas por fecha de inicio y cuya fecha de inicio no es anterior a la actual.

Reuniones confirmadas				X
Asunto	Fecha de comienzo	Usuario que convoca		
r3	06/09/2006 8:00:00	soypab1	Cancelar	^
r2	08/09/2006 9:00:00	soypab1	Cancelar	
r5	12/09/2006 10:00:00	soypab1	Cancelar	
r6	12/09/2006 12:00:00	soypab1	Cancelar	
reunion1	13/09/2006 9:00:00	soypab1	Cancelar	
r33	14/09/2006 10:00:00	soypab1	Cancelar	
r9	15/09/2006 10:00:00	soypab1	Cancelar	
	20/09/2006 9:00:00	soypab1	Cancelar	v

Cabecera: En la cabecera de la tabla, en el extremo izquierdo, se encuentra un texto que indica que se esta visualizando la tabla de reuniones confirmadas, mientras que en el extremo derecho se puede encontrar un botón con el que se cierra la ventana.

Asunto: En esta columna se sitúa el texto que el usuario ha introducido como asunto de la reunión. Los asuntos de cada reunión tienen asociado un botón cuya función es abrir una ventana emergente en la que se puede consultar y modificar los detalles de la reunión.

Fecha de comienzo: En esta columna se sitúa la fecha y la hora de comienzo de cada una de las reuniones.

Usuario que convoca: En esta columna se sitúa el identificador del usuario que convoca dicha reunión.

Cancelar: En la cuarta columna de la tabla, cada reunión tiene asociado un botón cancelar. Si el usuario que cancela la reunión, es el usuario convocador de la reunión, la reunión quedara cancelada y será informado a todos los asistentes mediante un mensaje de la aplicación y un correo electrónico. Si por el contrario no es el usuario que la convoca sino que es un asistente, el asistente cancelará su participación en la reunión siendo este hecho informado al creador de la reunión mediante un mensaje de la aplicación y un correo electrónico.

Tareas: Pulsando sobre esta opción se puede acceder a un listado de las tareas no vencidas, o sin fecha de vencimiento, ordenadas por fecha de vencimiento.

Tareas			
Asunto	Fecha vencimiento	Fecha alarma	
sinfv2	no	no	eliminar
sinfv5	no	no	eliminar
prueba17	01/09/2006 9:00:00	01/09/2006 8:00:00	eliminar
Prueba161	01/09/2006 10:00:00	no	eliminar
prueba162	02/09/2006 10:00:00	no	eliminar
prt1	03/09/2006 9:00:00	no	eliminar
prpr2	09/09/2006 8:00:00	no	eliminar

Cabecera: En la cabecera de la tabla se encuentra un texto que indica que se esta visualizando la tabla de tareas en el extremo izquierdo, mientras que en el extremo derecho se puede encontrar un botón con el que se cierra la ventana de tareas.

Asunto: En esta columna se sitúa el texto que el usuario ha introducido como asunto de la tarea. Los asuntos de cada tarea tienen asociado un botón cuya función es abrir una ventana

emergente en la que se puede consultar y modificar los detalles de la tarea.

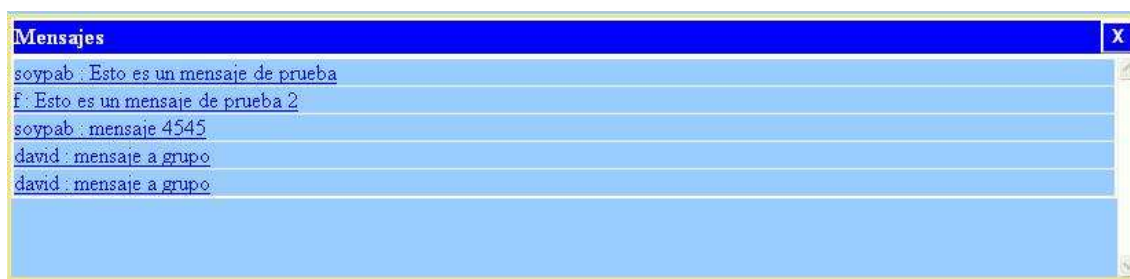
Fecha de vencimiento: En esta columna se sitúa la fecha de vencimiento de cada una de las tareas, y el texto “no” en caso de no tener fecha de vencimiento.

Alarma: En esta columna se sitúan la fecha y la hora de la alarma de cada una de las tareas, y el texto “no” en caso de no tener alarma puesta.

Eliminar: En la cuarta columna de la tabla, cada tarea tiene asociado un botón eliminar, que permitirá eliminar las tareas después de haber dado la confirmación de que se desea eliminarla en una ventana emergente que aparecerá.

La zona de datos de la tabla dispone de un scrollbar vertical que aparecerá automáticamente cuando el número de filas de la tabla sobrepase el tamaño de la página, con el fin de que no se pierda de vista nunca el resto de controles de la página y a la vez moviendo el scrollbar se pueda acceder a todos los datos.

Mensajes: Pulsando sobre esta opción se puede acceder a un listado de los mensajes del usuario, estos mensajes pueden provenir tanto de otros usuarios como del sistema, en forma de avisos y notificaciones.



Cabecera: En la cabecera de la tabla se encuentra un texto que indica que se está visualizando la tabla de mensajes en el extremo izquierdo, mientras que en el extremo derecho se puede encontrar un botón con el que se cierra la ventana de mensajes.

Mensajes: Cada una de las filas corresponde a un mensaje nuevo, se puede encontrar primero el remitente del mensaje, seguido por el mensaje en sí. En el caso de ser un mensaje del sistema el es remitente será “Sistema”.

Avanzado: Agrupa las funcionalidades avanzadas de la aplicación. Desde este menú se puede acceder a toda la funcionalidad que ofrece la aplicación a través de redirecciones a otras páginas.



Nueva tarea: Desde este enlace se accede a la página de nueva tarea donde se puede crear una nueva tarea con todas las opciones que ofrece la aplicación.

Lista de tareas: Desde este enlace se accede a la lista de tareas que permite tener acceso a un listado de las tareas que se han ido introduciendo en el sistema, con distintas posibilidades de manipulación.

Nueva Reunión: Desde este enlace se accede a la página de nueva reunión, donde se puede crear una nueva reunión con todas las opciones que ofrece la aplicación.

Lista de reuniones: Desde este enlace se accede a la lista de reuniones que permite tener acceso a un listado de las reuniones que se han ido introduciendo en el sistema, con distintas posibilidades de manipulación.

Usuarios y grupos: Desde este enlace se accede a la página de usuarios y grupos, donde se puede manipular nuestras listas tanto de usuarios como de grupos.

Nuevo mensaje: Desde este enlace se accede a la página de nuevo mensaje, desde donde se puede enviar mensajes a diferentes usuarios y grupos.

Lista de mensajes: Desde este enlace se accede a la lista de mensajes que nos permite tener acceso a un listado de todos los mensajes recibidos, con distintas posibilidades de manipulación.

3.4 Nueva tarea

Esta página permitirá introducir una nueva tarea en el sistema.

Nueva tarea

Guarde una nueva tarea, coloque la fecha de vencimiento, y el asunto, tiene la posibilidad de que se le avise por correo electrónico, añada los archivos necesarios para esta tarea.

Guardar Cancelar

Asunto

Fecha de vencimiento

agosto de 2006

	l	m	m	j	v	s	d
31	1	2	3	4	5	6	
7	8	9	10	11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
28	29	30	31	1	2	3	
4	5	6	7	8	9	10	

08 00

Aviso 10 min antes

Aviso por correo electrónico

[Archivos adjuntos](#)

Periodicidad

Notas

La página se compone de distintos apartados:

Título de la página: como en toda la aplicación se sitúa en la esquina superior izquierda, y en este caso indica al usuario que se encuentra en nueva tarea.

Breve descripción : Acompaña al título de la página y proporciona al usuario una breve descripción de la funcionalidad de la página.

Botón guardar : La función de este botón consiste en guardar en el sistema una nueva tarea bajo los parámetros que ha introducido el usuario en la página. La operación puede fallar en el caso de que no se haya introducido nada en el campo de asunto, o bien que este marcado el tic de fecha de vencimiento y no haya ninguna fecha seleccionada, en este caso se mostrará un mensaje al usuario con el error para que lo corrija. En caso de no fallar la operación, se cerrará la ventana emergente en la que se había abierto la página.

Botón cancelar: La función de este botón es cancelar la creación de la nueva tarea, tras pedir confirmación, si ésta es positiva, procederá a cerrarse la ventana emergente.

Asunto: En este cuadro de texto el usuario tendrá que introducir un texto que describa la tarea que está creando. Este campo es necesario para poder crear la tarea, en caso de que esté vacío, se pedirá que se rellene para poder

continuar. El asunto se mostrará en distintos puntos de la aplicación y junto con la fecha y la hora(si tiene) será el identificador de la tarea.

Fecha de vencimiento: La fecha de vencimiento de una tarea se considera el momento, o el último momento en el que se puede llevar a cabo. En caso de no estar marcado el tic, significará que la tarea no tiene una fecha en concreto, con lo que no se podrán marcar fecha, hora, periodicidad o avisos.

Calendario: Inicialmente no posee ninguna fecha seleccionada, el usuario tendrá que seleccionar aquí la fecha de vencimiento. En caso de estar marcado el cuadro fecha de vencimiento, es necesario introducir una fecha. Se mostrará un mensaje avisando al usuario de que le faltan por introducir datos.

Hora: La hora se puede seleccionar en las dos listas desplegables que se encuentran a la derecha del calendario. La primera representa las horas, y es posible seleccionar cualquiera de las 24 horas del día. La segunda lista desplegable, representa los minutos, en este caso solo se podrá seleccionar las horas en punto o las medias horas. Inicialmente estará seleccionada las 08:00, las 8 de la mañana.

Aviso: Esta opción permite que la aplicación avise de la tarea, seleccionando en el cuadro desplegable cuanto tiempo antes deseamos que se avise. La aplicación avisará mediante el envío de un mensaje, que podremos ver en la página principal, en el cuadro de mensajes, y en la página de mensajes. Inicialmente esta opción no estará seleccionada.

Aviso por correo electrónico: Esta opción permite ser avisado por la aplicación de la tarea mediante un mensaje de correo electrónico. Este aviso se enviara a todos los correos electrónicos que el usuario tenga dados de alta en la aplicación. El aviso por correo se enviara en el mismo momento que el aviso en el caso de que este seleccionado, y se enviara en el momento en el que venza la tarea en el caso de no estar seleccionado el aviso.

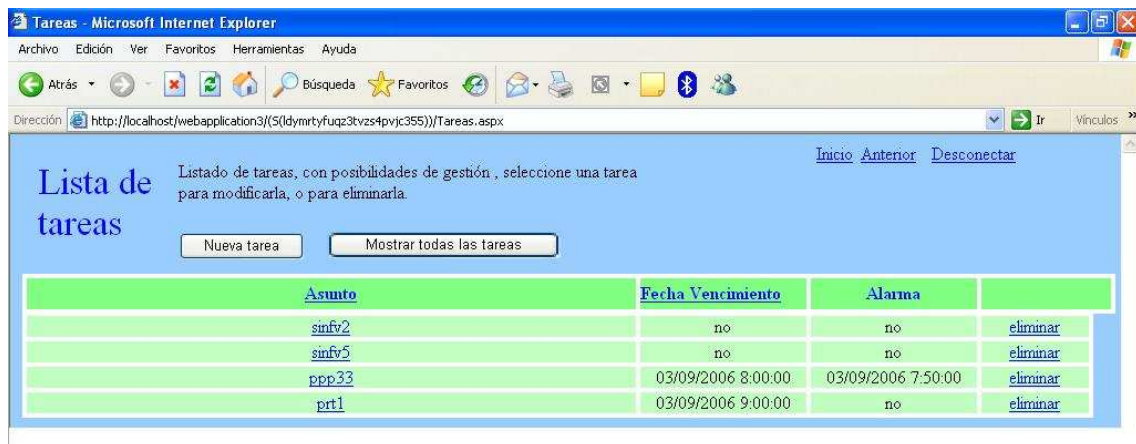
Botón periodicidad: Permite dirigirse a la página de periodicidad, donde se podrá introducir una periodicidad para la tarea. Solo se encuentra visible en el caso de que este marcada la fecha de vencimiento. Para poder acceder a marcar la periodicidad, se debe haber marcado alguna fecha de vencimiento, en caso contrario se pedirá que se marque para poder continuar.

Notas: En este cuadro de texto se puede introducir todas las anotaciones referentes a la tarea que interese recordar. Posee barras de desplazamiento verticales en el caso de que el texto introducido sobrepase el tamaño de cuadro.

Archivos adjuntos: Pulsando sobre este enlace accedemos a una página donde podemos agregar archivos adjuntos a la tarea. Debajo de este enlace aparece la tabla de los archivos que están ya adjuntos.

3.5 Lista de tareas

Esta página nos permite tener acceso a un listado de las tareas que se han ido introduciendo en el sistema, con distintas posibilidades de manipulación que se detallan a continuación.



La página se compone de distintos apartados:

Título de la página: como en toda la aplicación se sitúa en la esquina superior izquierda, y en este caso indica al usuario que se encuentra en la lista de tareas.

Breve descripción : Acompaña al título de la página y proporciona al usuario una breve descripción de la funcionalidad de la página

Botones de navegación: Se sitúan en la esquina superior derecha y proporcionan enlaces con distintos puntos de la aplicación. Inicio redirige a la página de principal de la aplicación. Anterior a la página de la cual venimos. Desconectar lleva a la página de acceso a la aplicación.

Botón nueva tarea: Nos proporciona una ventana emergente en la que se puede dar de alta una nueva tarea en el sistema.

Botón mostrar todas las tareas / mostrar tareas no vencidas: Este botón sirve para configurar la tabla de datos. Cuando el texto de este botón es mostrar todas las tareas, situación que ocurre la primera vez que se accede, en la tabla de datos se están mostrando la tareas cuya fecha de vencimiento aun no ha pasado o no tienen. Por el contrario si el texto es mostrar tareas no vencidas, se están mostrando todas las tareas que tenemos en el sistema. El efecto de pulsar el botón es pasar de un estado a otro.

Tabla de datos: La tabla de datos tiene dos partes diferenciadas, la cabecera, en verde oscuro y la zona de datos, de verde más claro. En la cabecera se tienen los textos indicativos del contenido de las columnas que se sitúan justo debajo:

Asunto: En esta columna se sitúa el texto que el usuario ha introducido como asunto de la tarea. Este botón tiene además la función de colocar las tareas de la tabla por orden alfabético según esta columna. Los asuntos de cada tarea tienen asociado un botón cuya función es abrir una ventana emergente en la que se puede consultar y modificar los detalles de la tarea.

Fecha de vencimiento: En esta columna se sitúa la fecha de vencimiento de cada una de las tareas, y el texto “no” en caso de no tener fecha de vencimiento. Además el botón tiene la función de ordenar las tareas de la tabla según la fecha de vencimiento.

Alarma: En esta columna se sitúan la fecha y la hora de la alarma de cada una de las tareas, y el texto “no” en caso de no tener puesta alarma.

Eliminar: En la cuarta columna de la tabla, sin cabecera, cada tarea tiene asociado un botón eliminar, que permitirá eliminar las tareas después de haber dado la confirmación de que se desea eliminarla en una ventana emergente que aparecerá.

La zona de datos de la tabla dispone de un scrollbar vertical que aparecerá automáticamente cuando el número de filas de la tabla sobrepase el tamaño de la página, con el fin de que no se pierda de vista nunca el resto de controles de la página y a la vez moviendo el scrollbar se pueda acceder a todos los datos.

Características importantes: El estado inicial de la página se muestran las tareas no vencidas ordenadas por fecha de vencimiento. Las modificaciones del estado de la página son guardadas a lo largo de la sesión, con el fin de que cada vez que se vuelva encontremos la configuración que habíamos dejado anteriormente.

3.6 Detalles tarea

La configuración y el funcionamiento es muy similar al de la página nueva tarea, salvo unas pequeñas diferencias que a continuación se detallan.

Cuando se accede a esta página todos los controles tienen los datos de la tarea cuyos detalles se desean ver. Estos datos se cargan en el momento en el que se carga la página. En esta página podremos modificar los datos de la tarea y guardar las modificaciones en el sistema en el caso de pulsar el botón Guardar cambios. En el caso de cancelar, la tarea quedará como estaba perdiendo todos los cambios realizados.

3.7 Nueva reunión

Esta página permite convocar una nueva reunión, para completar la reunión tendremos que seguir siete pasos, dos de ellos opcionales.

Nueva Reunión Guardar Cancelar

Primer paso: Asunto:

Segundo paso: Descripción:

Tercer paso: Reunión con fecha fija
 Busqueda de la mejor fecha en un rango de fechas

Rango en el que buscar una fecha para la reunión: Rango en el que buscar una hora para la reunión

Inicio: Hora 08 00

Finalización: Hora 08 00

Restricciones Periodicidad

Cuarto paso: Duración de la reunión: Minutos

Quinto paso: Usuarios convocados ---->

Ver compatibilidades

Sexto paso: [Archivos adjuntos](#)
(Opcional)

Séptimo paso: Configurar avisos:
(Opcional) Aviso en la aplicación 10 min antes
 Aviso por correo electrónico

Título de la página: como en toda la aplicación se sitúa en la esquina superior izquierda, y en este caso indica al usuario que se encuentra en nueva reunión.

Botón guardar : La función de este botón es guardar la nueva reunión, mostrara un mensaje si se han omitido alguno de los pasos no opcionales

Botón cancelar: La función de este botón es cancelar la creación de la nueva reunión, tras pedir confirmación, si ésta es positiva, procederá a cerrarse la ventana emergente.

Asunto: En este cuadro de texto el usuario tendrá que introducir un texto que describa la reunión que está creando. Este campo es necesario para poder crear la reunión, en caso de que este vacío, no se podrá crear la reunión. El asunto se mostrará en distintos puntos de la aplicación y junto con la fecha y la hora(si tiene) será el identificador de la reunión.

Descripción: En este cuadro de texto se debe introducir una breve descripción de la reunión que se esta convocando con el fin de facilitar cierta información a los asistentes.

Búsqueda de fecha: En este tercer paso, la opción preseleccionada es la de búsqueda de la mejor fecha en un rango de fechas, en esta opción se debe seleccionar un rango de fechas y un rango de horas en el que es posible la celebración de la reunión, para tras conocer las preferencias de los usuarios, escoger la mejor fecha para la celebración de la reunión. En caso de seleccionar esta opción, la reunión no quedará confirmada hasta que el creador fije una fecha definitiva. En el caso que se seleccione reunión con fecha fija, el usuario debe seleccionar una fecha y una hora para la celebración de la reunión y esta pasará a reuniones confirmadas.

Botón restricciones: La función de este botón es dar acceso a una pagina donde el usuario puede meter sus restricciones. Estas restricciones están asociadas al usuario y no a una reunión en concreto, es decir, que las restricciones que un usuario introduzca para esta reunión, serán tenidas en cuenta para futuras reuniones.

Botón periodicidad: Permite dirigirse a la página de periodicidad, donde se podrá introducir una periodicidad para la reunión. La periodicidad se tomará desde que se fije la fecha de la reunión, esta se repetirá según la periodicidad introducida.

Duración de la reunión: En este cuarto paso el usuario debe introducir el numero de minutos, horas o días que va a durar la reunión, seleccionando en la lista desplegable la opción que deseemos.

Usuarios convocados: Si se pulsa sobre este enlace se accede a la pagina donde se puede seleccionar los usuarios y los grupos que van a ser convocados a la reunión. En el cuadro inmediatamente a la derecha se pueden observar que usuarios y grupos han sido convocados.

Botón ver compatibilidades: Pulsando sobre este botón se accede a una página donde se pueden consultar las restricciones de los usuarios convocados a la reunión, con el fin de facilitar el encontrar la fecha mas adecuada.

Archivos adjuntos: Pulsando sobre este enlace accedemos a una página donde podemos agregar archivos adjuntos a la reunión. Debajo de este enlace aparece la tabla de los archivos que están ya adjuntos.

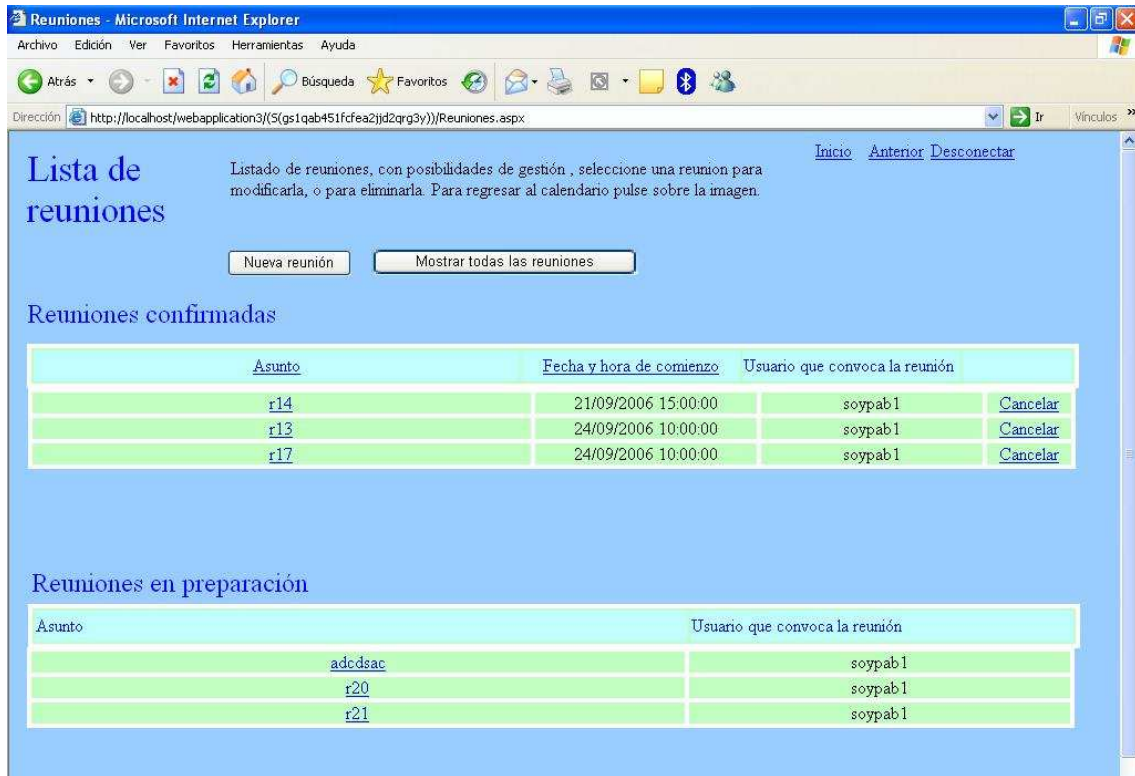
Configurar avisos:

Aviso: Esta opción permite que la aplicación avise de la reunión, seleccionando en el cuadro desplegable cuanto tiempo antes deseamos que se avise. La aplicación avisará a todos los usuarios que estén convocados mediante el envío de un mensaje, que se podrá ver en la página principal, en el cuadro de mensajes, y en la página de mensajes. Inicialmente esta opción no estará seleccionada.

Aviso por correo electrónico: Esta opción permite ser avisado por la aplicación de la tarea mediante un mensaje de correo electrónico. Este aviso se enviara a los correos electrónicos de todos los usuarios dados de alta en la reunión. El aviso por correo se enviara en el mismo momento que el aviso en el caso de que este seleccionado, y se enviara en el momento de inicio de la reunión en el caso de no estar seleccionado el aviso.

3.8 Lista de reuniones

Esta página nos permite tener acceso al listado de reuniones confirmadas, y al listado de reuniones en preparación.



Título de la página: como en toda la aplicación se sitúa en la esquina superior izquierda, y en este caso indica al usuario que se encuentra en la lista de reuniones.

Breve descripción : Acompaña al título de la página y proporciona al usuario una breve descripción de la funcionalidad de la página

Botones de navegación: Se sitúan en la esquina superior derecha y proporcionan enlaces con distintos puntos de la aplicación. Inicio redirige a la página de principal de la aplicación. Anterior a la página de la cual venimos. Desconectar lleva a la página de acceso a la aplicación.

Botón nueva reunión: Nos proporciona una ventana emergente en la que se puede convocar una nueva reunión.

Botón mostrar todas las reuniones / mostrar reuniones no vencidas: Este botón sirve para configurar la tabla de reuniones confirmadas. Cuando el texto de este botón es mostrar todas las reuniones, situación que ocurre la primera vez que se accede, en la tabla de reuniones confirmadas se están mostrando las reuniones cuya fecha de inicio aun no ha pasado. Por el contrario si el texto es mostrar reuniones no vencidas, se están mostrando todas las reuniones que tenemos en el sistema. El efecto de pulsar el botón es pasar de un estado a otro.

Tabla de reuniones confirmadas: La tabla de datos tiene dos partes diferenciadas, la cabecera, en azul y la zona de datos en verde. En la cabecera se tienen los textos indicativos del contenido de las columnas que se sitúan justo debajo:

Asunto: En esta columna se sitúa el texto que el usuario creador ha introducido como asunto de la reunión. Este botón tiene además la función de colocar las reuniones de la tabla por orden alfabético según esta columna. Los asuntos de cada reunión tienen asociado un botón cuya función es abrir una ventana emergente en la que se puede consultar y modificar los detalles de la reunión.

Fecha y hora de comienzo: En esta columna se sitúa la fecha de comienzo de cada una de las reuniones. Además el botón tiene la función de ordenar las reuniones de la tabla según la fecha de inicio. Inicialmente las reuniones de esta tabla están ordenadas según la fecha de inicio.

Usuario que convoca: En esta columna se sitúa el identificador del usuario que convoca dicha reunión.

Cancelar: En la cuarta columna de la tabla, cada reunión tiene asociado un botón cancelar. Si el usuario que cancela la reunión, es el usuario convocador de la reunión, la reunión quedará cancelada y será informado a todos los asistentes mediante un mensaje de la aplicación y un correo electrónico. Si por el contrario no es el usuario que la convoca sino que es un asistente, el asistente cancelará su participación en la reunión siendo este hecho informado al creador de la reunión mediante un mensaje de la aplicación y un correo electrónico.

Tabla de reuniones en preparación: La tabla de datos tiene dos partes diferenciadas, la cabecera, en azul y la zona de datos en verde. En la cabecera se tienen los textos indicativos del contenido de las columnas que se sitúan justo debajo:

Asunto: En esta columna se sitúa el texto que el usuario creador ha introducido como asunto de la reunión. Los asuntos de cada reunión tienen asociado un botón cuya función es abrir una ventana emergente en la que se puede consultar y modificar los detalles de la reunión. Las reuniones de esta tabla están ordenadas según esta columna.

Usuario que convoca: En esta columna se sitúa el identificador del usuario que convoca dicha reunión.

Las zonas de datos de la tabla dispone de un scrollbar vertical que aparecerá automáticamente cuando el número de filas de la tabla sobrepase el tamaño de la página, con el fin de que no se pierda de vista nunca el resto de controles de la página y a la vez moviendo el scrollbar se pueda acceder a todos los datos.

Características importantes: El estado inicial de la página se muestran las tareas no vencidas ordenadas por fecha de vencimiento. Las modificaciones del estado de la página son guardadas a lo largo de la sesión, con el fin de que cada vez que se vuelva encontremos la configuración que habíamos dejado anteriormente.

3.9 Detalles de reunión

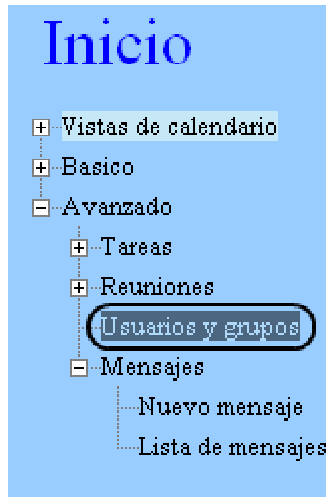
La configuración de esta página es similar a la de nueva reunión, pero en este caso aparecerán los campos rellenos con los datos de la reunión correspondiente. Existen dos configuraciones de esta página, una en el caso de que acceda a los detalles el creador de la reunión, y otra en la que accede un usuario convocado a la reunión.

En la primera configuración, el usuario creador podrá modificar cualquier campo de la reunión, siendo estos cambios avisados por mensaje en la aplicación y por correo electrónico a todos los usuarios convocados. Además, en caso de no estar confirmada la reunión, se podrá confirmar la fecha definitiva.

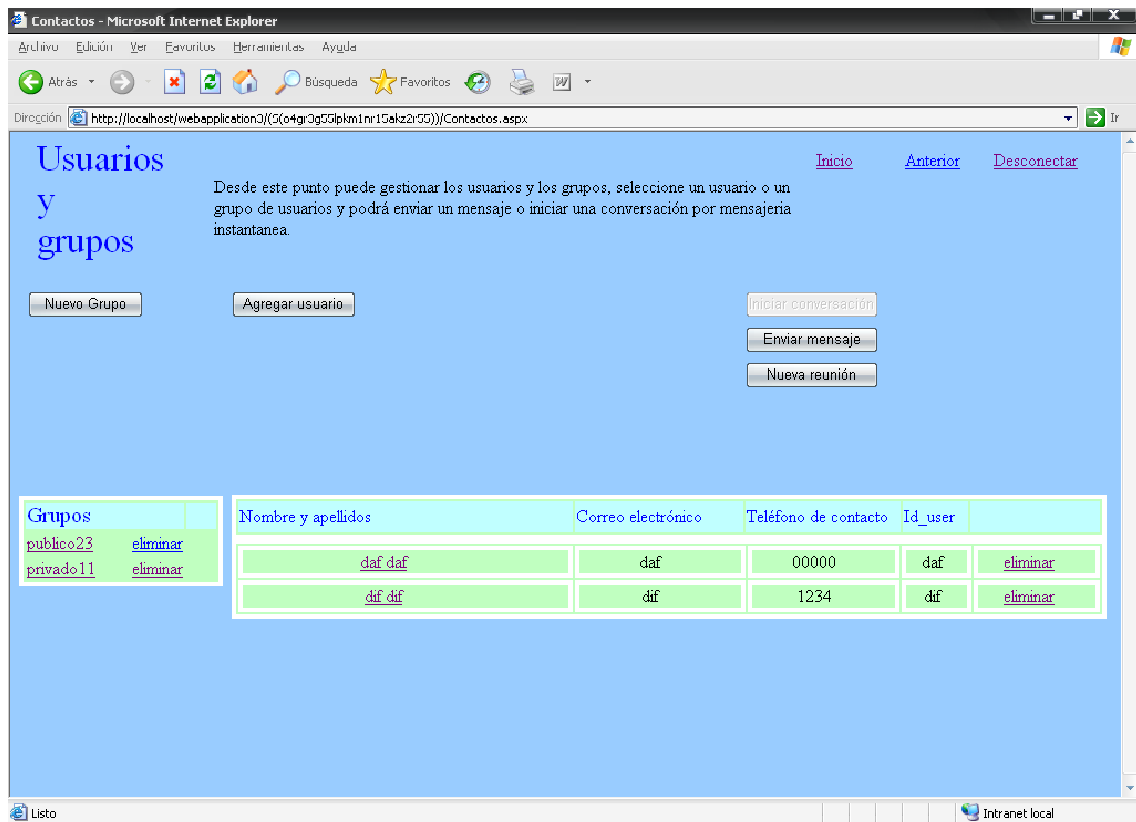
En la segunda configuración, el usuario convocado, podrá consultar los detalles de la reunión, pero no podrá modificar nada, salvo sus preferencias en el caso de que la reunión no haya sido confirmadas. A su vez, en caso de que la reunión no haya sido confirmada, el usuarios podrá confirmar su asistencia, siendo esto notificado al creador de la reunión.

3.10 Usuarios y grupos

Esta es la página a la que se accede desde la página principal pulsando sobre usuarios y grupos en el menú lateral:



Esta página se encarga de administrar los usuarios sobre los cuales se podrán enviar mensajes y hacer reuniones de una manera más sencilla, para evitar tener que recordar identificadores de usuario exactos. Este es el aspecto que tiene la página:



Se distinguen dos zonas, una central y derecha esta dedicada para la gestión de usuarios individuales; y una izquierda para la gestión de los grupos.

Gestión de usuarios:

La zona superior-derecha permite acceder a las páginas "Enviar Mensaje" y "nueva reunión".

Más centrado se encuentra el botón "Agregar usuario" que es el que permite agregar nuevos usuarios a la tabla de usuarios. Una vez se pulsa este botón se muestra una lista desplegable con los usuarios registrados en el sistema, para añadirlo a la lista se selecciona y se pulsa nuevamente al botón agregar usuario, mostrando un mensaje si el resultado ha sido el adecuado:

1

2

Usuario añadido correctamente.

La tabla inferior informa de los usuarios agregados para el cliente actual, mostrando su nombre, apellidos, e-mail, teléfono e identificador de usuario, así como también un enlace para eliminar este contacto de la lista.

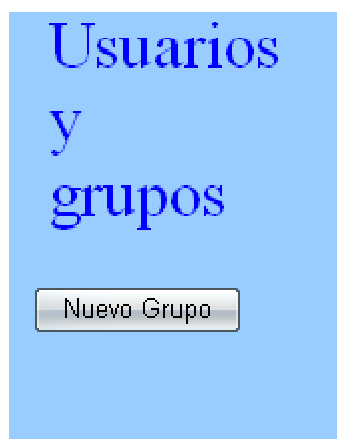
Nombre y apellidos	Correo electrónico	Teléfono de contacto	Id_user	
daf daf	daf	00000	daf	eliminar
dif dif	dif	1234	dif	eliminar

Gestión de grupos:

Al lado de la tabla de usuarios se encuentra la tabla de grupos, que muestra los grupos a los que pertenece el cliente, el cliente puede actuar sobre esta tabla en el caso de que fuera un grupo privado o un grupo público y el cliente fuera su creadores estos casos las acciones que se permiten son eliminar (pulsando sobre eliminar) o modificar (pulsando sobre el nombre).

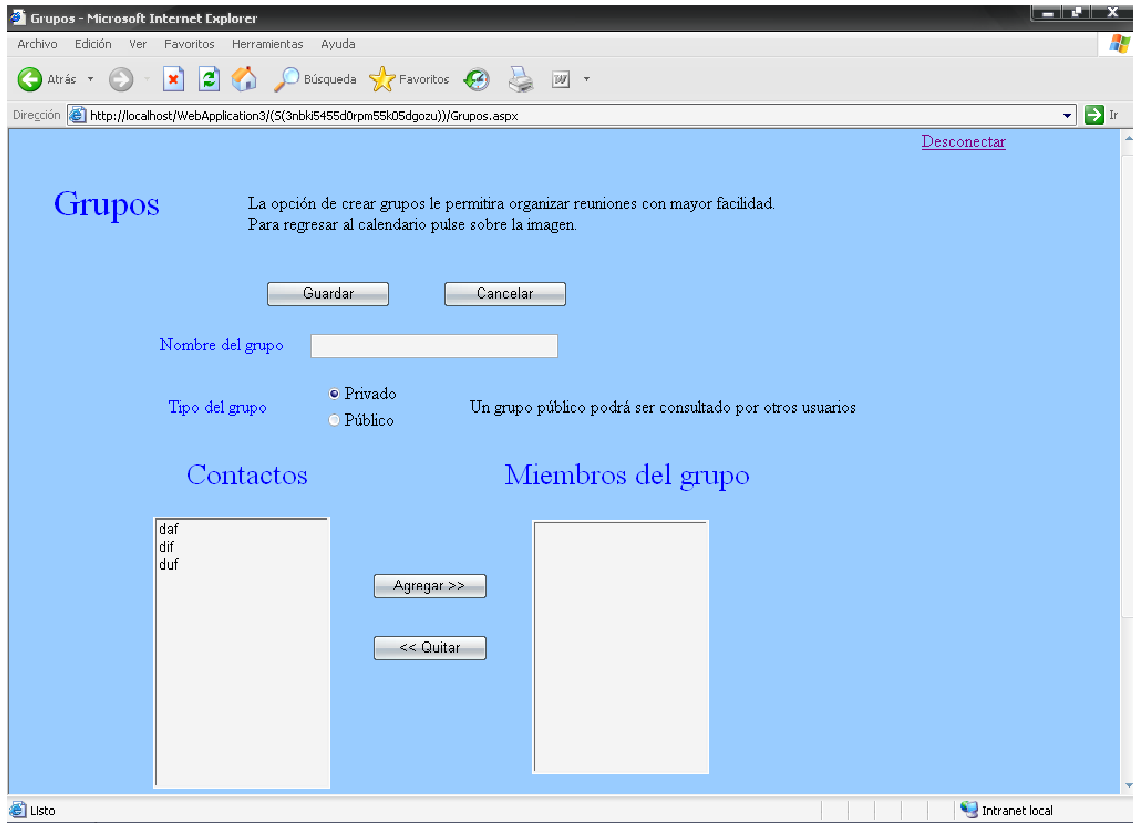
Grupos	
publico23	eliminar
privado11	eliminar

Encima de esta tabla se encuentra el botón nuevo grupo que enlaza con una página para crear grupos:



3.11 Nuevo grupo

Se llega a esta página tras pulsar nuevo grupo en la página usuarios y grupos, la página esta diseñada para crear nuevos grupos y su aspecto es el siguiente:



Para crear el nuevo grupo se deben seguir los siguientes pasos:

Nombre del grupo:

Se debe crear un nombre para este grupo, dicho nombre no puede contener ningún carácter que sea la barra espaciadora. Si se trata de un grupo público además no se puede usar ningún nombre de grupo público que ya exista en el sistema.

Tipo de grupo:

En este apartado se especifica que tipo de grupo queremos crear, y es obligatorio escoger entre uno de los dos tipos.

Un grupo privado será visto de forma exclusiva por el creador y ninguno de los usuarios que pertenezcan a él obtendrán ninguna información sobre este grupo.

Mientras que al crear un grupo público todos los integrantes del grupo tendrán conocimiento de ello, y podrán obtener la información del grupo.

Tipo del grupo	<input checked="" type="radio"/> Privado	Un grupo público podrá ser consultado por otros usuarios
	<input type="radio"/> Público	

Miembros del grupo:

El siguiente paso consiste en escoger entre los usuarios que tenemos agregados los que se desean añadir al grupo, para ello se dispone de dos listas, a la izquierda están contenidos todos los usuarios de la lista de usuarios, y a la derecha los que van a formar parte del nuevo grupo, para añadirlos a la lista de la derecha se selecciona un contacto y se pulsa el botón "Agregar >>", este paso se puede realizar varias veces o de una sola vez si se mantiene pulsado el botón "control" mientras se selecciona con el ratón a los contactos. En caso de error se permite seleccionar un contacto (o varios) de la lista de la derecha, pulsando el botón "<< Quitar" se borrara de la lista.

<h3>Contactos</h3> <div><ul style="list-style-type: none">dafdifdaf</div>	<div><p>Agregar >></p><p><< Quitar</p></div>	<h3>Miembros del grupo</h3> <div><ul style="list-style-type: none">dafdaf</div>
---	--	--

Guardar:

Para finalizar si se desea guardar el grupo hay que pulsar el botón "Guardar" situado en la parte superior de la pantalla, en caso contrario pulsar el botón "Cancelar".

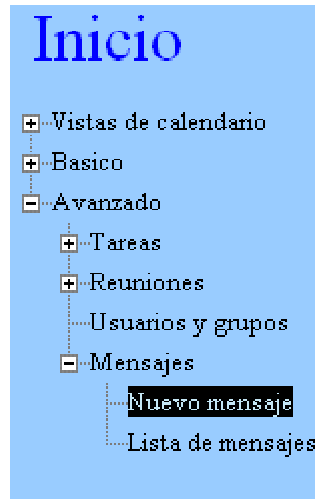
La opción de crear grupos le permitira organizar reuniones con mayor facilidad.
Para regresar al calendario pulse sobre la imagen.

Guardar

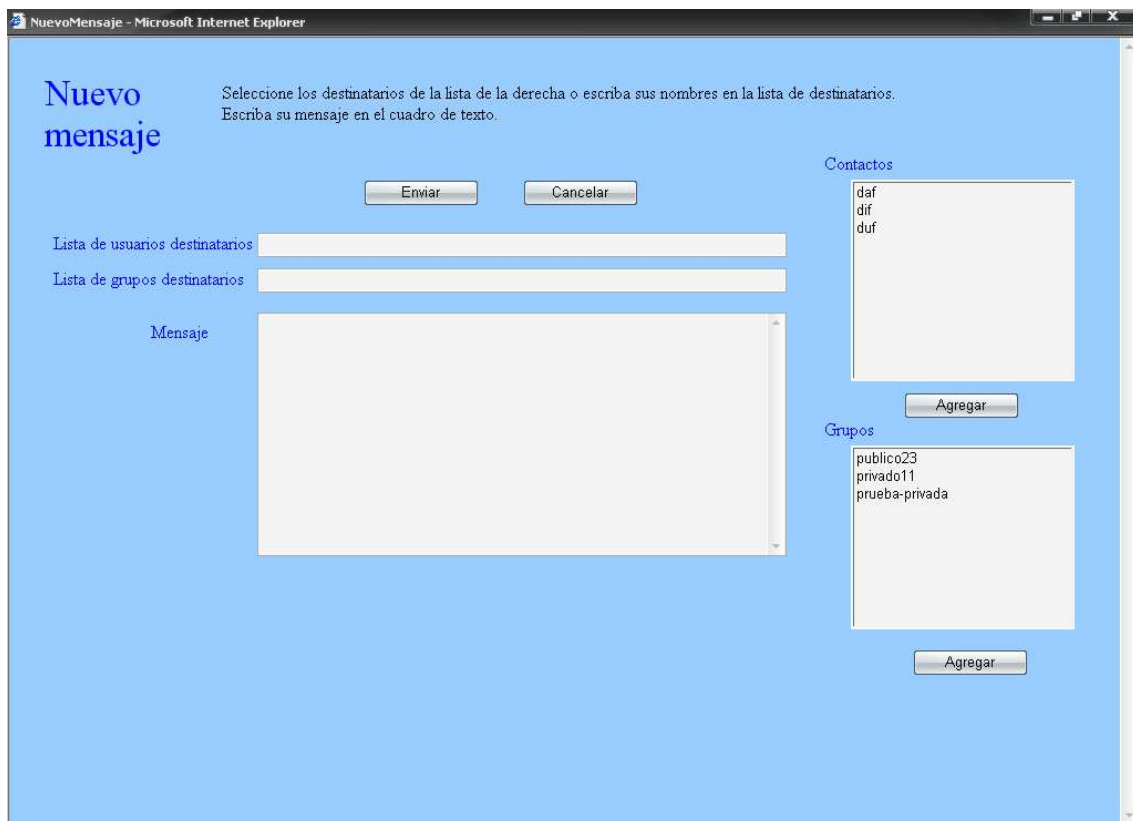
Cancelar

3.12 Nuevo mensaje

Se trata de una ventana emergente que se crea cuando se pulsa Nuevo mensaje en el menú lateral de la página principal:



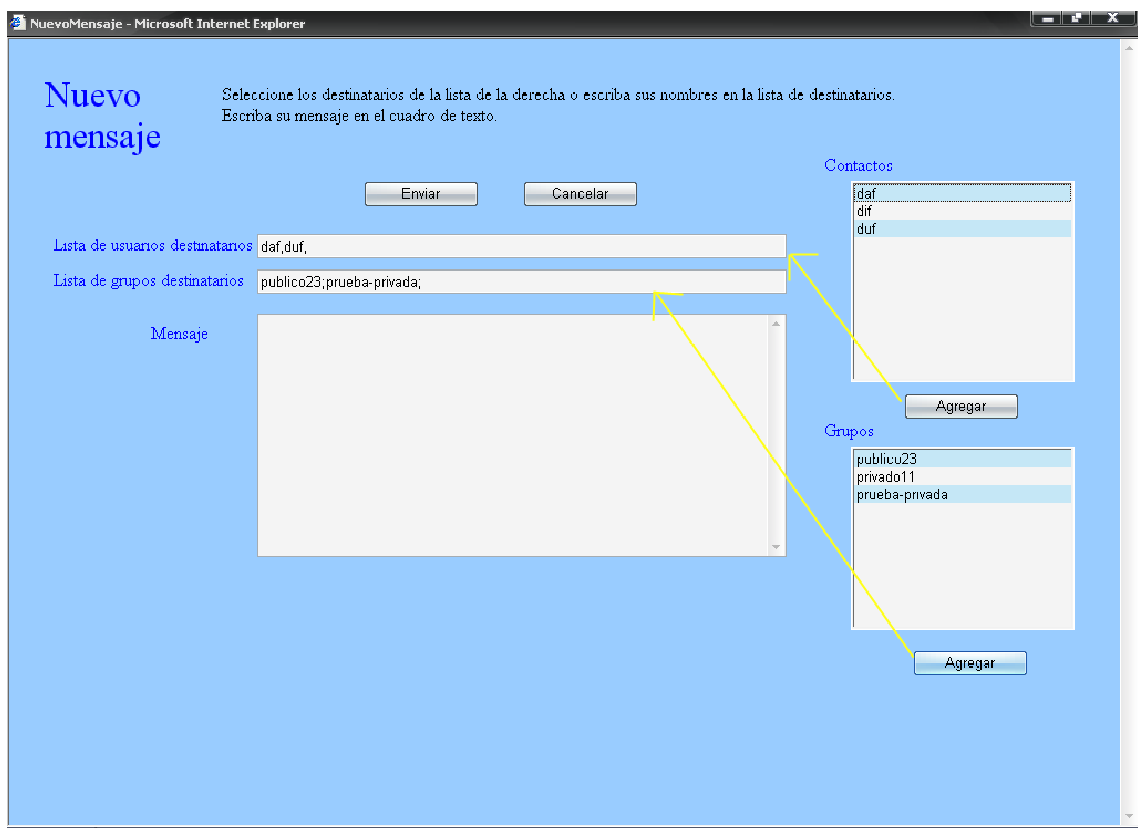
En esta página se aprecian dos campos para direcciones, un cuadro mas grande para escribir el mensaje y dos listas laterales, su aspecto es el siguiente:



Para crear un nuevo mensaje se deben seguir los siguientes pasos:

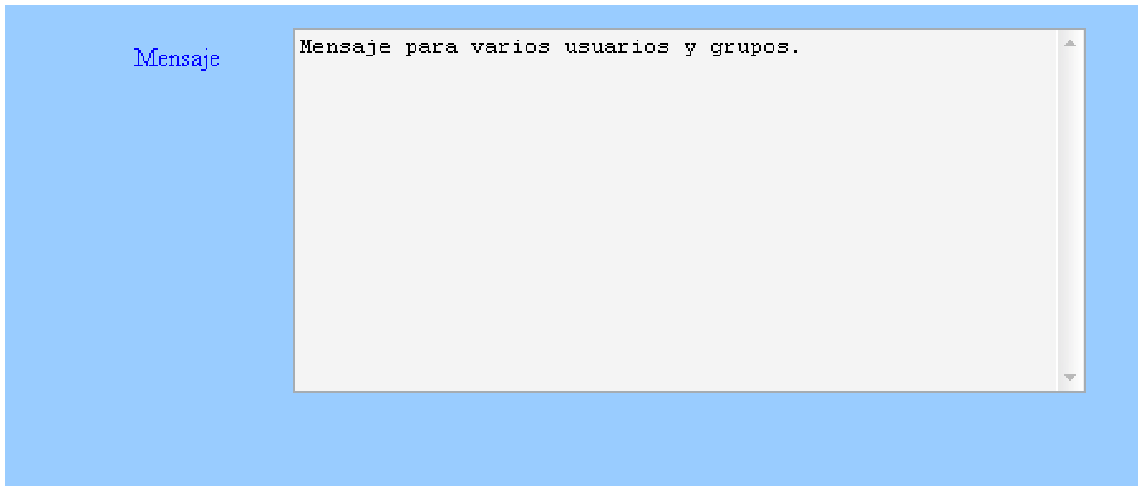
Seleccionar destinatario:

En las listas laterales se encuentran las listas de usuarios y grupos. Sobre ellas se pueden escoger los contactos a los que se pretende enviar el mensaje, una vez seleccionados se pueden añadir pulsando el botón agregar que se encuentra debajo de la lista. De igual modo se puede enviar el mensaje a todos los miembros de un grupo, seleccionando el grupo y pulsando el botón añadir que hay debajo de la lista. Al pulsar añadir, los contactos o grupos seleccionados se añadirán, separados por comas al contenido de los campos "Lista de usuarios destinatarios" o "Lista de grupos destinatarios" según proceda. En estos campos el usuario también puede escribir el identificador de usuario o grupo que desee sin necesidad de tenerlo en las listas de usuarios y grupos.



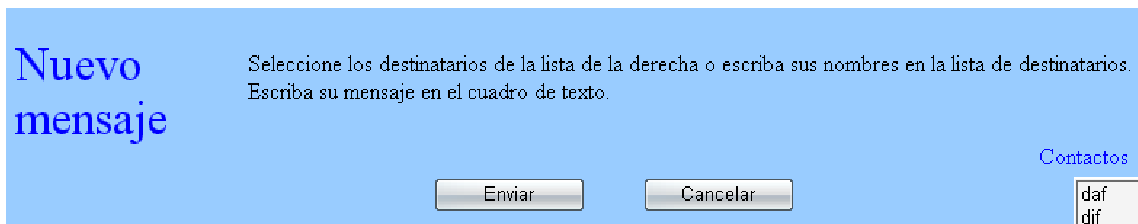
Redactar mensaje:

El siguiente paso es rellenar el cuadro de Mensaje con el contenido del mensaje que se desee enviar.



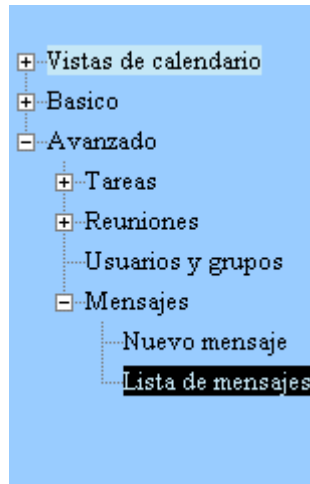
Confirmar envío:

Una vez realizados los anteriores pasos se enviará el mensaje cuando se pulse el botón enviar, mostrando un mensaje de envío satisfactorio, en caso de error se puede pulsar el botón cancelar y se cerrará la ventana emergente.

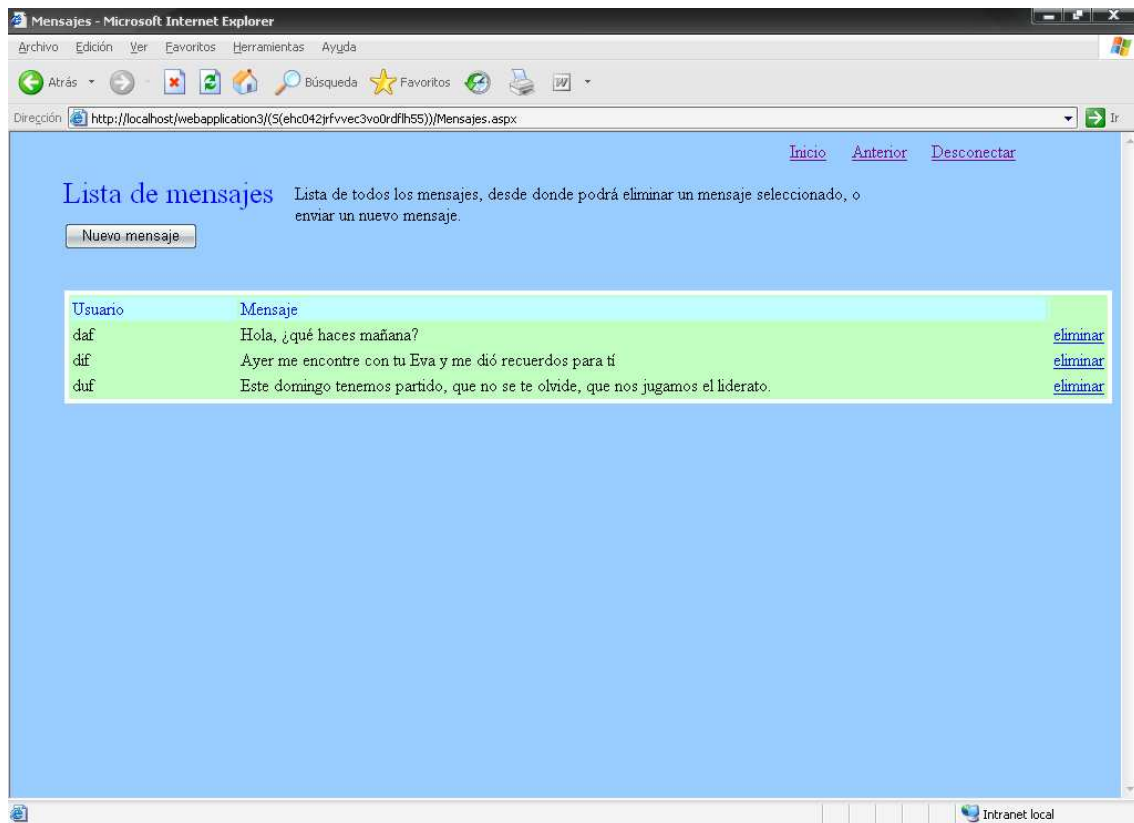


3.13 Lista de mensajes

A esta página se accede desde la principal, pulsando el enlace Lista de mensajes del menú izquierdo → avanzado:

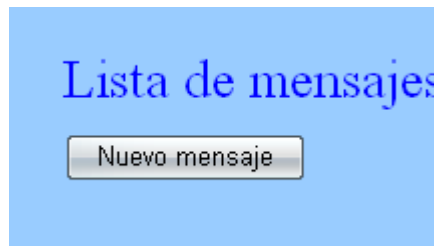


La página tiene el siguiente aspecto:



Los enlaces de navegación de arriba-derecha funcionan de manera similar al resto de páginas.

Esta página tiene un enlace a la página Nuevo mensaje:



Debajo de este enlace nos encontramos con la lista de mensajes recibidos, con el siguiente aspecto:

Usuario	Mensaje	
daf	Hola, ¿qué haces mañana?	eliminar
dif	Ayer me encontré con tu Eva y me dió recuerdos para tí	eliminar
duf	Este domingo tenemos partido, que no se te olvide, que nos jugamos el liderato.	eliminar

Aquí se observa en primer lugar el remitente y después el texto del mensaje, y a la derecha del todo hay un enlace con el texto "eliminar" cuya funcionalidad es eliminar el mensaje de esa línea:

Usuario	Mensaje	
daf	Hola, ¿qué haces mañana?	eliminar
dif	Ayer me encontré con tu Eva y me dió recuerdos para tí	eliminar
duf	Este domingo tenemos partido, que no se te olvide, que nos jugamos el liderato.	eliminar

Usuario	Mensaje	
daf	Hola, ¿qué haces mañana?	eliminar
duf	Este domingo tenemos partido, que no se te olvide, que nos jugamos el liderato.	eliminar

3.14 Periodicidad

Esta página permitirá introducir periodicidad tanto para las tareas como para las reuniones.

PeriodicidadTareas - Microsoft Internet Explorer

Periodicidad

Seleccione el tipo de repetición que desea para su tarea en la lista de la izquierda, y seleccione cuando desea que finalicen las repeticiones en la lista de la derecha

Aceptar Cancelar

Repetir: Fijar fecha de final

Nunca
Esta tarea no se repite

Diariamente
Repetir esta tarea cada día

Semanalmente
Repetir esta tarea cada semana(s) en

Lunes Martes Miércoles Jueves Viernes Sábado Domingo

Mensualmente
 En el de cada mes(es)

En la fecha de esta tarea

Anualmente
 En el de

Repetir cada año en la fecha de la tarea

Finalizar el:

agosto de 2006						
l	m	m	j	v	s	d
21	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

La página se compone de distintos apartados:

Título de la página: como en toda la aplicación se sitúa en la esquina superior izquierda, y en este caso indica al usuario que se encuentra en Periodicidad.

Breve descripción : Acompaña al título de la página y proporciona al usuario una breve descripción de la funcionalidad de la página.

Botón Aceptar: Almacena temporalmente la periodicidad que se ha introducido, con el fin de guardarla definitivamente asociada a una reunión o una tarea.

Boton Cancelar: Regresa a la página desde la que se ha accedido sin guardar los cambios que se han introducido.

Tipos de periodicidad: Existen 5 opciones entre las que se puede elegir, una opción excluye a las demás.

Nunca: Esta opción sirve para indicar que no se va a repetir, que no tiene periodicidad. Esta inicialmente marcada y va acompañada de una breve descripción.

Diariamente: Esta opción sirve para indicar una periodicidad diaria, a la misma hora que la fecha original. Va acompañada de una breve descripción.

Semanalmente: Esta opción es a su vez configurable con varios parámetros:

Número de semanas: Se debe seleccionar cada cuantas semanas se va a repetir. Tenemos un rango entre 1 y 4 para elegir.

Días de la semana: Se debe seleccionar que días de la semana se va a repetir. Si se tiene seleccionado este tipo de repetición, tiene que estar marcado al menos un día, en caso contrario se mostrará un mensaje indicándolo.

Mensualmente: Esta opción representa un tipo de periodicidad mensual y permite elegir entre dos opciones

Frecuencia no fija: Permite elegir un día de la semana, y si es el primero, segundo, tercero, cuarto o último del mes. También se puede seleccionar cada cuantos meses se va a repetir.

Frecuencia fija: Se repetirá en la fecha del original, todos los meses.

Anualmente: Esta opción representa un tipo de periodicidad anual y permite elegir entre dos opciones

Frecuencia no fija: Permite elegir un día de la semana, si es el primero, segundo, tercero, cuarto o último del mes, y el mes en el que se va a repetir todos los años.

Frecuencia fija: Se repetirá en la fecha original todos los años.

Fecha de final: Permite seleccionar hasta que fecha vamos a tener repeticiones.

Sin fecha de final: La repeticiones periódicas se van a suceder indefinidamente.

Con fecha de final: Se puede marcar la fecha a partir de la cual no se sucederán mas repeticiones periódicas. Si está marcada esta opción es necesario marcar alguna fecha, de lo contrario se avisará al usuario que la debe marcar.

3.15 Archivos adjuntos

Desde esta pagina se pueden gestionar los archivos adjuntos tanto de las tareas como de las reuniones.



Título de la página: como en toda la aplicación se sitúa en la esquina superior izquierda, y en este caso indica al usuario que se encuentra en la lista de archivos adjuntos.

Breve descripción : Acompaña al título de la página y proporciona al usuario una breve descripción de la funcionalidad de la página.

Botón aceptar: La acción de este botón es confirmar que deseamos adjuntar los archivos de la tabla a la reunión o la tarea.

Botón adjuntar: Se adjuntara a la tabla el archivo cuya ruta se encuentra en el cuadro de texto.

Botón examinar: Abre un explorador del disco mediante el cual podemos obtener la ruta del archivo deseado.

Tabla de archivos adjuntos: En esta tabla se muestran los archivos que ya han sido adjuntados.

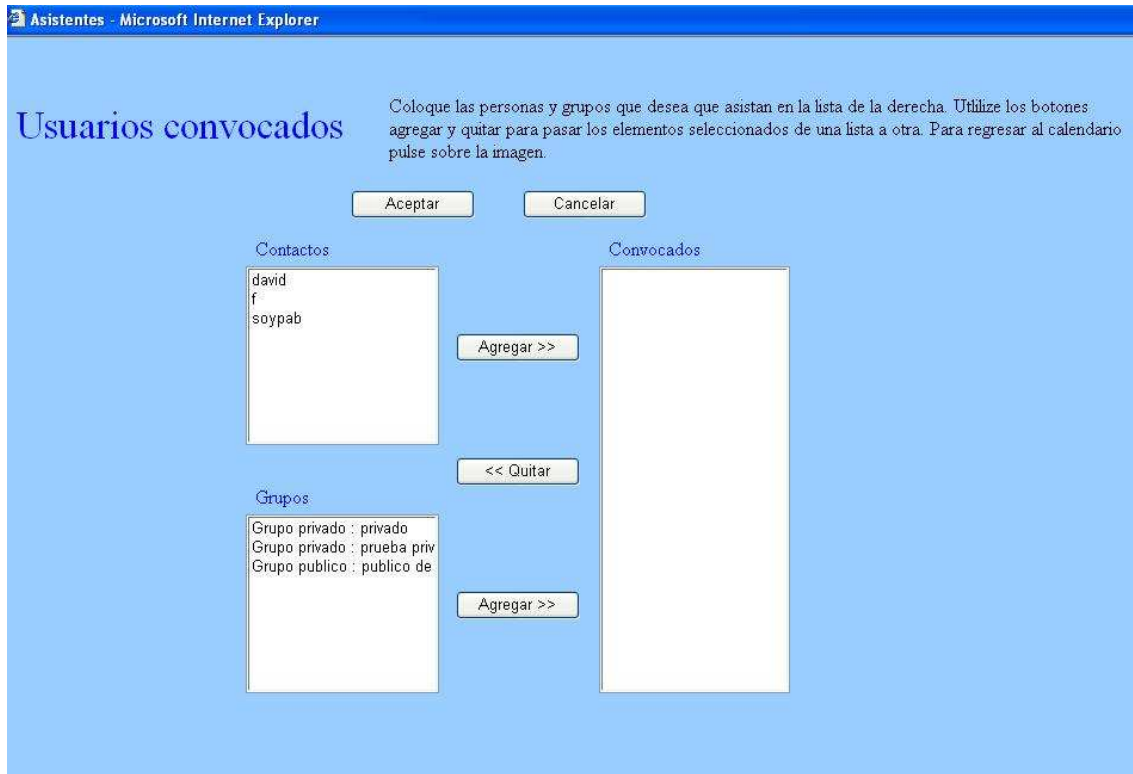
Datos adjuntos: En esta columna se sitúan los nombres de los archivos que se van a adjuntar.

Tamaño: En esta columna se sitúan los tamaños de los archivos en kilobytes.

Eliminar: En esta columna se tienen enlaces cuya función es eliminar el archivo de la tabla.

3.16 Usuarios convocados

Desde esta pagina podemos convocar a usuarios y a grupos a nuestras reuniones.



Título de la página: como en toda la aplicación se sitúa en la esquina superior izquierda, y en este caso indica al usuario que se encuentra en Usuarios convocados.

Breve descripción : Acompaña al título de la página y proporciona al usuario una breve descripción de la funcionalidad de la página.

Botón aceptar: La función de este botón es confirmar que la lista de convocados es correcta y regresar a la pagina anterior.

Botón cancelar: La función de este botón es regresar a la pagina anterior sin guardar los cambios realizados.

Lista de contactos: En esta lista se encuentran todos los usuarios que el usuario a admitido como contactos. Los usuarios seleccionados de esta lista serán agregados a la lista de convocados.

Lista de grupos: En esta lista se encuentran todos los grupos privados que el usuario a dado de alta, y todos los grupos públicos. Los grupos seleccionados de esta lista pasaran a la lista de convocados. Cada grupo esta precedido por un texto que indica si se trata de un grupo privado o un grupo público.

Lista de convocados: En esta lista se encuentran todos los usuarios y grupos que van a ser convocados.

Botones agregar: La función de estos botones es pasar los usuarios o los grupos seleccionados a la lista de convocados.

Botón quitar: La función de este botón es borrar de la lista de convocados los usuarios y grupos seleccionados.

3.17 Preferencias

Esta página sirve para que el usuario pueda consultar y modificar sus restricciones y sus preferencias para todas las reuniones. Existen 4 tipos de vista como en la página principal.



Título de la página: como en toda la aplicación se sitúa en la esquina superior izquierda, y en este caso indica al usuario que se encuentra en Preferencias.

Botón de navegación: Se encuentra en la esquina superior izquierda, con el texto anterior, y se utiliza para volver a la página de a que se venía.

Botón guardar: La función de este botón consiste en guardar los cambios que el usuario ha realizado sobre sus preferencias, quedando almacenadas para futuras reuniones.

Botón cancelar: La función de este botón consiste en regresar a la pagina de la que veníamos, sin guardar los cambios realizados.

Menú de vistas: En este menú desplegable que aparece en el lateral tenemos la opción de seleccionar las diferentes vistas del calendario. Las distintas configuraciones a las que podemos acceder son equivalentes a las indicadas en la explicación de la página principal.

Vista de mes: Esta es la vista seleccionada por defecto, podemos visualizar un calendario con todos los días del mes. Cada uno de los días se colorea de forma que si el día entero esta ocupado, preferible, o asumible, el dia se coloreara del correspondiente color, en otro caso, se mostrara como libre. Pulsando sobre cualquiera de los días, todas sus horas serán marcadas con la preferencia que tengamos seleccionada en

el menú de preferencias, pasando a la vista de día del correspondiente día pulsado.

Resto de vistas: La configuración del resto de vistas, también es igual a la descrita en la pagina principal. En este caso, cada una de las horas de los diferentes días estará coloreado según las preferencias introducidas por el usuario. Estarán en libre por defecto, y en alguna de las otras preferencias si ha sido modificada. Pulsando sobre cada una de las celdas con una preferencia seleccionada, la celda se coloreará del color correspondiente, indicando que ha sido marcada.

Menú de preferencias: Se sitúa inmediatamente debajo del menú de vistas, en el se puede seleccionar entre cuatro diferentes opciones. Cada una de estas opciones va acompañada del color en el que se marcara cuando seleccionemos una celda del calendario, a modo de leyenda, de forma que el usuario pueda asociar cada uno de los colores que observa en el calendario con cada una de las correspondientes preferencias.

Libre: Esta preferencia indica que ese día y esa hora se encuentra libre, para esta opción se ha utilizado el color verde oscuro.

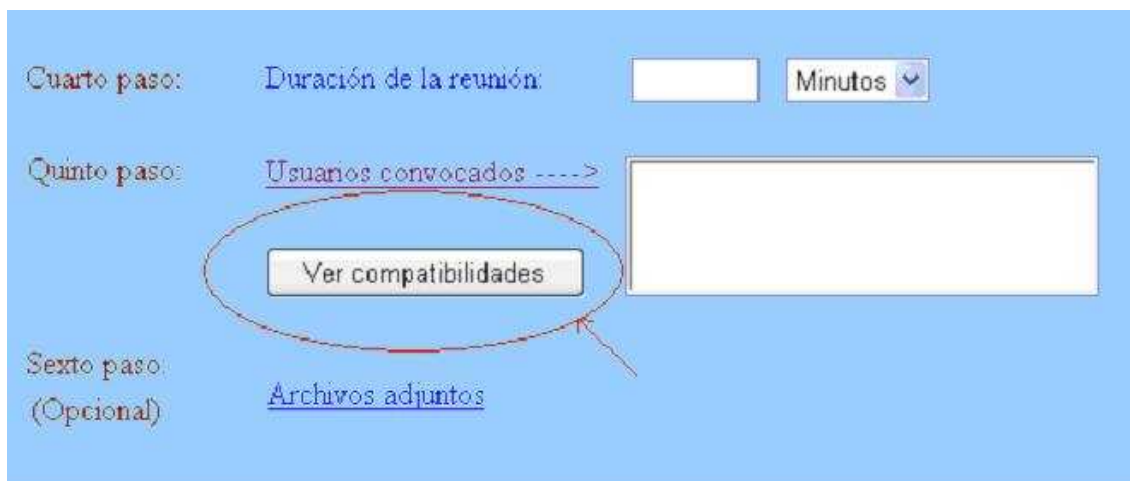
Preferible: Esta preferencia indica que las celdas marcadas con esta opción son un buen momento para realizar la reunión. Para esta opción se ha utilizado el color verde claro.

Asumible: Esta preferencia indica que en las celdas marcadas con esta opción es posible realizar la reunión aunque no es el mejor momento. Para esta opción se ha utilizado el color amarillo.

Ocupado: Esta preferencia indica que en las celdas marcadas con esta opción no es posible realizar la ya que. Para esta opción se ha utilizado el color rojo.

3.18 Probar compatibilidades

Esta es la página a la que se accede a la hora de probar las compatibilidades entre los distintos usuarios convocados a una reunión. Para acceder a la página en cuestión hay que pulsar sobre el botón "Ver compatibilidades" que se encuentra en la página de Nueva Reunión o en la de "Detalles de Reunión". Obsérvese la imagen para más detalle :



A continuación se detallan las partes de las que se compone la página :

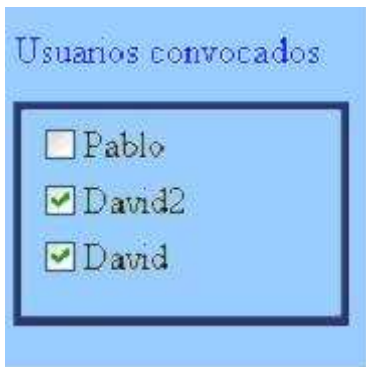
Título de la página: como en toda la aplicación se sitúa en la esquina superior izquierda, y en este caso indica al usuario que se encuentra en la página de probar compatibilidades.

Botones de navegación: Se sitúan en la esquina superior derecha y nos proporcionan enlaces con distintos puntos de la aplicación. En este caso solo se dispone del botón "Anterior" que nos redirige a la página de la cual venimos.

Menú lateral: El menú lateral se compone de un nodos principal.

Vistas del calendario: Al desplegar este menú se tiene acceso a diferentes configuraciones del calendario que se sitúa en la parte central. Se puede seleccionar la vista de día, semana, semana laboral o mes y en todas las vistas exceptuando la de mes además se puede seleccionar el grano de división en 30 minutos o en una hora.

Usuarios convocados: Aparecen justo debajo de las vistas del calendario. Se rellenará el cuadro correspondiente con tantos usuarios como haya convocados para la reunión, además del usuario creador. Cada uno de ellos aparecerá con un cuadradito seleccionable y, al hacerlo, se verán en las distintas vistas las restricciones de horarios de ese sujeto. Cuando haya varios usuarios seleccionados en la lista de convocados, el sistema mostrará en las vistas la fecha menos problemática para todos ellos. No obstante posteriormente se describe de forma más detallada con un ejemplo.



Leyenda: Aparecen los colores que usa la aplicación para mostrar la situación de cada fecha en el calendario que puede ser :

- Ocupada : Aparece la fecha y hora de color rojo.
- Asumible : Aparece la fecha y hora de color amarillo.
- Preferible : Aparece la fecha y hora de color verde lima.
- Libre : Aparece la fecha y hora de color verde oscuro.
- Recomendada por el sistema : Aparece la fecha y hora de color negro.

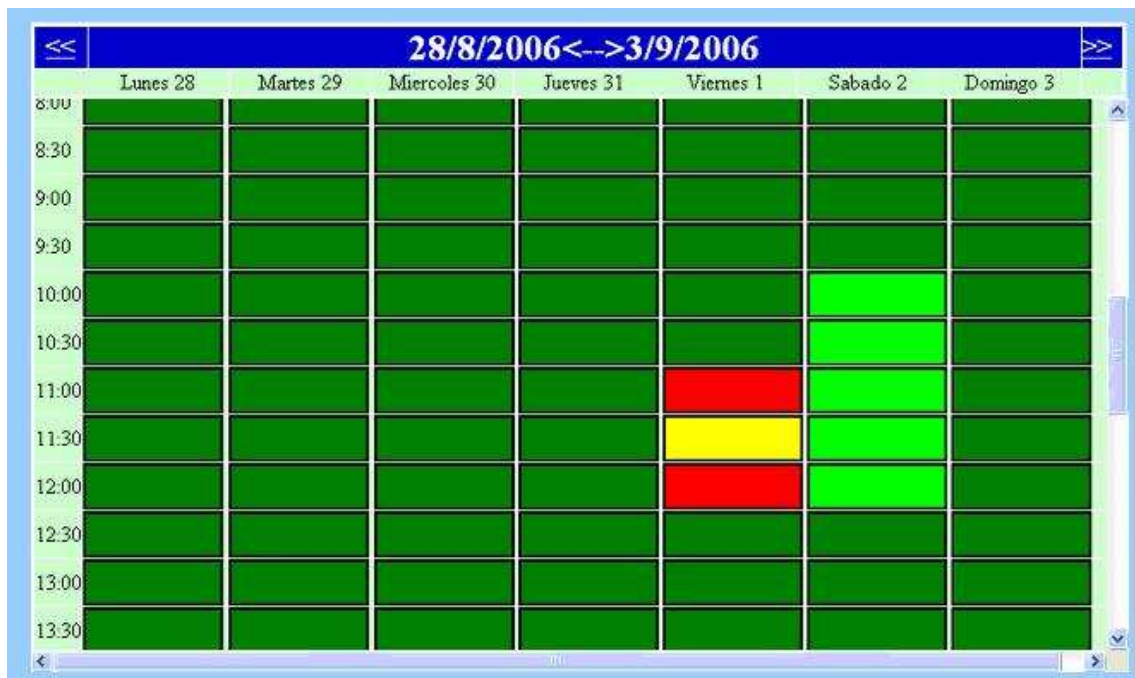


Zona central : Es la zona del calendario que cambia en función de la vista y grano que haya seleccionados.

Ejemplo de funcionamiento

A continuación se explica un ejemplo de funcionamiento de la página para apreciar su funcionalidad. El ejemplo consiste en lo siguiente :

Un usuario crea una reunión en el menú avanzado de la página principal. Tras rellenar los campos de asunto y descripción decide que quiere convocar una reunión para hablar con sus compañeros acerca de un proyecto de la asignatura de ingeniería superior informática "sistemas informáticos". Como a esta reunión deben asistir los 3 integrantes del grupo, (incluido el creador de la reunión) convoca a dos usuarios (los otros dos integrantes del grupo) y decide buscar una fecha para la reunión entre el 1 y el 2 de septiembre entre las 10 y las 12 horas. En este caso el creador de la reunión fija sus preferencias de horario de la siguiente forma :



El usuario convocado David2 fija sus preferencias de la siguiente forma :

28/8/2006<-->3/9/2006						
Lunes 28	Martes 29	Miercoles 30	Jueves 31	Viernes 1	Sabado 2	Domingo 3
8:30						
9:00						
9:30						
10:00				Yellow	Yellow	
10:30				Yellow	Yellow	
11:00				Yellow	Yellow	
11:30				Yellow	Yellow	
12:00				Yellow	Yellow	
12:30						
13:00						
13:30						
14:00						

El usuario convocado Pablo fija sus preferencias de la siguiente forma :

28/8/2006<-->3/9/2006						
Lunes 28	Martes 29	Miercoles 30	Jueves 31	Viernes 1	Sabado 2	Domingo 3
8:30						
9:00						
9:30						
10:00				Green		
10:30				Green		
11:00				Green		
11:30				Red	Red	
12:00				Red	Red	
12:30						
13:00						
13:30						
14:00						

Una vez fijada la fecha todos los usuarios, el creador de la reunión (David) accede a sus detalles de reunión y pulsa sobre el botón "Ver compatibilidades". Cuando no hay ningún usuario seleccionado le aparece lo siguiente.



Al no haber ningún usuario seleccionado en el cuadro de usuarios convocados, la aplicación informa que durante esas horas todos los usuarios seleccionados (en este caso ninguno) tienen la fecha y hora libres.

Ahora, si el usuario creador de la reunión selecciona a Pablo, le aparecerán las restricciones de Pablo. Además la aplicación buscará los huecos de horas que se encuentren libres para todos (en este caso sólo para Pablo), después los huecos que se encuentren preferibles y asumibles. Si hay algún hueco en el que todos los usuarios seleccionados tengan su restricción como ocupada, la aplicación no informará nada porque no habrá fecha disponible para esa fecha y hora, valga la redundancia.

Cuando haya conflictos de restricciones a una misma hora, la aplicación mostrará por pantalla la más restrictiva de todas ellas. Por ejemplo, si un usuario tiene las 12 horas del viernes 1 de septiembre como libre, y otro usuario tiene la misma hora como ocupada, en la tabla se mostrará como ocupada.



Finalmente el creador de la reunión selecciona a todos los usuarios y observa lo siguiente.



4. Requisitos

4.1 Introducción

Esta parte del documento es una especificación de requisitos software (ERS) para la aplicación Complu-Meeting siguiendo las directrices dadas por el estándar "IEEE *Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998.*"

4.1.1. Propósito

El objeto de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que se desea construir. El documento va dirigido tanto al equipo de desarrollo, como al director de proyecto. Este documento sirvió como medio de comunicación entre las partes implicadas.

4.1.2. Ámbito del sistema

Esta aplicación está pensada para toda clase de usuarios que deseen organizar su tiempo de una forma sencilla. Servirá como una herramienta de configuración del tiempo disponible en función de restricciones de carácter temporal o personal. Está pensada, sobre todo, para aquellos grupos de personas configurados de tal manera que necesiten reunirse periódicamente y, que debido a las diferentes ocupaciones de cada miembro integrante, les resulte muy difícil encontrar una fecha disponible para tal cometido. Sin embargo, como el fundamento de la aplicación será la sencillez de uso, ésta podrá ser usada por cualquier persona interesada en tener una agenda personal configurable, sencilla y fácil de usar.

4.1.3. Definiciones, acrónimos y abreviaturas

4.1.3.1. Definiciones

Usuario

Persona que utiliza la aplicación final sin ningún privilegio extra sobre otros usuarios que usen la misma aplicación, y sin ninguna capacidad para la modificación de la misma.

Contactos

Pueden ser usuarios o grupos. Son las personas que conoce el usuario de la aplicación y puede contactar con ellas.

Administrador

También puede ser denominado **súper usuario**. Es la persona que pudiendo usar la aplicación, además tiene los privilegios para dar de alta o de baja a nuevos usuarios, así como modificar la aplicación por cuestiones de mantenimiento o mejoras. Esta persona será única, es decir, solo habrá un usuario que sea administrador en la aplicación.

Gestor de reunión

Es cualquier usuario que organiza una reunión con otros usuarios y, debido a que es él el que la organiza, tiene capacidad para concretar la fecha final.

Grupos

Cada usuario podrá crear un grupo de usuarios siendo en ese caso líder del grupo, o pertenecer a uno de ellos por invitación. La utilidad del grupo se manifiesta a la hora de programar reuniones entre sus integrantes y a la hora de enviar mensajes. Se detallará en puntos posteriores.

Líder del grupo

Es el usuario que crea un grupo de usuarios y que tiene más privilegios que el resto de los integrantes del grupo con respecto al grupo en concreto.

Tarea

Cualquier hecho que el usuario pretende realizar en un determinado instante de tiempo y que coloca en el calendario de su aplicación, teniendo ésta la posibilidad de avisarle con antelación a la fecha .

Reunión

Es una tarea en la que se convocarán a varios usuarios en función de sus restricciones temporales. Se explicará de forma más detallada en puntos posteriores.

Mensaje

Como su propio nombre indica es un mensaje que se envía a uno o varios usuarios/grupos para informarles por medio de la aplicación en lugar de por otras vías como correo electrónico. Se explicará de forma más detallada en puntos posteriores.

Asistentes

Son los usuarios convocados a una reunión por el usuario que crea dicha reunión. El nombre asistente no implica que dicho usuario vaya a la reunión porque puede no tener compatibilidades horarias y, en ese caso, quedar fuera de la misma.

Periodicidad

Intervalo de tiempo en el que una determinada tarea o reunión se repetirá consecutivamente.

4.1.3.2. Acrónimos

ERS

Documento de Especificación de Requisitos Software.

4.1.3.3. Abreviaturas

ERS

Especificación de requisitos software.

J2ME

Java 2 Platform Micro Edition.

4.1.4. Referencias

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830, 1998.
- Documentación de requisitos del proyecto Vega Solaris.

4.1.5. Visión general del documento

Este documento consta de tres secciones. Esta sección es la Introducción y proporciona una visión general de la ERS. En la Sección 2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles. En la sección 3 se definen detalladamente los requisitos que debe satisfacer el sistema.

4.2. Descripción general

En esta sección se presenta una descripción de alto nivel del sistema. Se presentarán las principales áreas a las cuales el sistema debe dar soporte, las funciones que el sistema debe realizar, la información utilizada, las restricciones y cualquier otro tipo de factores que afecten al desarrollo del mismo.

4.2.1. Perspectiva del producto

El sistema estará alojado en un servidor remoto que aún se desconoce e interactuará con una base de datos. El usuario accederá a la aplicación a través de un PC conectándose a Internet al servidor anteriormente descrito.

4.2.2. Funciones del sistema

Primero hay que decir que como ya se tiene en marcha un prototipo de la aplicación funcionando, confeccionado mediante revisiones periódicas entre los integrantes del grupo de desarrollo y el jefe de proyecto, se irá explicando la funcionalidad de la aplicación basándose en dicho prototipo.

4.2.2.1. Alta en el sistema

Solo se podrá utilizar la aplicación si se registra uno en ella. Para ello hay que darse de alta en el sistema mediante un formulario que se explicará más detalladamente en el punto 3.

4.2.2.2. Baja en el sistema

Inicialmente en el prototipo este punto aparece opaco al usuario ya que será solo el administrador el que tendrá privilegios para dar de baja en el sistema a un determinado usuario.

4.2.2.3. Vistas del calendario

Cada usuario en la aplicación dispondrá de un calendario donde se verán reflejadas sus tareas y reuniones. Para ello dispone de 3 vistas: →Vista de mes.

→Vista de semana.

→Vista de día.

4.2.2.3.1 Vista de mes

Se corresponde con la vista típica de un calendario donde se muestra un mes entero en columnas separadas por días de la semana.

Haciendo clic con el ratón en el día correspondiente se podrá acceder a la vista del día.

4.2.2.3.2 Vista de semana

Es la vista típica de un horario o de una agenda personal. Cada columna se corresponde con un día de la semana y en cada fila hay unas determinadas horas que podrán mostrarse con mayor o menor grano; es decir, cada hora o cada media hora. Haciendo clic con el ratón en el día correspondiente se podrá acceder a la vista del día.

4.2.2.3.1 Vista del día

Es una tabla en la que hay una serie de filas que se corresponden con las horas para ese día. Es en esta vista donde se detallará en que hora o rango de ellas está programada la tarea o reunión correspondiente. Como en la vista de semana se pueden ver las filas cada hora o cada media hora.

4.2.2.4. Tareas

Aquí es donde cada usuario programa sus tareas. Cada tarea tiene un asunto, un lugar donde se especifica la tarea en si de una forma más detallada y unos archivos adjuntos que pueden aparecer o no. Puede tener una fecha de vencimiento o no. Si aparece fecha de vencimiento entonces se le ofrece al usuario la posibilidad de que la aplicación le avise con una antelación que se puede fijar, teniendo la posibilidad de que sea también a través del correo electrónico del usuario (además de en la propia aplicación).

4.2.2.5. Lista de tareas

Es una vista de las tareas que el usuario tiene pendiente de realizar. Aparecen en una tabla ordenada por filas donde las columnas corresponden al nombre de la tarea, el asunto, la fecha de vencimiento (en el caso de que la haya), la alarma (si la tiene) y la fecha de la alarma (si la tiene).

En esta vista se pueden modificar las tareas en espera de ser realizadas así como eliminarlas o acceder directamente a la zona de crear una nueva tarea.

4.2.2.6. Reunión

Es la pantalla donde se puede crear una reunión. El usuario que crea una reunión se convierte en el gestor de reunión. Para crear una reunión hay que seguir una serie de pasos que se detallarán en el punto 3.

4.2.2.7. Lista de reuniones

Similar a la lista de tareas solo que aparecen 2 tablas. En una de ellas están las reuniones confirmadas y en la otra las reuniones en fase de preparación, es decir, las que aún no han sido confirmadas por el gestor de reunión.

4.2.2.8. Usuarios y grupos

Es la pantalla donde aparecen la lista de contactos del usuario. Aparecen separados los usuarios individuales de los grupos. Es en esta pantalla donde un usuario crea un grupo, convirtiéndose en ese momento en el líder de dicho grupo. También se ofrece funcionalidad para la gestión de esos contactos: modificación, eliminación, etc. Se detallará en el punto 3.

4.2.3. Características del usuario

La aplicación está pensada para toda persona capacitada para usar un ordenador, con unos conocimientos básicos de informática de usuario. No es necesario ser un experto en el dominio para utilizar la aplicación ya que ésta se caracteriza por su sencillez en los controles y por un diseño muy intuitivo, siempre pensando en el usuario.

4.2.4. Restricciones

Sobre todo hay que tener en cuenta las restricciones que se les presentan a los desarrolladores de la aplicación.

El entorno de programación para el desarrollo de la aplicación será Microsoft Visual Studio 2003 que integra la tecnología. NET. La naturaleza de esta plataforma de desarrollo impone ciertas restricciones sobre todo a la hora de usar componentes en los formularios Web. Esto hace que algunos controles se implementen de una forma más complicada y ofrezcan menos funcionalidad de la esperada.

Otras posibles restricciones surgen por la base de datos que vayamos a utilizar. De momento está en fase de investigación pero estimamos que no dará problemas ni en cuanto a la carga de datos, ni en cuanto a la comunicación con la Web alojada en el servidor.

4.2.5. Suposiciones y dependencias

4.2.5.1. Suposiciones

Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado por el equipo de diseño atendiendo sugerencias de la comunidad. Cualquier petición de cambios en la especificación debe ser aprobada por el equipo de diseño y gestionada por el mismo.

4.2.5.2. Dependencias

No hemos encontrado dependencias de la aplicación con otros motivos, aparte de las obvias: el servidor donde se alojará, etc.

4.3.Requisitos específicos

4.3.1. Interfaces externas

4.3.1.1.Interfaces de usuario

La interfaz con el usuario es a través de la página Web y va a ser muy sencilla e intuitiva. En cada una de las pantallas o páginas Web por las que vaya navegando el usuario, se mostrará ayuda en forma de "tool tips". Estos consejos se mostrarán por pantalla cuando el usuario coloque el ratón encima de determinados controles. No obstante la funcionalidad está tan clara que no será necesario acceder a ellos para saber utilizar la aplicación.

4.3.1.2.Interfaces hardware

No han sido definidas interfaces hardware.

4.3.1.3. Interfaces software

No han sido definidas interfaces software.

4.3.1.4. Interfaces de comunicación

No han sido definidas interfaces software.

4.3.2. Funciones

4.3.2.1. Alta en el sistema

Descripción de la función: Permite que un usuario se dé de alta en el sistema rellenando un pequeño cuestionario con una serie de campos obligatorios y opcionales. Los campos obligatorios son necesarios para que dicho usuario se registre en la aplicación, mientras que los campos opcionales sirven para que el usuario de a conocer más características suyas al resto de usuarios, y no son necesarios para efectuar el registro.

Descripción de entrada: Ninguna.

Origen de la entrada: Se accede a la pantalla de alta en el sistema pulsando el botón "darse de alta" desde la pantalla de inicio de la aplicación.

Descripción de la salida: Si el proceso de darse de alta en el sistema se realiza correctamente, se accede a la página principal de la aplicación.

Destino de la salida: Página de inicio de la aplicación.

Requisito: Ninguno.

Precondiciones: El usuario no debe estar dado de alta en el sistema.

Poscondiciones: El usuario está dado de alta en el sistema.

4.3.2.2. Baja en el sistema

Descripción de la función: Un usuario se da de baja en el sistema y ya no puede acceder a la aplicación.

Descripción de entrada: La lleva a cabo el administrador.

Origen de la entrada: El administrador accediendo a su aplicación en la parte de contactos puede eliminar (dar de baja) a un usuario de la aplicación.

Descripción de la salida: El usuario que ha sido dado de baja no puede acceder a la aplicación con su anterior nombre de usuario y contraseña.

Destino de la salida: Terminal.

Requisito: Es responsabilidad del administrador dar de baja en la aplicación a un usuario si considera que el uso que está haciendo éste en la misma no es el correcto.

Precondiciones: El usuario debe estar dado de alta en el sistema.

Poscondiciones: El usuario está dado de baja en el sistema.

4.3.2.3. Vistas del calendario

4.3.2.3.1 Vista de mes

Descripción de la función: Se accede a la vista del calendario mostrando un mes completo. Es la vista por defecto de la aplicación. Desde aquí se puede acceder a toda la funcionalidad de la aplicación navegando por los botones desplegados.

Descripción de entrada: Ninguna.

Origen de la entrada: Se accede a esta pantalla al iniciar sesión en la aplicación, ya que es la pantalla inicial.

Descripción de la salida: Ninguna.

Destino de la salida: Página vista del mes de la aplicación.

Requisito: Ninguno.

Precondiciones: El usuario debe estar dado de alta en el sistema y ha iniciado sesión en la aplicación.

Poscondiciones: El usuario accede a la vista del mes de la aplicación.

4.3.2.3.2 Vista de semana

Descripción de la función: Se accede a la vista del calendario mostrando una semana completa, con la posibilidad de ocultar el sábado y el domingo, para no sobrecargar la pantalla con datos que no se desean. Desde aquí se puede acceder a toda la funcionalidad de la aplicación navegando por los botones desplegados.

Descripción de entrada: Ninguna.

Origen de la entrada: Se accede a esta pantalla al pulsar el botón vista de semana.

Descripción de la salida: Ninguna.

Destino de la salida: Página vista del mes de la aplicación.

Requisito: Ninguno.

Precondiciones: El usuario debe estar dado de alta en el sistema y ha iniciado sesión en la aplicación.

Poscondiciones: El usuario accede a la vista de la semana de la aplicación.

4.3.2.3 Vistas del día

Descripción de la función: Se accede a la vista del calendario mostrando un día completo. Es la vista más detallada de la aplicación y es aquí donde se encuentra la información más detallada sobre las tareas y reuniones programadas por el usuario. Desde aquí se puede acceder a toda la funcionalidad de la aplicación navegando por los botones desplegables.

Descripción de entrada: Ninguna.

Origen de la entrada: Se accede a esta pantalla al pulsar el botón vista de día.

Descripción de la salida: Ninguna.

Destino de la salida: Página vista del día de la aplicación.

Requisito: Ninguno.

Precondiciones: El usuario debe estar dado de alta en el sistema y ha iniciado sesión en la aplicación.

Poscondiciones: El usuario accede a la vista del día de la aplicación.

4.3.2.4. Tareas

Descripción de la función: Consiste en la creación de una tarea por parte del usuario. Cada tarea es un registro de información para el usuario que consta de una serie de campos:

-Asunto.

-Especificación de la tarea de una forma más detallada.

-Archivos adjuntos (opcionales)

-Fecha de vencimiento. Si aparece fecha de vencimiento entonces se le ofrece al usuario la posibilidad de que la aplicación le avise con una antelación que se puede fijar, teniendo la posibilidad de que sea también a través del correo electrónico del usuario (además de en la propia aplicación).

Cuando un usuario crea una tarea aparece en su calendario, que puede observar en las 3 vistas descritas anteriormente; de esta forma cuando haga clic con el ratón en el espacio destinado a la tarea concreta (en cualquiera de las 3 vistas), se le mostrará la información de los campos de la tarea de una forma más detallada.

Descripción de entrada: Ninguna.

Origen de la entrada: Se accede a esta pantalla al pulsar el botón Nueva tarea.

Descripción de la salida: Ninguna.

Destino de la salida: La nueva tarea planificada aparece en el calendario del usuario en el caso de que tenga fecha de vencimiento.

Requisito: Ninguno.

Precondiciones: El usuario debe estar dado de alta en el sistema y haber iniciado sesión en la aplicación.

Poscondiciones: El usuario ha creado una nueva tarea.

4.3.2.5. Lista de tareas

Descripción de la función: Es una vista de las tareas que el usuario tiene pendiente de realizar. Aparecen en una tabla ordenada por filas donde las columnas corresponden al nombre de la tarea, el asunto, la fecha de vencimiento (en el caso de que la haya), la alarma (si la tiene) y la fecha de la alarma (si la tiene). En esta vista se pueden modificar las tareas en espera de ser realizadas así como eliminarlas o acceder directamente a la zona de crear una nueva tarea.

Descripción de entrada: Ninguna.

Origen de la entrada: Se accede a esta pantalla al pulsar el botón Lista de tareas.

Descripción de la salida: Ninguna.

Destino de la salida: Ninguno, simplemente se accede a información.

Requisito: El usuario debe tener al menos una tarea creada para poder acceder a la lista de tareas.

Precondiciones: El usuario debe estar dado de alta en el sistema y haber iniciado sesión en la aplicación.

Poscondiciones: El usuario accede a su lista de tareas.

4.3.2.6. Reunión

Descripción de la función: Esta es la funcionalidad clave de la aplicación ya que consiste en la creación de una reunión entre el usuario que la crea y algunos de sus contactos/grupos.

Cuando un usuario crea una reunión se convierte en el líder de la reunión, es decir, en el integrante de la reunión que tendrá capacidad para fijar la fecha actual.

Se pueden crear dos clases de reuniones :

→Reuniones con fecha fija donde el creador de la reunión asigna una fecha y horas determinadas para dicha reunión.

→Reuniones donde se encuentra una fecha compatible con la mayoría de los asistentes, en un rango de fechas y horas determinado por el creador de la reunión.

Descripción de entrada: Ninguna.

Origen de la entrada: Se accede a esta pantalla al pulsar el botón Nueva reunión.

Descripción de la salida: Ninguna.

Destino de la salida: Ninguno, simplemente se accede a información.

Requisito: Ninguno.

Precondiciones: El usuario debe estar dado de alta en el sistema y haber iniciado sesión en la aplicación.

Poscondiciones: El usuario ha creado una nueva reunión.

4.3.2.7. Lista de reuniones

Descripción de la función: Se accede a la lista de reuniones del usuario.

Una reunión puede estar confirmada por todos los asistentes o en estado de preparación. Cuando una reunión está en estado de preparación puede ser, o bien porque hay algún/os asistente/s que aún no han indicado sus preferencias y restricciones en cuanto a fechas, o bien porque el líder de reunión aún no ha fijado la fecha definitiva para la reunión.

Cuando se hace clic sobre una reunión en preparación aparecen los detalles de la reunión.

Descripción de entrada: Ninguna.

Origen de la entrada: Se accede a esta pantalla al pulsar el botón Lista de reuniones.

Descripción de la salida: Ninguna.

Destino de la salida: Ninguno, simplemente se accede a información.

Requisito: Ninguno.

Precondiciones: El usuario debe estar dado de alta en el sistema y haber iniciado sesión en la aplicación.

Poscondiciones: El usuario accede a su lista de reuniones.

4.3.2.8. Usuarios y grupos

Descripción de la función: Se accede a los contactos del usuario. Desde esta pantalla se puede crear, modificar o eliminar un grupo y, en el caso de que sea el administrador el que acceda aquí, tendrá también la posibilidad de eliminar, modificar o crear nuevos usuarios. El usuario, desde aquí también podrá modificar sus propios datos.

Descripción de entrada: Ninguna.

Origen de la entrada: Se accede a esta pantalla al pulsar el botón Lista de usuarios.

Descripción de la salida: Ninguna.

Destino de la salida: Ninguno, simplemente se accede a información.

Requisito: Ninguno.

Precondiciones: El usuario debe estar dado de alta en el sistema y haber iniciado sesión en la aplicación.

Poscondiciones: El usuario accede a su lista de contactos.

5. Diseño y arquitectura de la aplicación

5.1. Introducción

La arquitectura y el diseño están necesariamente relacionados en el desarrollo de aplicaciones.

Complu-Meeting se pensó desde el principio como una aplicación centrada en el usuario. Los aspectos más a tener en cuenta en el diseño de la aplicación fueron, por lo tanto, todos aquellos relacionados con el usuario.

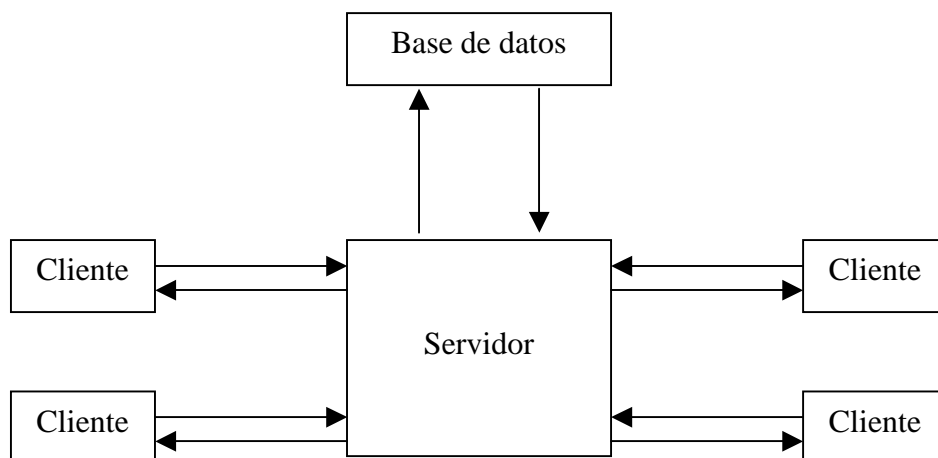
La mayoría de las aplicaciones comerciales de gestión de reuniones y tareas, contactos, citas... se centran sobre todo en el aspecto gráfico, en hacer algo visible y llamativo al usuario para que lo use, y dejan de lado o dan menos importancia a otros aspectos como la funcionalidad, la utilidad y la facilidad de uso.

Complu-Meeting sobre todo hace énfasis en esos aspectos descuidados de las aplicaciones e intenta relacionar al usuario con la aplicación a través de la facilidad de uso, la utilidad y la rapidez de gestión, descuidando los demás aspectos en los que otras aplicaciones más potentes y desarrolladas por grandes compañías la ganan naturalmente.

De esta forma, el diseño de Complu-Meeting comenzó con una serie de bocetos y pruebas y, en función de su utilidad se realizaron numerosos cambios teniendo en cuenta los aspectos que se describen anteriormente.

En el caso de esta aplicación, la arquitectura podría definirse utilizando el modelo cliente – servidor – base de datos, tal y como se ve en el dibujo.

5.2. Arquitectura



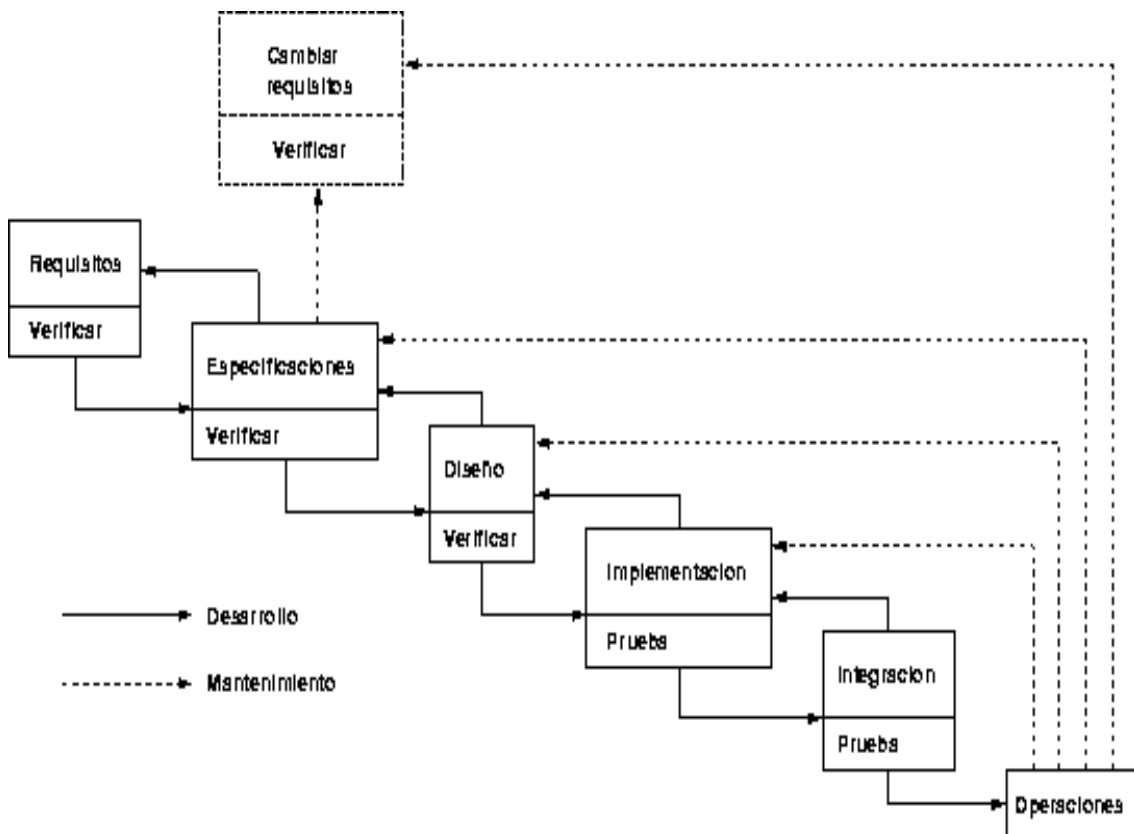
5.3. Diseño

Los aspectos fundamentales del diseño de la aplicación son los siguientes:

- El servidor va a ser el encargado de llevar a cabo todas las funciones de la aplicación relacionándose con la base de datos si es necesario.
- Los clientes van a acceder a la aplicación vía Internet conectándose al servidor a través de una dirección determinada, y no van a realizar ninguna gestión específica en relación con la aplicación, quedando ésta exclusivamente en manos del servidor.
- Cuando los clientes utilicen la aplicación guardarán cambios que quedarán reflejados en futuras visitas, tales como tareas o reuniones insertadas, cambio de horarios de reuniones creadas anteriormente, etc.

5.3.1 Modelo de desarrollo

El modelo de desarrollo de la aplicación se ha seguido teniendo en cuenta el modelo en cascada que se detalla en el siguiente cuadro:



En primer lugar se abordó la especificación de requisitos que fueron revisados por el director de proyecto. Como resultado de la revisión algunas ideas importantes fueron rechazadas por su inviabilidad o por su inadecuación en cuanto a su funcionalidad en el global de la aplicación.

De esta manera Complu-Meeting se convirtió, en su etapa de especificación, en una aplicación Web exclusivamente, y se descartaron otros tipos de uso contemplados en la especificación inicial del proyecto tales como comunicación con herramientas de gestión de reuniones, contactos (Outlook)

Complu-Meeting paso a ser una herramienta autónoma e independiente que había que implementar de cero para llevar a cabo toda su funcionalidad.

A partir de este punto vino la etapa de diseño de la aplicación propiamente dicha donde, por medio de bocetos carentes de funcionalidad, se fueron marcando las pautas de la aplicación en cuanto a forma y futura funcionalidad.

Como consecuencia de esta etapa hubo que hacer cambios en la especificación final revisada, ya que a la hora de visualizar la aplicación, algunas partes que a priori parecían importantes quedaron en un papel secundario, y otras a las que se les había dado menos importancia inicialmente adquirieron un papel más importante.

Un ejemplo de ello es la forma de navegación de la aplicación. Inicialmente, se había especificado que Complu-Meeting estuviese formada por una serie de páginas conectadas entre sí, cada una de ellas capaces de realizar tareas por separado y prácticamente independientes. A partir de la etapa de diseño se pensó que la herramienta sería más fácilmente usable si disponía de una página principal en la que se pudiesen realizar todas las funciones más destacables de la aplicación, si bien para realizarlas en algunos casos se requeriría acceder a otras páginas.

La etapa más importante, y para la que se dispuso de menos tiempo, fue la parte de implementación. Una pequeña parte de implementación se realizó durante la etapa de diseño. Al acabar la etapa de diseño el aspecto visual de la aplicación ya estaba prácticamente finalizado y lo único que faltaba era programar la funcionalidad de los distintos menús, botones, calendarios, etc. No obstante, a la hora de implementar la funcionalidad, el equipo se vio obligado a cambiar el aspecto de algunas tablas y calendarios que resultaban inservibles tal y como estaban diseñados. Buena parte de la culpa de todos los cambios la tuvo el hecho de usar una herramienta desconocida como Microsoft Visual Studio 2005 y el manejo de controles Web de aspx cuya potencia es inferior en cuanto a flexibilidad y funcionalidad.

La parte de implementación se hizo a la par que la parte de pruebas, pero las pruebas más exhaustivas tuvieron lugar a la hora de integrar las

distintas partes del código que funcionaba bien por separado pero con problemas interconectado. Esta parte se podría haber resuelto de un modo más eficiente mediante el uso de herramientas de sincronización tales como CVS(Concurrent Versions System) que permite una integración de código simultáneo, pero era más difícil su utilización en asp que la mejora que ofrece en el caso de usarlo.

Para facilitar la parte de mantenimiento se ofrecerá una parte en la memoria de funcionamiento de la aplicación a nivel de código fuente, así como un manual de usuario.

5.3.2 Diseño de alto nivel

A alto nivel la aplicación se organiza en torno a una página principal, que ofrece toda la funcionalidad al usuario (cliente)

Sin embargo, cada usuario accederá a la aplicación a través de una página de inicio. En esta página es en la que se ofrecerá la posibilidad de darse de alta como usuario de la aplicación. Para darse de alta hay que rellenar un pequeño formulario con una serie de campos obligatorios y otros opcionales que se encuentra en la página de alta.

Cada usuario dado de alta en el sistema podrá utilizarlo conectándose desde la página de inicio. Para ello deberá introducir su identificador de usuario y su contraseña. En caso de no recordar la contraseña, la aplicación ofrece la posibilidad de recordarla al usuario. Si este es el caso, el usuario deberá introducir en el campo de identificador de usuario su dirección de correo electrónico (aquella con la que se dio de alta en el sistema) y el sistema le avisará con un mensaje de correo de su identificador de usuario y contraseña.

Por otra parte, la aplicación también ofrece la posibilidad de recordar al usuario su identificador de usuario y / o contraseña seleccionándolo adecuadamente en la página de inicio antes de conectarse con la página principal. Este recordatorio se realiza a través del uso de cookies por lo que el usuario deberá tenerlas activadas en su equipo para que el recordatorio pueda realizarse adecuadamente.

Los aspectos más importantes de estas páginas de inicio y alta (así como de las demás) quedarán reflejados en el siguiente punto relacionado con el diseño de la aplicación a nivel de código fuente.

La página principal, como ya se ha comentado anteriormente, es la que ofrece toda la funcionalidad al usuario registrado en la aplicación. En ella es fácil distinguir tres partes claramente diferenciadas:

Zona de navegación: se sitúa en la esquina superior derecha y permite al usuario desplazarse entre las distintas páginas visitadas. Básicamente podrá

acceder a la página anteriormente visitada, a la página de modificación de sus datos (de alta) y a la página de inicio (desconexión del sistema)

Zona de acciones: se sitúa en la esquina superior izquierda a través de un menú desplegable. Las tres acciones principales de este menú son las vistas del calendario, las acciones básicas y las acciones avanzadas.

Zona de vista: la forma el resto de la página y es donde se detallarán los cambios que tengan efecto en la aplicación a través de las acciones que realice el usuario. Básicamente se compone de la vista del calendario y las distintas tablas que haya seleccionadas. Esta zona se explica con más detalle en la sección del manual que trata sobre la página principal.

Las demás páginas, a pesar de no ser tan rígidas en cuanto a estructura como la página principal, también dispondrán de capacidad de navegación (todas las páginas para las que sea útil su uso).

5.3.3 Diseño a nivel de código fuente

En esta sección se explican los aspectos más importantes de desarrollo de la aplicación a nivel de código fuente.

En primer lugar hay que distinguir dos partes claramente diferenciadas en cuanto al código:

- Código html de las páginas aspx.
- Código c# de las páginas .cs que se corresponden a cada página .aspx.

El código html de las páginas aspx (a las que se accede en la aplicación) es generado automáticamente por la herramienta de diseño de "Microsoft Visual Studio 2005".

En cambio el código .cs correspondiente a cada página .aspx es generado exclusivamente por nuestro equipo de desarrollo. Es a él al que haremos referencia a la hora de explicar los aspectos más importantes del código.

Cabe señalar que se detallará todo el código de cada página sino aquel que se considere más importante por su contenido y / o utilidad.

La parte de explicación vendrá acotada entre los comentarios EXPLICACIÓN

FIN DE EXPLICACIÓN.

La explicación comienza con la página de inicio aludida en puntos anteriores que será de las pocas que se lleven a cabo de una forma minuciosa debido a su brevedad en cuanto a código y a su importancia en cuanto a funcionalidad..

Página de Inicio

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

using System.Data.Odbc;

namespace WebApplication3
{
    /// <summary>
    /// Summary description for WebForm2.
    /// </summary>
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.CheckBox Verificacion2;
        protected System.Web.UI.WebControls.CheckBox Verificacion1;
        private string email;

        protected void Page_Load(object sender, System.EventArgs e)
        {
            if (!this.IsPostBack)
            {
                //buscamos las cookies que están habilitadas y las
                cargamos
                HttpCookieCollection coleccion = Request.Cookies;
                int num_cookies = coleccion.Count;
                for (int i = 0; i < num_cookies; i++)
                {
                    HttpCookie actual = Request.Cookies.Get(i);
                    if (actual.Name.Equals("usuario"))
                    {
                        this.Usuariobox.Text = actual.Value;
                        this.CheckBox2.Checked = true;
                    }
                    if (actual.Name.Equals("contrasena"))
                    {
                        this.contraseñabox.TextMode =
                        TextBoxMode.SingleLine;
                        this.contraseñabox.Text = actual.Value;
                        this.CheckBox1.Checked = true;
                    }
                }
            }
            if (this.CheckBox1.Checked && this.CheckBox2.Checked)
```

```

        this.CheckBox2.Checked = false;
    }
    else
        this.contraseñabox.TextMode = TextBoxMode.Password;
}

```

EXPLICACIÓN

El método Page Load formará parte de todas las páginas .aspx. tiene dos partes :

La parte de código correspondiente a **if (!this.IsPostBack)** que se ejecuta cuando se carga la página por primera vez.

La otra parte que se ejecuta cuando la página se recarga como respuesta a acciones del servidor que la modifiquen. En este caso al cargarse la página por primera vez, se buscan las cookies adecuadas y se cargan en las cajas de texto con sus valores correspondientes. Si la página se recarga una segunda o sucesivas veces se pone el texto de contraseña en modo contraseña para que no se vea lo que se teclea.

FIN EXPLICACIÓN

```

#region Web Form Designer generated code
override protected void OnInit(EventArgs e)
{
    //
    // CODEGEN: This call is required by the ASP.NET Web Form
Designer.
    //
    InitializeComponent();
    base.OnInit(e);
}

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
}
}
#endregion

```

EXPLICACIÓN

Esta parte es generada automáticamente por la herramienta de diseño de "Microsoft Visual Studio 2005".

FIN EXPLICACIÓN

```
protected void BotonInicio_Click(object sender,
System.EventArgs e)
    OdbcConnection Conn = new OdbcConnection("Driver={MySQL
ODBC 3.51 Driver};Server=localhost;" +
    "Trusted_Connection=yes;Database=complumeeting");
    Conn.Open();
    string mysqlquery1 = "Select count(*) from Usuario where
(Id_usuario = '" + this.Usuarioibox.Text + "') AND (Contrasena = '" +
this.contraseñaibox.Text + "')";
    OdbcCommand myCommand = new OdbcCommand(mysqlquery1,
Conn);
    Int64 c1 = (Int64)myCommand.ExecuteScalar();
    if (c1 == 1)
    {
        this.Session.Contents.Remove("anterior");
        this.Session.Contents.Add("anterior",
"/WebApplication3/Principal.aspx");
        this.Session.Contents.Remove("usuario");
        this.Session.Contents.Add("usuario",
this.Usuarioibox.Text);
        this.cookies();
        Response.Redirect("/WebApplication3/Principal.aspx");
    }
    else
    {
        string mysqlquery3 = "SELECT
count(usuario.Id_usuario)FROM usuario,e_mail WHERE usuario.Id_usuario
= e_mail.id_usuario && e_mail.E_mail = '" + this.Usuarioibox.Text + "'
&& usuario.Contrasena = '" + this.contraseñaibox.Text + "'";
        OdbcCommand myCommand3 = new OdbcCommand(mysqlquery3,
Conn);
        c1 = (Int64)myCommand3.ExecuteScalar();
        mysqlquery3 = "SELECT usuario.Id_usuario FROM
usuario,e_mail WHERE usuario.Id_usuario = e_mail.id_usuario &&
e_mail.E_mail = '" + this.Usuarioibox.Text + "' && usuario.Contrasena =
'" + this.contraseñaibox.Text + "'";
        myCommand3 = new OdbcCommand(mysqlquery3, Conn);
        string user = (string)myCommand3.ExecuteScalar();
        if (c1 == 1)
        {
            this.Session.Contents.Remove("anterior");
            this.Session.Contents.Add("anterior",
"/WebApplication3/Principal.aspx");
            this.Session.Contents.Remove("usuario");
            this.Session.Contents.Add("usuario", user);
            this.cookies();
        }
        Response.Redirect("/WebApplication3/Principal.aspx");
    }
    else
```

```

        {
            this.alerta("Introduzca un nombre de usuario y
contraseña válidos.");
        }
    }
    Conn.Close();
}
}

```

EXPLICACIÓN

Este evento se desencadena cuando se pulsas el botón de inicio para iniciar sesión en el sistema. Funciona accediendo a la base de datos para comprobar que el usuario este dado de alta en el sistema con la contraseña tecleada. Si no es el caso se avisa con el mensaje adecuado en un cuadro de texto.

FIN EXPLICACIÓN

```

protected void cookies()
{
    DateTime actual = DateTime.Now;
    actual = actual.AddYears(+1);
    if (CheckBox2.Checked) //guardar nombre de usuario como
cookie(eliminar cookie de contraseña)
    {
        HttpCookie usuario = new HttpCookie("usuario");
        usuario.Value = this.Usuario.Text;
        usuario.Expires = actual;
        Response.Cookies.Add(usuario);
        if (Request.Cookies["contrasena"] != null)
        {
            Response.Cookies["contrasena"].Value =
this.Usuario.Text;
            Response.Cookies["contrasena"].Expires =
DateTime.Now.AddDays(-1);
        }
    }
    else if (CheckBox1.Checked)
    { //guardar nombre de usuario y contraseña como cookies
        HttpCookie usuario = new HttpCookie("usuario");
        usuario.Value = this.Usuario.Text;
        usuario.Expires = actual;
        Response.Cookies.Add(usuario);
        HttpCookie contrasena = new HttpCookie("contrasena");
        contrasena.Value = this.contrasenabox.Text;
        contrasena.Expires = actual;
        Response.Cookies.Add(contrasena);
    }
    else
    {
        //hay que desabilitar todas las cookies
        actual = DateTime.Now;
    }
}

```

```

        actual = actual.AddDays(-1);
        int num_cookies = Request.Cookies.Count - 1;
        for (int i = 0; i <= num_cookies; i++)
        {
            HttpCookie tmp = Request.Cookies.Get(i);
            tmp.Expires = actual;
            Response.Cookies.Add(tmp);
        }
    }
}

```

EXPLICACIÓN

Esta función realiza toda la gestión de las cookies. Es importante señalar que para eliminar correctamente una cookie del equipo lo que hay que hacer es añadirla al sistema con un tiempo de vida anterior al actual. En otro caso no se elimina el archivo en si mismo del modo adecuado y ocasiona problemas.

FIN EXPLICACIÓN

```

private void alerta(string texto)
{
    string popupScript = "<script language='JavaScript'>" +
        "alert('" + texto + "')" + "</script>";
    if (!this.IsStartupScriptRegistered("alerta"))
        this.RegisterStartupScript("alerta", popupScript);
}

```

EXPLICACIÓN

Esta función abre un mensaje de alerta con el texto pasado como parámetro.

FIN EXPLICACIÓN

```

private string pedir_datos(string texto)
{
    string popupScript = "<script language='JavaScript'>" +
        "email = prompt('Introduzca su e-mail', '" + texto + "');"
    + "</script>";
    if (!this.IsStartupScriptRegistered("pedir_datos"))
        this.RegisterStartupScript("pedir_datos",
popupScript);
    this.email = email;
    return email;
}

```

EXPLICACIÓN

Esta función abre un mensaje de petición de texto con el texto pasado como parámetro, y lo recoge en la variable email (es para recordar la contraseña).

FIN EXPLICACIÓN

```
protected void BotonAlta_Click(object sender, System.EventArgs e)
{
    this.Session.Contents.Remove("anterior");
    this.Session.Contents.Remove("modifica");
    this.Session.Contents.Add("anterior",
"/WebApplication3/Inicio.aspx");
    this.Session.Contents.Add("modifica", "no");
    Response.Redirect("/WebApplication3/Alta.aspx");
}
```

EXPLICACIÓN

Esta función es la correspondiente a la pulsación del botón de alta y redirige a esa página en cuestión.

FIN EXPLICACIÓN

```
protected void CheckBox1_CheckedChanged(object sender,
System.EventArgs e)
{
    //Guardar mi nombre de usuario y mi contraseña como cookie
    if (CheckBox1.Checked)
        CheckBox2.Checked = false;
}

protected void CheckBox2_CheckedChanged(object sender,
System.EventArgs e)
{
    //Guardar mi nombre de usuario como cookie
    if (CheckBox2.Checked)
        CheckBox1.Checked = false;
}
```

EXPLICACIÓN

Si se pulsa guardar nombre de usuario hay que deseleccionar guardar nombre de usuario y contraseña y viceversa.

FIN EXPLICACIÓN

```
private void abreVentana(string ventana)
{
    //funcion que permite abrir una ventana emergente mediante
    javascript,
    //el parametro de entrada indica la pagina que cargamos

    string popupScript = "<script language='JavaScript'" +
        "window.open(' + ventana + "', 'CustomPopUp', " +

        "'directories=no,location=no,menubar=no,scrollbars=no,statusbar=no,width=300, height=150,resizable=no')" +
        "</script>";
    if (!this.IsStartupScriptRegistered("WOpen"))
    {
        this.RegisterStartupScript("WOpen", popupScript);
    }
}
```

EXPLICACIÓN

Esta función abre una ventana en segundo plano en forma de "Pop up".

FIN EXPLICACIÓN

```
protected void LinkButton1_Click(object sender, EventArgs e)
{
    if (this.Usuariobox.Text == "")
        this.alerta("Introduca su dirección de correo
    electrónico en el campo Usuario.");
    else
    {
        OdbcConnection Conn = new
    OdbcConnection("Driver={MySQL ODBC 3.51 Driver};Server=localhost;" +
        "Trusted_Connection=yes;Database=complumeeting");
        Conn.Open();
        string mysqlquery = "SELECT count(e_mail.E_mail) FROM
    e_mail WHERE e_mail.E_mail = '" + this.Usuariobox.Text + "'";
        OdbcCommand myCommand = new OdbcCommand(mysqlquery,
    Conn);

        Int64 c1 = (Int64)myCommand.ExecuteScalar();
        if (c1 == 1)
        {
```


EXPLICACIÓN

Esta función es una de las más importantes de la página de inicio y es llamada cuando el usuario pulsa sobre el enlace "¿Olvidó la contraseña?".

Pide que el usuario introduzca su email en el campo de Usuario y posteriormente realiza la consulta con la base de datos para comprobar que dicho email se encuentre disponible en la misma. Si no se encuentra disponible el email avisa al usuario y si se encuentra disponible obtiene el identificador de usuario y contraseña de la base de datos para posteriormente mandarselo por email.

FIN EXPLICACIÓN

Fin página de Inicio-----

A continuación se comentará la página de alta, también de manera exhaustiva debido a su importancia en toda la aplicación.

Página de Alta-----

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

using System.Data.Odbc;

namespace WebApplication3
{
    public partial class Alta : System.Web.UI.Page
    {
        private string email;
        private string nombre;
        private string apellidos;
        private string contraseña;
        private string contraseña2;
        private string organizacion;
        private string direccion;
        private string telefono_personal;
        private string telefono_trabajo;
        private string telefono_movil;
        private string fax;

        protected void Page_Load(object sender, System.EventArgs e)
        {
            string modifica;

            modifica=(string)this.Application.Contents["modifica"];

            if (modifica=="si" )
            {
                string usuario=
                (string)this.Application.Contents["usuario"];
                this.id_usuario.Text = usuario;
                this.id_usuario.Enabled = false;
                string my_sql = "SELECT Nombre, Apellidos,
                Organizacion, Direccion_personal, Direccion_trabajo,
                Telefono_personal, Telefono_trabajo, Telefono_movil, Fax FROM usuario
                WHERE (Id_usuario = '" + usuario + "')";

                OdbcConnection Conn = new
                OdbcConnection("Driver={MySQL ODBC 3.51 Driver};Server=localhost;" +
                "Trusted_Connection=yes;Database=complumeeting");
            }
        }
    }
}
```

```

Conn.Open();
OdbcCommand myCommand = new OdbcCommand(my_sql, Conn);
OdbcDataReader dr = myCommand.ExecuteReader();
string my_sql2 = "SELECT E_mail FROM E_mail
WHERE(id_usuario = '" + usuario + "')";
OdbcCommand myCommand2 = new OdbcCommand(my_sql2,
Conn);

OdbcDataReader dr2 = myCommand2.ExecuteReader();
dr.Read();
dr2.Read();
this.Nombre.Text = dr.GetString(0);
this.Apellidos.Text = dr.GetString(1) ;
    this.Email.Text =dr2.GetString(0);
this.Organizacion.Text = dr.GetString(2);
    this.Lcontrase2.Visible=true;
    this.lcontrase.Visible=false;
    this.labcontrase.Visible = false;
    this.labcontrase2.Visible = true;
    this.labelcontra.Visible =false;
    this.TextBox3.Visible =true;
    this.labcontraseb.Visible=true;
    this.ButtonSubmit.Text = "Guardar Cambios";

dr.Close();
dr2.Close();
Conn.Close();
}
else
{
this.id_usuario.Enabled = true;
    this.labcontrase2.Visible = false;
    this.labcontrase.Visible = true;
    this.lcontrase.Visible=true;
    this.Lcontrase2.Visible=false;

    this.TextBox3.Visible =false;
    this.labcontraseb.Visible=false;
    this.ButtonSubmit.Text = "Enviar";
}

this.nombre = this.Nombre.Text;
this.apellidos = this.Apellidos.Text;
this.contraseña = this.Contraseña.Text;
this.contraseña2 = this.Contraseña2.Text;
this.email = this.Email.Text;
this.organizacion = this.Organizacion.Text;
this.telefono_personal = this.Tel_personal.Text;
this.telefono_movil = this.Tel_movil.Text;
this.telefono_trabajo = this.Tel_trabajo.Text;
this.fax = this.Fax.Text;
}

```

EXPLICACIÓN

Este es el evento que se ejecuta cuando se carga la página. Hay que distinguir en esta parte de código si se accede a esta página para modificar los datos de un usuario o para darse de alta un nuevo usuario. Ésto se hace gracias a la variable de sesión "modifica". La funcionalidad del método es rellenar las variables en función de lo que haya en las cajas de texto.

FIN EXPLICACIÓN

```

#region Web Form Designer generated code
override protected void OnInit(EventArgs e)
{
    //
    // CODEGEN: This call is required by the ASP.NET Web
Form Designer.
    //
    InitializeComponent();
    base.OnInit(e);
}

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
}
#endregion

```

EXPLICACIÓN

Esta parte es generada automáticamente por la herramienta de diseño de "Microsoft Visual Studio 2005".

FIN EXPLICACIÓN

```

protected void ButtonSubmit_Click(object sender,
System.EventArgs e)
{
    string modifica =
(string)this.Application.Contents["modifica"];

    if (modifica == "no")
    {
        if ((this.id_usuario.Text == "") || (this.Nombre.Text
== "") || (this.Apellidos.Text == "") ||
            (this.Contraseña.Text == "") ||
(this.Contraseña2.Text == "") ||
            (this.Email.Text == ""))
            this.alerta("Faltan campos obligatorios por
rellenar");
        else
        {
            if (this.contraseña != this.contraseña2)
                this.alerta("Las contraseñas introducidas no
coinciden.");
            else
            {
                OdbcConnection Conn = new
OdbcConnection("Driver={MySQL ODBC 3.51 Driver};Server=localhost;" +

```

```

"Trusted_Connection=yes;Database=complumeeting");
    Conn.Open();
    string mysqlquery1 = "Select count(id_usuario)
from Usuario where id_usuario = '" + this.id_usuario.Text + "'";
    OdbcCommand myCommand = new
OdbcCommand(mysqlquery1, Conn);
    Int64 c1 = (Int64)myCommand.ExecuteScalar();
    string mysqlquery2 = "Select count(E_mail)
from e_mail where E_mail = '" + this.Email.Text + "'";
    string mysqlquery3 = "Select count(E_mail)
from e_mail where E_mail = '" + this.email2.Text + "'";
    string mysqlquery4 = "Select count(E_mail)
from e_mail where E_mail = '" + this.email3.Text + "'";
    OdbcCommand myCommand2 = new
OdbcCommand(mysqlquery2, Conn);
    Int64 c2 = (Int64)myCommand2.ExecuteScalar();
    OdbcCommand myCommand3 = new
OdbcCommand(mysqlquery3, Conn);
    Int64 c3 = (Int64)myCommand3.ExecuteScalar();
    OdbcCommand myCommand4 = new
OdbcCommand(mysqlquery4, Conn);
    Int64 c4 = (Int64)myCommand4.ExecuteScalar();
    Conn.Close();
    if (c1 > 0 || c2 > 0 || c3 > 0 || c4 > 0)
    {
        if (c1 > 0)
            this.alerta("El identificador de
usuario ya esta siendo utilizado, introduzca otro");
        if (c2 > 0)
            this.alerta("Ya existe alguien dado de
alta con el E_maill1 utilizado");
        if (c3 > 0)
            this.alerta("Ya existe alguien dado de
alta con el E_mail2 utilizado");
        if (c4 > 0)
            this.alerta("Ya existe alguien dado de
alta con el E_mail3 utilizado");
    }
    else
    {
        string dir_personal =
this.Dir_personal.Text + " " + this.Cod_postal_personal.Text + " " +
this.Ciudad_personal.Text + " " + this.Provinncia_personal.Text + " "
+ this.Pais_personal.Text;
        string dir_trabajo = this.Dir_trabajo.Text
+ " " + this.Cod_postal_trabajo.Text + " " + this.Ciudad_trabajo.Text
+ " " + this.Provinca_trabajo.Text + " " + this.Pais_trabajo.Text;
        string mysqlquery = "Insert into
usuario(id_usuario, Nombre, Apellidos, Contrasena,
Organizacion,Direccion_personal, Direccion_trabajo, Telefono_personal,
Telefono_trabajo,Telefono_movil, Fax, Administrador) Values (' +
this.id_usuario.Text + "','" +
this.Nombre.Text + "','" + this.Apellidos.Text + "','" +
this.Contrasena.Text + "','" + this.Organizacion.Text + "','" +
dir_personal + "','" + dir_trabajo + "','" + this.Tel_personal.Text +
 "','" + this.Tel_trabajo.Text + "','" + this.Tel_movil.Text + "','" +
this.Fax.Text + "','0)";
        this.Inserta_base_Datos(mysqlquery);
        if (this.Email.Text != "")
        {

```

```

        mysqlquery = "Insert into
e_mail(E_mail,id_usuario) Values('" + this.Email.Text + "','" +
this.id_usuario.Text + "')";
        this.Inserta_base_Datos(mysqlquery);
    }
    if (this.email2.Text != "")
    {
        mysqlquery = "Insert into
e_mail(E_mail,id_usuario) Values('" + this.email2.Text + "','" +
this.id_usuario.Text + "')";
        this.Inserta_base_Datos(mysqlquery);
    }
    if (this.email3.Text != "")
    {
        mysqlquery = "Insert into
e_mail(E_mail,id_usuario) Values('" + this.email3.Text + "','" +
this.id_usuario.Text + "')";
        this.Inserta_base_Datos(mysqlquery);
    }
}

Response.Redirect("/WebApplication3/Inicio.aspx");
}
}
}
else {
    string usuario =
(string)this.Application.Contents["usuario"];
    string my_sql = "SELECT contrasena FROM usuario WHERE(
Id_usuario = '" + usuario + "')";
    OdbcConnection Conn = new
OdbcConnection("Driver={MySQL ODBC 3.51 Driver};Server=localhost;" +
"Trusted_Connection=yes;Database=complummeeting");
    Conn.Open();
    OdbcCommand myCommand = new OdbcCommand(my_sql, Conn);
    OdbcDataReader dr = myCommand.ExecuteReader();
    dr.Read();
    string contraseñav = dr.GetString(0);
    dr.Close();
    if (this.Contrasena.Text == contraseñav ){
        string dir_personal = this.Dir_personal.Text + "~"
+ this.Cod_postal_personal.Text + "~" + this.Ciudad_personal.Text +
"~" + this.Provincia_personal.Text + "~" + this.Pais_personal.Text;
        string dir_trabajo = this.Dir_trabajo.Text + "~" +
this.Cod_postal_trabajo.Text + "~" + this.Ciudad_trabajo.Text + "~" +
this.Provincia_trabajo.Text + "~" + this.Pais_trabajo.Text;
        this.nombre = this.Nombre.Text;
        string mysqlquery = "Update usuario set Nombre
='"+this.nombre+"', Apellidos='"+this.Apellidos.Text+"', Organizacion
='"+this.Organizacion.Text+"',Direccion_personal='"+dir_personal+"',
Direccion_trabajo='"+dir_trabajo+"', Telefono_personal
='"+this.Tel_personal.Text+"', Telefono_trabajo
='"+this.Tel_trabajo.Text+
"','Telefono_movil='"+this.Tel_movil.Text+"', Fax
='"+this.Fax.Text+"' WHERE id_usuario='"+usuario+"'";
        this.Inserta_base_Datos(mysqlquery);
    }

Response.Redirect("/WebApplication3/Inicio.aspx");
//rellenar para cambiar datos.

```

```

    }
}

```

EXPLICACIÓN

Ésta es la función más importante de la página de alta y es un evento desencadenado por la pulsación del botón darse de alta. De nuevo hay que distinguir si lo que se quiere es modificar datos o dar de alta un nuevo usuario.

Si lo que se realiza es dar de alta un usuario, primero hay que comprobar que los datos que son necesario rellenar no estén vacíos, después que no haya otro usuario con el mismo id de usuario o e-mail, etc. Todo esto se consigue realizando consultas a la base de datos.

Si lo que se realiza es la modificación de datos del usuario actual, entonces se obtiene la id de usuario a través de la variable de sesión "usuario" (que se rellena con el valor adecuado en la página de inicio explicada anteriormente), y posteriormente se actualizan los datos cambiados en la base de datos con una consulta "update".

FIN EXPLICACIÓN

```

        protected void Linkbutton1_Click(object sender,
System.EventArgs e)
        {
            this.Confirmacion();
        }
        protected void Nombre_TextChanged(object sender, EventArgs e)
        {
            this.nombre = this.Nombre.Text;
        }
        protected void Apellidos_TextChanged(object sender, EventArgs
e)
        {
            this.apellidos = this.Apellidos.Text;
        }
        protected void Contraseña_TextChanged(object sender, EventArgs
e)
        {
            this.contraseña = this.Contraseña.Text;
        }
        protected void Contraseña2_TextChanged(object sender,
EventArgs e)
        {
            this.contraseña2 = this.Contraseña2.Text;
        }
        protected void Email_TextChanged(object sender, EventArgs e)
        {
            this.email = this.Email.Text;
        }
    }
}

```

```

e)     protected void TextBox5_TextChanged(object sender, EventArgs
        {
            this.organizacion = this.Organizacion.Text;
        }
e)     protected void TextBox6_TextChanged(object sender, EventArgs
        {
            this.telefono_personal = this.Tel_personal.Text;
        }
e)     protected void TextBox7_TextChanged(object sender, EventArgs
        {
            this.telefono_movil = this.Tel_movil.Text;
        }
e)     protected void TextBox8_TextChanged(object sender, EventArgs
        {
            this.telefono_trabajo = this.Tel_trabajo.Text;
        }
e)     protected void TextBox9_TextChanged(object sender, EventArgs
        {
            this.fax = this.Fax.Text;
        }

```

EXPLICACIÓN

Todas estas funciones son similares. Cuando se cambia algún cuadro de texto se invoca al evento correspondiente que realiza la modificación de la variable a la que hace referencia.

FIN EXPLICACIÓN

```

private void Inserta_base_Datos(string Ins_SQL)
{
    OdbcConnection Conn = new OdbcConnection("Driver={MySQL
ODBC 3.51 Driver};Server=localhost;" +
        "Trusted_Connection=yes;Database=complumeeting");
    Conn.Open();
    OdbcCommand myCommand = new OdbcCommand(Ins_SQL, Conn);
    myCommand.ExecuteNonQuery();
    Conn.Close();
}

```

EXPLICACIÓN

Esta función realiza la consulta en la base de datos pasada como parámetro "Ins_SQL"

FIN EXPLICACIÓN

Fin página de Alta-----

A continuación se comenta el código de la página principal. Como es la más extensa, se detallarán sólo las funciones más destacables

Página Principal-----

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using Microsoft.Web.UI.WebControls;
using System.Data.Odbc;

namespace WebApplication3
{
    /// <summary>
    /// Summary description for WebForm1.
    /// </summary>
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, System.EventArgs e)
        {
            if (this.Session.Contents["estado_principal"] == null)
            {
                Estado_principal est = new
                Estado_principal(this.Calendar1.TodaysDate, "mes", "1hora", "laboral",
                "ninguno");
                this.guarda_variable_memoria("estado_principal", est);
            }
            this.carga_estado_pagina();

            if (this.TableMensajes.Visible)
            {
                this.inserta_tabla_mensajes(this.TableDatos);
            }
            if (this.TableReunion.Visible)
            {
                this.inserta_tabla_reuniones(this.TableDatos);
            }
            if (this.TableTareas.Visible)
            {
                this.inserta_tabla_tareas(this.TableDatos);
            }

            this.Calendar1.SelectedDate = new DateTime();
        }
    }
}
```

```

        this.borra_controles_tabla();
        this.inicializa_tabla_semana();
        DateTime d = this.inicializaTablaSem();
        if (this.PanelReunionR.Visible ||
this.PanelTareaR.Visible)
        {
            this.inserta_tareas_reuniones_botones(d);
        }
        else
        {
            this.borra_variables_tareaR();
            this.borra_variables_reunionR();
            this.borra_botones_tabla();
            this.inserta_tareas_celdas(d);
            this.carga_colores_iniciales_celdas();
            this.inserta_reuniones_celdas(d);
        }

        this.carga_colores_iniciales_botones();
        if (this.PanelReunionR.Visible)
        {
            this.rellena_lista_usuarios_convocados();
            this.elimina_repeticones_convocados();
            this.carga_visualizacion_reunion_R();
        }
        else
        {
            this.borra_variables_reunionR();
        }
    }
}

```

EXPLICACIÓN

Cuando se carga la página principal hay que actualizar adecuadamente las tablas y las vistas. Se hace del modo siguiente :

1º → Se carga el estado de la página o se crea un estado de la página donde los parámetros iniciales son : tipo de vista = mes, día actual seleccionado = día de hoy, tipo de horario laboral, tipo de separación entre horas = cada una hora (también puede ser cada media hora). Posteriormente se explicará como se realiza esta carga.

2º → Se insertan en las tablas correspondientes los mensajes, las reuniones y las tareas.

3º → Se inicializa la tabla de semana (la que contendrá todas las vistas excepto la de mes)

4º → Si los paneles de reuniones básicas o tareas básicas son visibles se actualizan los botones de las vistas.

5º → Si los paneles no son visibles se "resetea" la tabla de semana eliminando los botones anteriores, cargando los botones iniciales, y cargando las tareas y reuniones.

6º → Si el panel de reuniones básicas está visible, se rellena la lista de usuarios convocados (sin repeticiones), sino se borran todas las variables que hagan referencia a reuniones básicas.

FIN EXPLICACIÓN

```
#region Web Form Designer generated code
override protected void OnInit(EventArgs e)
{
    //
    // CODEGEN: This call is required by the ASP.NET Web Form
Designer.
    //

    InitializeComponent();
    base.OnInit(e);
}

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.TreeViewMenu.SelectedIndexChange += new
Microsoft.Web.UI.WebControls.SelectEventHandler(this.TreeViewMenu_Sele
ctedIndexChange);
    this.Calendar1.DayRender += new
DayRenderEventHandler(this.Calendar1_DayRender);
}
#endregion
```

EXPLICACIÓN

Esta parte es generada automáticamente por la herramienta de diseño de "Microsoft Visual Studio 2005".

FIN EXPLICACIÓN

```

        private void Calendar1_DayRender(object sender,
System.Web.UI.WebControls.DayRenderEventArgs e)
        {
            this.inserta_tareas_vista_mes(e);
            this.inserta_reuniones_vista_mes(e);
        }

```

EXPLICACIÓN

Cuando se pinta un día en el calendario se cargan las reuniones y tareas en el caso de que haya para ese día determinado.

FIN EXPLICACIÓN

```

        protected void LinkButton5_Click(object sender,
System.EventArgs e)
        {
            this.Session.Contents.Remove("estado_principal");
            this.Session.Contents.Remove("anterior");
            this.Session.Contents.Add("anterior",
"/WebApplication3/(S(" + this.Session.SessionID +
"))/Principal.aspx");
            Response.Redirect("/WebApplication3/(S(" +
this.Session.SessionID + ")/inicio.aspx");
        }

```

EXPLICACIÓN

Cuando se pulsa sobre el botón de navegación anterior, se guarda como página anterior la actual y se accede a la página anterior.

FIN EXPLICACIÓN

```

        protected void SemanaPosterior_Click(object sender,
System.EventArgs e)
        {
            Estado_principal est =
(Estado_principal)this.carga_variable_memoria("estado_principal");
            switch (est.vista)
            {
                case "dia":
                    est.dia = est.dia.AddDays(1);
                    break;
                case "semana":
                    est.dia = est.dia.AddDays(7);
                    break;
                case "semanalaboral":
                    est.dia = est.dia.AddDays(7);
                    break;
            }
        }

```

```

    }
    this.guarda_variable_memoria("estado_principal", est);
    this.Page_Load(null, null);
}

```

EXPLICACIÓN

Cuando se pulsa sobre el símbolo >> de la tabla de semana se accede a la semana, o día siguiente en función de la vista seleccionada actualmente. Para ello se guarda en la variable de sesión "estado_principal" y después se recarga la página.

FIN EXPLICACIÓN

```

    protected void SemanaAnterior_Click(object sender,
System.EventArgs e)
    {
        Estado_principal est =
(Estado_principal)this.carga_variable_memoria("estado_principal");
        switch (est.vista)
        {
            case "dia":
                est.dia = est.dia.AddDays(-1);
                break;
            case "semana":
                est.dia = est.dia.AddDays(-7);
                break;
            case "semanalaboral":
                est.dia = est.dia.AddDays(-7);
                break;
        }
        this.guarda_variable_memoria("estado_principal", est);
        this.Page_Load(null, null);
    }
}

```

EXPLICACIÓN

Cuando se pulsa sobre el símbolo <<< de la tabla de semana se accede a la semana, o día anterior en función de la vista seleccionada actualmente. Para ello se guarda en la variable de sesión "estado_principal" y después se recarga la página.

FIN EXPLICACIÓN

```

        private void
inserta_tareas_vista_mes(System.Web.UI.WebControls.DayRenderEventArgs
e)
    {
        string usuario = (string)this.Session.Contents["usuario"];
        string dia = e.Day.Date.Year.ToString() + "-" +
e.Day.Date.Month.ToString() + "-" + e.Day.Date.Day.ToString();
        string my_sql = "SELECT      Asunto, Fecha, Hora FROM tarea
WHERE (Fecha = '" + dia + "') && (id_usuario_creador = '" + usuario +
"' ) ORDER BY Fecha, Hora";

        OdbcConnection Conn = new OdbcConnection("Driver={MySQL
ODBC 3.51 Driver};Server=localhost;" +
        "Trusted_Connection=yes;Database=complumeting");
        Conn.Open();
        OdbcCommand myCommand = new OdbcCommand(my_sql, Conn);
        OdbcDataReader dr = myCommand.ExecuteReader();
        string s = "";
        while (dr.Read())
        {
            s = "<br>" + dr.GetString(0);
            e.Cell.Controls.Add(new LiteralControl(s));
        }
        dr.Close();
        Conn.Close();
    }

```

EXPLICACIÓN

Inserta las tareas cuando la vista seleccionada es mes; para ello realiza consultas en la base de datos para ver si hay tareas en el día actual.

FIN EXPLICACIÓN

```

        private void
inserta_reuniones_vista_mes(System.Web.UI.WebControls.DayRenderEventAr
gs e)
    {
        string usuario = (string)this.Session.Contents["usuario"];
        string dia_ini = e.Day.Date.Year.ToString() + "-" +
e.Day.Date.Month.ToString() + "-" + e.Day.Date.Day.ToString() + "
00:00:00";
        string dia_fin = e.Day.Date.Year.ToString() + "-" +
e.Day.Date.Month.ToString() + "-" + e.Day.Date.Day.ToString() + "
23:59:59";

        string my_sql = "SELECT Asunto FROM reunion WHERE
((Fecha_ini >= '" + dia_ini + "') && " +
"(Fecha_ini <= '" + dia_fin + "')) && " +
"((Id_reunion IN (SELECT Id_reunion FROM convocado WHERE (id_usuario =
'" + usuario + "'))) || " +
"(id_usuario_creador = '" + usuario + "'))"; ;

```

```

        OdbcConnection Conn = new OdbcConnection("Driver={MySQL
ODBC 3.51 Driver};Server=localhost;" +
        "Trusted_Connection=yes;Database=complumeeting");
        Conn.Open();
        OdbcCommand myCommand = new OdbcCommand(my_sql, Conn);
        OdbcDataReader dr = myCommand.ExecuteReader();
        string s = "";
        while (dr.Read())
        {
            s = "<br>" + dr.GetString(0);
            e.Cell.Controls.Add(new LiteralControl(s));
        }
        dr.Close();
        Conn.Close();
    }
}

```

EXPLICACIÓN

Inserta las reuniones cuando la vista seleccionada es mes; para ello realiza consultas en la base de datos para ver si hay reuniones en el día actual.

FIN EXPLICACIÓN

```

        /*inicializa la tabla en la que se visualizan la vista de dia,
        * de semana y de semana laboral, con todos sus horarios,
        * devuelve el primer dia de la semana*/
        private DateTime inicializaTablaSem()
        {
            //inicializa la tabla donde se muestran las diferentes
vistas, mostrara la vista
            //seleccionada en "vista"
            DateTime d = this.escribe_titulo();
            return d;
        }

        /*inicializa la tabla de semana*/
        private void inicializa_tabla_semana()
        {
            int n = 0;
            int lb = 0;
            Estado_principal est =
(Estado_principal)this.carga_variable_memoria("estado_principal");
            for (int i = 0; i < 48; i++)
            {
                for (int j = 1; j < 8; j++)
                {
                    ((Button)this.TableSemana.Rows[i].Cells[j].Controls[0]).Text = "";
                    this.TableSemana.Rows[i].Cells[j].ID =
n.ToString();
                    this.TableSemana.Rows[i].Cells[j].Visible = true;
                }
            }
        }
    }
}

```

```

        n++;
        for (int k = 1; k <
this.TableSemana.Rows[i].Cells[j].Controls.Count; k++)
        {
this.TableSemana.Rows[i].Cells[j].Controls[k].ID = "Link-" +
lb.ToString();
            lb++;
        }
    }
    if (est.grano == "1hora")
    {
+ 2)        for (int i = 1; i < this.TableSemana.Rows.Count; i = i
            {
                this.TableSemana.Rows[i].Visible = false;
            }
    }
    else
    {
+ 2)        for (int i = 1; i < this.TableSemana.Rows.Count; i = i
            {
                this.TableSemana.Rows[i].Visible = true;
            }
    }
    switch (est.vista)
    {
        case "mes":
            break;
        case "semana":
            this.inicializa_tabla_semana_completa();
            break;
        case "semanalaboral":
            this.inicializa_tabla_semana_laboral();
            break;
        case "dia":
            this.inicializa_tabla_semana_dia();
            break;
    }
    this.carga_vista_horario();
}

```

EXPLICACIÓN

Inicializa la tabla de semana en función de la vista seleccionada y el grano seleccionado. Esos metodos harán visibles parte de la tabla o no en función de la vista pero no se detallarán.

FIN EXPLICACIÓN

```

        /*funcion generica que dependiendo de la vista seleccionada
pone
correctamente la cabecera de la tabla, devuelve el primer dia
de la semana*/
private DateTime escribe_titulo()
{
    Estado_principal est =
(Estado_principal)this.carga_variable_memoria("estado_principal");
    switch (est.vista)
    {
        case "mes":
            break;
        case "semana":
            return this.escribe_titulo_semana();
            break;
        case "semanalaboral":
            return this.escribe_titulo_semanalaboral();
            break;
        case "dia":
            return this.escribe_titulo_dia();
            break;
    }
    return est.dia;
}
}

```

EXPLICACIÓN

Función que escribe el título en la cabecera de la tabla de semana en función de la vista seleccionada.

FIN EXPLICACIÓN

```

private void inserta_tareas_reuniones_botones(DateTime
primer_dia)
{
    Estado_principal est =
(Estado_principal)this.carga_variable_memoria("estado_principal");
    switch (est.vista)
    {
        case "mes":
            break;
        case "semana":
            DateTime aux = primer_dia;
            for (int i = 1; i < 8; i++)
            {
                this.inserta_tareas_vista_botones(aux, i);
                this.inserta_reuniones_vista_botones(aux, i);
                aux = aux.AddDays(1);
            }
            break;
        case "semanalaboral":
            aux = primer_dia;
            for (int i = 1; i < 6; i++)

```

```

        {
            this.inserta_tareas_vista_botones(aux, i);
            this.inserta_reuniones_vista_botones(aux, i);
            aux = aux.AddDays(1);
        }
        break;
    case "dia":
        this.inserta_tareas_vista_botones(primer_dia, 1);
        this.inserta_reuniones_vista_botones(primer_dia,
1);
        break;
    }
}
}

```

EXPLICACIÓN

Función que inserta las tareas y las reuniones en los botones de la vista semana, semana laboral y día cuando se está anotando una tarea rápida o una reunión rápida. Para ello llama a las funciones "inserta_tareas_vista_botones" e "inserta_reuniones_vista_botones" de forma adecuada dependiendo de la vista actual.

FIN EXPLICACIÓN

```

        /*inserta tareas en los botones de las diferentes vistas*/
        private void inserta_tareas_vista_botones(DateTime actual, int
Columna)
        {
            Estado_principal est =
(Estado_principal)this.carga_variable_memoria("estado_principal");
            string usuario = (string)this.Session.Contents["usuario"];
            string dia = actual.Year.ToString() + "-" +
actual.Month.ToString() + "-" + actual.Day.ToString();
            string my_sql = "SELECT Asunto, Fecha, Hora FROM tarea
WHERE (Fecha = '" + dia + "') && (id_usuario_creador = '" + usuario +
"') ORDER BY Fecha, Hora";

            OdbcConnection Conn = new OdbcConnection("Driver={MySQL
ODBC 3.51 Driver};Server=localhost;" +
"Trusted_Connection=yes;Database=complumeeting");
            Conn.Open();
            OdbcCommand myCommand = new OdbcCommand(my_sql, Conn);
            OdbcDataReader dr = myCommand.ExecuteReader();

            while (dr.Read())
            {
                string hora = dr.GetString(2);
                int h = 0;
                int c = 0;
                string horal;
                string hora2;
                if (h >= 0 && h <= 9)
                {

```

```

        hora1 = ("0" + h.ToString() + ":00:00");
        hora2 = ("0" + h.ToString() + ":30:00");
    }
    else
    {
        hora1 = (h.ToString() + ":00:00");
        hora2 = (h.ToString() + ":30:00");
    }
    while ((hora != hora1) && (hora != hora2))
    {
        h++;
        c = c + 2;
        if (h >= 0 && h <= 9)
        {
            hora1 = ("0" + h.ToString() + ":00:00");
            hora2 = ("0" + h.ToString() + ":30:00");
        }
        else
        {
            hora1 = (h.ToString() + ":00:00");
            hora2 = (h.ToString() + ":30:00");
        }
    }
    if (hora != hora1)
    {
        c++;
    }
}
dr.Close();
Conn.Close();
}

```

```

((Button)this.TableSemana.Rows[c].Cells[Columna].Controls[0]).Text =
((Button)this.TableSemana.Rows[c].Cells[Columna].Controls[0]).Text + "
" + dr.GetString(0);

```

EXPLICACIÓN

Función que inserta tareas en los botones de las diferentes vistas. Lo más señalado de este método es la conversión de horas y minutos que hay que realizar para insertar la tarea en la celda correspondiente de la vista de semana. El otro método "inserta_reuniones_vista_botones" no se detallará porque es análogo.

FIN EXPLICACIÓN

```

        /*se pasa como parametro el identificador de la celda pulsada y
        y en que dia comienza la tabla,
        * y devuelve a que dia y hora corresponde la celda*/
        private DateTime identificaCelda(String pulsada, DateTime
DiaComienzo)
        { //dado un boton busca a que dia de la semana y a que hora
pertenece.
            DateTime d = DiaComienzo;
            Estado_principal est =
(Estado_principal) this.carga_variable_memoria("estado_principal");
            switch (est.vista)
            {
                case "semana":
                    for (int j = 1; j < 8; j++)
                    {
                        for (int i = 0; i < 48; i++)
                            if
(this.TableSemana.Rows[i].Cells[j].ID.Equals(pulsada))
                                {
                                    return d;
                                }
                            else
                                {
                                    d = d.AddMinutes(30);
                                }
                        }
                    break;
                case "semanalaboral":
                    for (int j = 1; j < 8; j++)
                    {
                        for (int i = 0; i < 48; i++)
                            if
(this.TableSemana.Rows[i].Cells[j].ID.Equals(pulsada))
                                {
                                    return d;
                                }
                            else
                                {
                                    d = d.AddMinutes(30);
                                }
                        }
                    break;
                case "dia":
                    for (int i = 0; i < 48; i++)
                    {
                        if
(this.TableSemana.Rows[i].Cells[1].ID.Equals(pulsada))
                            {
                                return d;
                            }
                        else
                            {
                                d = d.AddMinutes(30);
                            }
                    }
                    break;
            }
            return d;
        }
    }

```

EXPLICACIÓN

Identifica la celda pulsada y la devuelve valiéndose de que al crear la tabla se ha añadido un identificador numérico a cada celda que se puede leer para comprobar que columna y fila se ha pulsado.

FIN EXPLICACIÓN

```
/*carga el estado guardado de la pagina principal*/
private void carga_estado_pagina()
{
    Estado_principal est =
    (Estado_principal)this.Session.Contents["estado_principal"];
    //this.TreeViewMenu = est.menu;
    switch (est.vista)
    {
        case "dia":
            this.TableSem.Visible = true;
            this.PanelVistaSemana.Visible = true;
            this.Calendar1.Visible = false;
            if (this.TreeViewMenu.Nodes[0].Nodes.Count == 4)
            {
                this.CreaNodoHorario();
                this.CreaNodoGrano();
            }
            break;
        case "semana":
            this.TableSem.Visible = true;
            this.PanelVistaSemana.Visible = true;
            this.Calendar1.Visible = false;
            if (this.TreeViewMenu.Nodes[0].Nodes.Count == 4)
            {
                this.CreaNodoHorario();
                this.CreaNodoGrano();
            }
            break;
        case "semanalaboral":
            this.TableSem.Visible = true;
            this.PanelVistaSemana.Visible = true;
            this.Calendar1.Visible = false;
            if (this.TreeViewMenu.Nodes[0].Nodes.Count == 4)
            {
                this.CreaNodoHorario();
                this.CreaNodoGrano();
            }
            break;
        case "mes":
            this.TableSem.Visible = false;
            this.PanelVistaSemana.Visible = false;
            this.Calendar1.Visible = true;
            this.Calendar1.SelectedDate = est.dia;
            if (this.TreeViewMenu.Nodes[0].Nodes.Count == 6)
            {
                this.TreeViewMenu.Nodes[0].Nodes[5].Remove();
                this.TreeViewMenu.Nodes[0].Nodes[4].Remove();
            }
            break;
    }
}
```

```

    }

    switch (est.panel_visible)
    {
        case "Mensajes":
            this.TableMensajes.Visible = true;
            this.PanelTablas.Visible = true;
            this.TableReunion.Visible = false;
            this.TableTareas.Visible = false;
            this.PanelTareaR.Visible = false;
            this.PanelReunionR.Visible = false;
            break;
        case "Reuniones":
            this.TableMensajes.Visible = false;
            this.PanelTablas.Visible = true;
            this.TableReunion.Visible = true;
            this.TableTareas.Visible = false;
            this.PanelTareaR.Visible = false;
            this.PanelReunionR.Visible = false;
            break;
        case "Tareas":
            this.TableMensajes.Visible = false;
            this.PanelTablas.Visible = true;
            this.TableReunion.Visible = false;
            this.TableTareas.Visible = true;
            this.PanelTareaR.Visible = false;
            this.PanelReunionR.Visible = false;
            break;
        case "Anotar tarea":
            this.PanelTareaR.Visible = true;
            if (this.TextBoxAsuntoTR.Text == "")
            {
                this.ButtonGardarTareaR.Enabled = false;
            }
            this.LabelTareaR.Visible = false;
            this.PanelReunionR.Visible = false;
            this.PanelTablas.Visible = false;
            break;
        case "Convocar reunion":
            this.PanelReunionR.Visible = true;
            this.PanelTareaR.Visible = false;
            this.ButtonGuardarReunionR.Enabled = false;
            this.LabelRR.Visible = false;
            this.PanelTablas.Visible = false;
            break;
        case "ninguno":
            this.TableMensajes.Visible = false;
            this.PanelTablas.Visible = false;
            this.TableReunion.Visible = false;
            this.TableTareas.Visible = false;
            this.PanelTareaR.Visible = false;
            this.PanelReunionR.Visible = false;
            break;
    }
}
}

```

EXPLICACIÓN

Carga el estado de la página valiéndose de la variable de sesión "estado_principal" haciendo visibles o no visibles tablas de manera adecuada en función de la vista y el horario seleccionados.

FIN EXPLICACIÓN

```
protected void ButtonGardarTareaR_Click(object sender,
EventArgs e)
{
    Estado_principal est =
    (Estado_principal)this.carga_variable_memoria("estado_principal");
    string usuario = (string)this.Session.Contents["usuario"];
    DateTime dia_hora =
    (DateTime)this.Session.Contents["dia_y_hora_seleccionada"];
    string Fecha = dia_hora.Year.ToString() + "-" +
    dia_hora.Month.ToString() + "-" + dia_hora.Day.ToString();
    string Hora = dia_hora.TimeOfDay.Hours + ":" +
    dia_hora.TimeOfDay.Minutes + ":" + dia_hora.TimeOfDay.Seconds;
    String sqlquery = "insert into
    tarea(id_usuario_creador,Asunto,Fecha,Hora,Aviso_correo,Notas)
    Values(' " +
                                usuario + "','" +
    this.TextBoxAsuntoTR.Text + "','" + Fecha + "','" + Hora + "','" + 0 +
    "','"');";
    this.Inserta_base_Datos(sqlquery);
    this.ButtonGardarTareaR.Enabled = false;
    this.LabelTareaR.Text = "Tarea " +
    this.TextBoxAsuntoTR.Text + " añadida";
    this.TextBoxAsuntoTR.Text = "";
    this.LabelTareaR.Visible = true;
    this.Session.Contents.Remove("celda_seleccionada");
    this.Session.Contents.Remove("texto_boton_seleccionado");
    this.Page_Load(null, null);
}
```

EXPLICACIÓN

Guarda la tarea creada al pulsar sobre el menú tarea rápida. Para ello lee la fecha y la hora seleccionadas de una variable de sesión. Después de realizar la consulta sql para inserta la tarea borra las demás variables implicadas en la creación de una tarea, para que en futuras creaciones no interfieran con la siguiente tarea.

FIN EXPLICACIÓN

```

private DateTime dame_siguiente_fecha(string id_tarea, DateTime
actual)
{
    string my_sql = "SELECT Fecha, Hora, Tipo_periodicidad
FROM tarea WHERE (Id_tarea = '" + id_tarea + "')";

    DateTime d = new DateTime();
    OdbcConnection Conn = new OdbcConnection("Driver={MySQL
ODBC 3.51 Driver};Server=localhost;" +
        "Trusted_Connection=yes;Database=complumeeting");
    Conn.Open();
    OdbcCommand myCommand = new OdbcCommand(my_sql, Conn);
    OdbcDataReader dr = myCommand.ExecuteReader();
    dr.Read();
    if ((!dr.IsDBNull(0)) && (!dr.IsDBNull(1)))
    {
        if (!dr.IsDBNull(2))
        {
            DateTime d1 = dr.GetDate(0);
            d1 = d1.Add(dr.GetTime(1));
            string tipo_periodicidad = dr.GetString(2);
            switch (tipo_periodicidad)
            {
                case "Fija":
                    d =
this.dame_siguiente_fecha_periodicidad_fija(id_tarea, d1, actual);
                    break;
                case "semanalmente":
                    d =
this.dame_siguiente_fecha_periodicidad_semanal(id_tarea, d1, actual);
                    break;
                case "mensualmente":
                    d =
this.dame_siguiente_fecha_periodicidad_mensual(id_tarea, d1, actual);
                    break;
                case "anualmente":
                    d =
this.dame_siguiente_fecha_periodicidad_anual(id_tarea, d1, actual);
                    break;
            }
        }
        else
        {
            d = dr.GetDate(0);
            d = d.Add(dr.GetTime(1));
        }
    }

    dr.Close();
    Conn.Close();
    return d;
}

```

EXPLICACIÓN

Dada una tarea periódica identificada por `id_tarea`, y una fecha(actual) nos devuelve la ocurrencia más próxima después de la fecha actual de la tarea periódica.

Si la tarea no es periódica, devuelve su fecha original, y si la fecha actual es posterior a la fecha de finalización también devuelve la fecha original. Devuelve null si no tiene fecha de vencimiento. Hay distintos tipos de periodicidad como periodicidad de frecuencia fija, semanal, mensual y anual. A continuación se detalla la función que obtiene la siguiente fecha para el tipo de periodicidad de frecuencia fija, las demás no se detallarán por ser similares.

FIN EXPLICACIÓN

```
/*Dada una tarea de frecuencia fija, devuelve la proxima fecha
de vencimiento,
 * su comportamiento es igual al de
dame_siguiente_fecha(id_tarea) */
private DateTime dame_siguiente_fecha_periodicidad_fija(string
id_tarea, DateTime primera_fecha, DateTime actual)
{
    string my_sql = "SELECT Duracion,ultima_repeticion FROM
periodicidad_tarea_frec_fija WHERE (id_tarea = '" + id_tarea + "')";
    DateTime d = primera_fecha;
    OdbcConnection Conn = new OdbcConnection("Driver={MySQL
ODBC 3.51 Driver};Server=localhost;" +
        "Trusted_Connection=yes;Database=complumeeting");
    Conn.Open();
    OdbcCommand myCommand = new OdbcCommand(my_sql, Conn);
    OdbcDataReader dr = myCommand.ExecuteReader();
    dr.Read();
    DateTime ultima_repeticion = (DateTime)dr.GetDate(1);
    if (ultima_repeticion.Equals(new DateTime()))
    {
        string duracion = (string)dr.GetString(0);
        switch (duracion)
        {
            case "dia":
                while (d < actual)
                {
                    d = d.AddDays(1);
                }
                break;
            case "mes":
                while (d < actual)
                {
                    d = d.AddMonths(1);
                }
                break;
            case "ano":
                while (d < actual)
                {
                    d = d.AddYears(1);
                }
        }
    }
}
```

```

        break;
    }
}
else
{
    if (ultima_repeticion < actual)
    {
        d = primera_fecha;
    }
    else
    {
        string duracion = (string)dr.GetString(0);
        switch (duracion)
        {
            case "dia":
                while (d < actual)
                {
                    d = d.AddDays(1);
                }
                if (d > ultima_repeticion)
                {
                    d = primera_fecha;
                }
                break;
            case "mes":
                while (d < actual)
                {
                    d = d.AddMonths(1);
                }
                if (d > ultima_repeticion)
                {
                    d = primera_fecha;
                }
                break;
            case "ano":
                while (d < actual)
                {
                    d = d.AddYears(1);
                }
                if (d > ultima_repeticion)
                {
                    d = primera_fecha;
                }
                break;
        }
    }
}
dr.Close();
Conn.Close();
return d;
}

```

EXPLICACIÓN

Lo primero que hace la función es obtener de la base de datos la duración y la fecha de la última repetición de la tarea. Si no se ha repetido todavía la tarea se devuelve la siguiente fecha dependiendo del tipo de duración. Si la tarea ya se había repetido pueden ocurrir dos cosas:

1 → Que la tarea ya se había repetido antes de la fecha actual. En ese caso se devuelve una fecha vacía (new DateTime()).

2 → en otro caso se calcula la siguiente fecha sumando a la fecha inicial la duración determinada.

FIN EXPLICACIÓN

```
/*comprueba que existe el usuario pasado como parametro en la
base de datos*/
private bool existe_usuario(string id_usuario)
{
    string my_sql = "SELECT count(*) FROM usuario where
Id_usuario = '" + id_usuario + "'";

    OdbcConnection Conn = new OdbcConnection("Driver={MySQL
ODBC 3.51 Driver};Server=localhost;" +
        "Trusted_Connection=yes;Database=complumeeting");
    Conn.Open();
    OdbcCommand myCommand = new OdbcCommand(my_sql, Conn);
    Int64 num = (Int64)myCommand.ExecuteScalar();
    Conn.Close();
    if (num == 0)
    {
        return false;
    }
    else
    {
        return true;
    }
}
```

EXPLICACIÓN

Esta función accediendo a la base de datos indica si el usuario pasado como parámetro existe realmente en la base de datos.

FIN EXPLICACIÓN

Fin página Principal-----

6. Diseño de la base de datos

Descripción de conceptos:

Entidad	Descripción	Atributos	Clave
Usuario	Usuarios del sistema	Id_usuario Nombre Apellidos Contraseña Organización* Dirección_personal Dirección_trabajo Teléfono personal* Teléfono trabajo* Teléfono móvil* Fax* Administrador Restricciones	Id_usuario
E-mail	Direcciones de e-mail de los usuarios del sistema	e-mail	e-mail
Grupo _publico	Conjunto de usuarios relacionados accesible por todo el mundo	Nombre	Nombre
Grupo _privado (débil)	Conjunto de usuarios relacionados accesible por el creador	Nombre	Nombre Id_usuario
Tarea (débil)	Tarea que le interesa recordar a un usuario	Asunto Aviso_correo(si,no) Notas Fecha Hora Aviso(minutos antes de la hora) Tipo_periodicidad	Id_usuario Asunto Fecha Hora
Reunión(débil)	Reuniones entre usuarios	Asunto Descripción Aviso_correo(si,no) Fecha_ini Fecha_fin Aviso(minutos antes de la hora)	Id_usuario Asunto Fecha_ini Hora_ini

		Tipo_periodicidad	
Periodicidad	Representa la periodicidad de una tarea o una reunión	Ultima_repeticion	Ultima_repeticion
Periodicidad_Frec_Fija	Representa la periodicidad fija	Duración	Duración
Periodicidad_Semanal	Periodicidad semanal	Día Intervalo	Día Intervalo
Periodicidad_Mensual	Periodicidad mensual	Cardinal Día Intervalo	Cardinal Día Intervalo
Periodicidad_anual	Periodicidad anual	Cardinal Día Mes	Cardinal Día Mes
Archivo _tarea (débil)	Archivos adjuntos de datos de una tarea	Nombre _ archivo Datos Tamaño	Id_usuario Asunto Nombre _ archivo
Archivo _reunión (débil)	Archivos adjuntos de datos de una reunión	Nombre _ archivo Datos Tamaño	Id_usuario Asunto Nombre _ archivo

Relación	Descripción	Entidades involucradas	Atributos
Periodicidad_Tarea	Las tareas pueden tener periodicidad	Usuario Periodicidad	
Periodicidad_Reunión	Las reuniones pueden tener periodicidad	Usuario Periodicidad	
Email_usuario	Los usuarios pueden tener e-mail	Usuario E-mail	
Contactos	Los usuarios tienen asociados contactos	Usuario(Principal) Usuario(Contacto)	
Mensaje	Mensaje de texto de un usuario a otro, o de la aplicación a un usuario	Usuario(Emisor) Usuario(Receptor)	Texto Fecha
Creador _ grupo _público	Usuario creador de un determinado grupo público	Usuario Grupo _público	
Pertenece _grupo _público	Los usuarios pertenecen a determinados grupos públicos	Usuario Grupo _público	
Creador _ grupo	Usuario creador	Usuario	

_privado	de un determinado grupo privado	Grupo _privado	
Pertenece _grupo _privado	Los usuarios pertenecen a determinados grupos privados	Usuario Grupo _privado	
Archivos adjuntos tarea	Los archivos adjuntos de una tarea	Tarea Archivo _tarea	
Creador _ tarea	El usuario que creó una determinada tarea	Tarea Usuario	
Convocado	Usuarios que están convocados a una reunión	Reunión Usuario	
Creador _ reunión	Usuario que convoca la reunión	Reunión Usuario	
Archivos adjuntos reunión	Los archivos adjuntos de una reunión	Reunión Archivo _reunión	

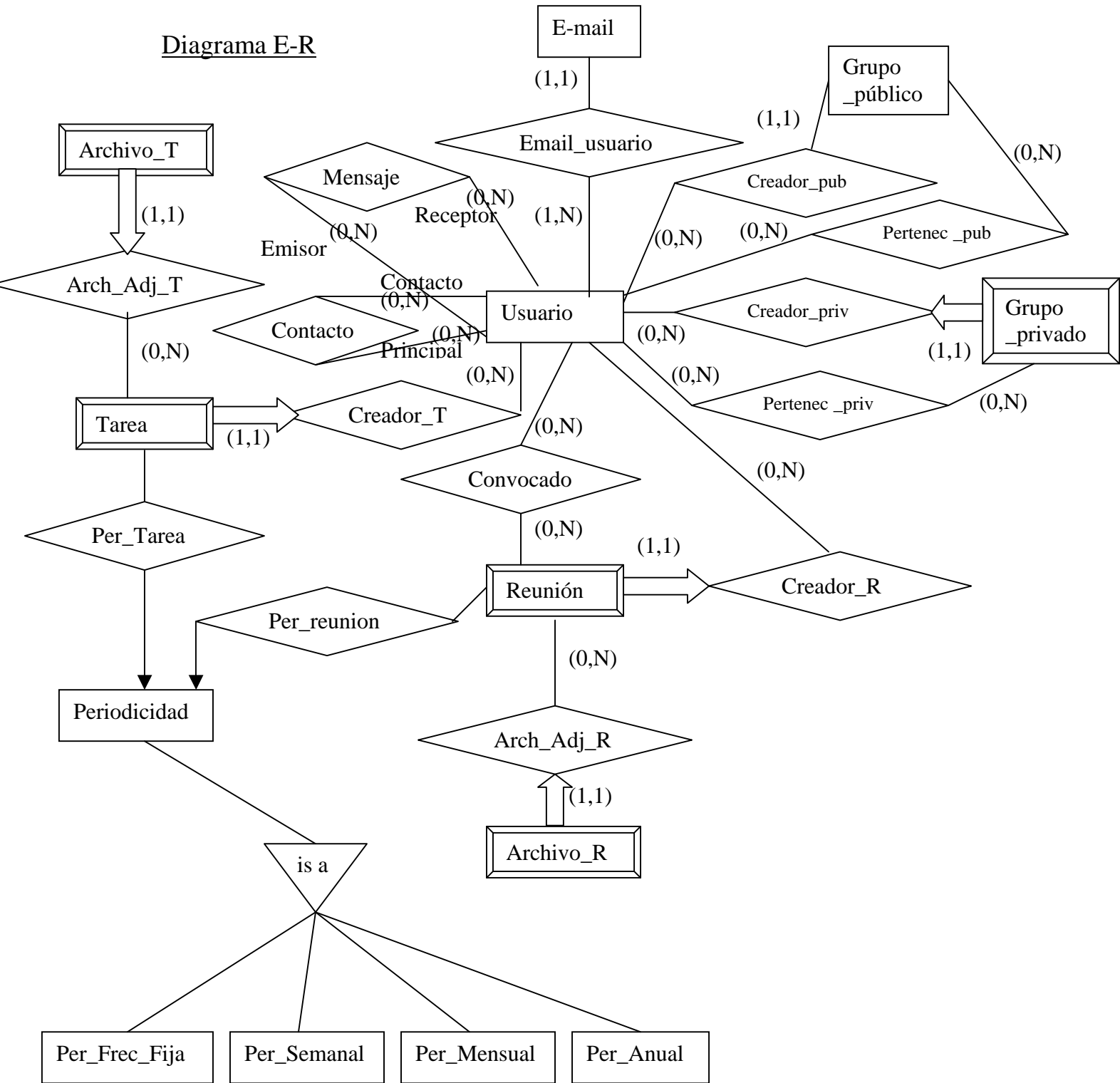
Derivaciones:

Rn1	A partir de la fecha, la hora y la periodicidad de una tarea se pueden deducir todas las fechas de esa tarea
Rn2	A partir de la fecha, la hora y el aviso de una tarea se puede deducir la fecha y la hora a la que se avisara
Rn3	A partir de la fecha, la hora y el aviso de una reunión se puede deducir la fecha y la hora a la que se avisara
Rn4	Con la fecha, la hora y la duración de una reunión se puede deducir la fecha y la hora de finalización de esa reunión

Restricciones :

Rn5	El tamaño de los archivos adjuntos de una tarea será limitado
Rn6	El tamaño de los archivos adjuntos de una reunión será limitado
Rn7	Cada reunión y cada tarea de un usuario debe corresponderse con una restricción de un determinado tipo para ese usuario

Diagrama E-R



Las entidades débiles cogen su clave de las relaciones marcadas con

Traducción al modelo relacional:

Usuario(id_usuario, Nombre, Apellidos, Contraseña, Organización,
Dirección_personal, Dirección_trabajo, Teléfono_personal, Teléfono_trabajo,
Teléfono_móvil, Fax, Administrador)

E-mail(e-mail, id_usuario)

Restricciones(id_usuario, Tipo, Inicio_rango, Fin_rango)

Grupo_publico(Nombre, id_usuario_creador)

Grupo_privado(Nombre, id_usuario_creador)

Contacto(id_usuario_principal, id_usuario_contacto)

Tarea(id_tarea, id_usuario_creador, Asunto, Fecha, Hora, Aviso_correo, Notas,
Aviso, Tipo_periodicidad)

Reunión(id_reunion, id_usuario_creador, Asunto, Fecha_ini, Fecha_fin,
Descripción, Aviso_correo, Aviso, Tipo_periodicidad)

Rango_fechas_reunion(id_reunion, fecha_inicio, hora_inicio, fecha_fin, hora_fin,
duracion)

Archivo_Tarea(id_archivo, id_tarea, Nombre_archivo, Datos, Tamaño)

Archivo_Reunión(id_archivo, id_reunion, Nombre_archivo, Datos, Tamaño)

Mensaje(id_mensaje, id_usuario_emisor, id_usuario_receptor, Texto, Fecha)

Pertenece_grupo_público(id_usuario, Nombre_grupo)

Pertenece_grupo_privado(Nombre_grupo, id_usuario_creador, id_usuario)

Convocado(id_reunion, id_usuario)

Periodicidad_Tarea_Frec_Fija(id_tarea, Duración, ultima_repeticion)

Periodicidad_Tarea_Semanalmente(id_tarea, Día, Intervalo, ultima_repeticion)

Periodicidad_Tarea_Mensualmente(id_tarea, Cardinal, Día, Intervalo,
ultima_repeticion)

Periodicidad_Tarea_Anualmente(id_tarea, Cardinal, Día, Mes, ultima_repeticion)

Periodicidad_Reunión_Frec_Fija(id_reunion, Duración, ultima_repeticion)

Periodicidad_Reunión_Semanalmente(id_reunion, Día, Intervalo, ultima_repeticion)

Periodicidad_Reunión_Mensualmente(*id_reunion, Cardinal, Día, Intervalo, ultima_repeticion*)

Periodicidad_Reunión_Anualmente(*id_reunion, Cardinal, Día, Mes, ultima_repeticion*)

En las restricciones, el inicio del rango y el fin del rango, son el rango de fechas entre los que tenemos un determinado tipo de restricción.

Para representar la periodicidad se han utilizado varias tablas que almacenan los diversos tipos de periodicidad que podemos encontrar en la aplicación. El campo tipo_periodicidad puede tener los valores Fija, semanalmente, mensualmente, anualmente, con lo que sabremos en cual de las otras tablas tenemos almacenada la periodicidad.

En la tabla Frec_Fija el campo duración es el intervalo de repetición de la tarea / reunión que podrá ser día, mes o año. Ej. Mensualmente en el día y la hora fijadas.

En la tabla Semanalmente se repetirá los “días” (Lunes, Martes, ...) de cada “Intervalo” semanas. Ej. Los lunes y martes de cada 2 semanas.

En la tabla mensualmente, se repetirá el “cardinal”(Primer, segundo, tercer, cuarto, ultimo) “día” (Lunes, Martes, ...) de cada “Intervalo” meses. Ej. El primer miércoles de cada 2 meses.

En la tabla anualmente, se repetirá el “cardinal”(Primer, segundo, tercer, cuarto, ultimo) “día” (Lunes, Martes, ...) del mes “Mes”. Ej. El ultimo Lunes de Marzo.

Dependencias funcionales:

Tarea:

id_tarea → id_usuario_creador, Asunto, Fecha, Hora

id_usuario_creador, Asunto, Fecha, Hora es superclave

Reunión:

id_reunion → id_usuario_creador, Asunto, Fecha, Hora

id_usuario_creador, Asunto, Fecha, Hora es superclave

Tras examinar las dependencias funcionales presentes en el diseño, podemos concluir que el grado de formalización de este esquema es la forma normal de Boyce-Codd, ya que todas las dependencias funcionales están cubiertas por las claves o por superclaves en el caso de la tabla Tarea y la tabla Reunión

Restricciones de integridad referencial

E-mail(*id_usuario*) \subseteq Usuario(*id_usuario*)

Restricciones(*id_usuario*) \subseteq Usuario(*id_usuario*)

Grupo_publico(*id_usuario_creador*) \subseteq Usuario(*id_usuario*)

Grupo_privado(*id_usuario_creador*) \subseteq Usuario(*id_usuario*)

Tarea(*id_usuario_creador*) \subseteq Usuario(*id_usuario*)

Reunión(*id_usuario_creador*) \subseteq Usuario(*id_usuario*)

Archivo_Tarea(*id_tarea*) \subseteq Tarea(*id_tarea*)

Archivo_Reunión(*id_reunión*) \subseteq Reunión(*id_reunión*)

Mensaje(*id_usuario_emisor*) \subseteq Usuario(*id_usuario*)

Mensaje(*id_usuario_receptor*) \subseteq Usuario(*id_usuario*)

Pertenece_grupo_público(*id_usuario*) \subseteq Usuario(*id_usuario*)

Pertenece_grupo_público(*Nombre_grupo*) \subseteq Grupo_publico(*Nombre*)

Pertenece_grupo_privado(*Nombre_grupo*) \subseteq Grupo_privado(*Nombre*)

Pertenece_grupo_privado(*id_usuario_creador*) \subseteq Usuario(*id_usuario*)

Pertenece_grupo_privado(*id_usuario*) \subseteq Usuario(*id_usuario*)

Convocado(*id_reunion*) \subseteq Reunión(*id_reunión*)

Convocado(*id_usuario*) \subseteq Usuario(*id_usuario*)

Periodicidad_Tarea_Frec_Fija(*id_tarea*) \subseteq Tarea(*id_tarea*)

Periodicidad_Tarea_Semanalmente(*id_tarea*) \subseteq Tarea(*id_tarea*)

Periodicidad_Tarea_Mensualmente(*id_tarea*) \subseteq Tarea(*id_tarea*)

Periodicidad_Tarea_Anualmente(*id_tarea*) \subseteq Tarea(*id_tarea*)

Periodicidad_Reunión_Frec_Fija(*id_reunion*) \subseteq Reunión(*id_reunion*)

Periodicidad_Reunión_Semanalmente(*id_reunion*) \subseteq Reunión(*id_reunion*)

Periodicidad_Reunión_Mensualmente(*id_reunion*) \subseteq Reunión(*id_reunion*)

Periodicidad_Reunión_Anualmente(*id_reunion*) \subseteq Reunión(*id_reunion*)

Diseño físico de la base de datos

En el diseño físico se usa la sintaxis de mysql.

Creación de las tablas:

Create table usuario (

Id_usuario char(50),
Nombre char(50) not null,
Apellidos char(50) not null,
Contrasena char(20) not null,
Organizacion char(50),
Direccion_personal Text,
Direccion_trabajo Text,
Telefono_personal char(9),
Telefono_trabajo char(9),
Telefono_movil char(9),
Fax char(9),
Administrador boolean not null,
Primary key (Id_usuario)

);

Create table E_mail(

E_mail char(50),
id_usuario char(50) references usuario(Id_usuario),
Primary key(e_mail)

);

Create table Restricciones(

id_usuario char(50) references usuario(Id_usuario),
Tipo enum('libre', 'preferible', 'asumible', 'ocupado'),
Inicio_rango DateTime,
Fin_rango DateTime,
Primary key (Id_usuario, Tipo, Inicio_rango, Fin_rango)

);

```
Create table Grupo_publico(  
Nombre char(50),  
id_usuario_creador char(50) references usuario(Id_usuario),  
    Primary key (Nombre)  
);
```

```
Create table Grupo_privado(  
Nombre char(50),  
id_usuario_creador char(50) references usuario(Id_usuario),  
    Primary key (Nombre, id_usuario_creador)  
);
```

```
Create table Contacto(  
id_usuario_principal char(50) references usuario(Id_usuario),  
id_usuario_contacto char(50) references usuario(Id_usuario),  
);
```

```
Create table Tarea(  
    Id_tarea int auto_increment,  
    id_usuario_creador char(50) references usuario(Id_usuario),  
    Asunto text,  
    Fecha Date,  
    Hora Time,  
    Aviso_correo boolean,  
    Notas text,  
    Aviso Time,  
    Tipo_periodicidad enum('Fija', 'semanalmente', 'mensualmente',  
'anualmente'),  
    Primary key(Id_tarea)  
);
```

```
Create table reunion(  
    Id_reunion int auto_increment,  
    id_usuario_creador char(50) references usuario(Id_usuario),  
    Asunto text,  
    Fecha_ini DateTime,  
    Fecha_fin DateTime,  
    Descripcion text,  
    Aviso_correo boolean,  
    Aviso Time,  
    Tipo_periodicidad enum('Fija', 'semanalmente', 'mensualmente',  
'anualmente'),  
    Primary key(Id_reunion)  
);
```

```
create table rango_fechas_reunion(  
    Id_reunion int,  
    Fecha_inicio date,  
    Fecha_fin date,
```

```

        Hora_inicio time,
        Hora_fin time,
        Duracion time,
        Primary key(Id_reunion)
);

create table archivo_tarea(
    Id_archivo int auto_increment,
    Id_tarea int references Tarea(Id_tarea),
    Nombre_archivo char(50),
    Datos longblob,
    Tamano int,
    Primary key (Id_archivo)
);

create table archivo_reunion(
    Id_archivo int auto_increment,
    Id_reunion int references Reunion(Id_reunion),
    Nombre_archivo char(50),
    Datos longblob,
    Tamano int,
    Primary key (Id_archivo)
);

create table Mensaje(
    Id_mensaje int auto_increment,
    Id_usuario_emisor char(50) references usuario(Id_usuario),
    Id_usuario_receptor char(50) references usuario(Id_usuario),
    Texto Text,
    Fecha DateTime,
    Primary key (Id_mensaje)
);

create table Pertenece_grupo_publico(
    id_usuario char(50) references Usuario(id_usuario),
    Nombre_grupo char(50) references Grupo_publico(Nombre),
    Primary key (id_usuario, Nombre_grupo)
);

create table Pertenece_grupo_privado(
    id_usuario_creador char(50) references Usuario(id_usuario),
    Nombre_grupo char(50) references Grupo_privado(Nombre),
    id_usuario char(50) references Usuario(id_usuario),
    Primary key (id_usuario, Nombre_grupo, id_usuario_creador)
);

create table Convocado(
    Id_reunion int references Reunion(Id_reunion),
    id_usuario char(50) references Usuario(id_usuario),
    primary key(Id_reunion, id_usuario)

```

);

```
create table Periodicidad_Tarea_Frec_Fija(  
    id_tarea int references Tarea(Id_tarea),  
    Duracion enum('dia', 'mes', 'ano') not null,  
    ultima_repeticion DateTime,  
    Primary key(id_tarea)
```

);

```
create table Periodicidad_Tarea_Semanalmente(  
    id_tarea int references Tarea(Id_tarea),  
    Dia enum('lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado',  
'domingo'),  
    Intervalo int,  
    ultima_repeticion DateTime,  
    Primary key(id_tarea, Dia)
```

);

```
create table Periodicidad_Tarea_Mensualmente(  
    id_tarea int references Tarea(Id_tarea),  
    Cardinal enum('Primer', 'segundo', 'tercer', 'cuarto', 'ultimo'),  
    Dia enum('lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado',  
'domingo'),  
    Intervalo int,  
    ultima_repeticion DateTime,  
    Primary key(id_tarea)
```

);

```
create table Periodicidad_Tarea_Anualmente(  
    id_tarea int references Tarea(Id_tarea),  
    Cardinal enum('Primer', 'segundo', 'tercer', 'cuarto', 'ultimo'),  
    Dia enum('lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado',  
'domingo'),  
    Mes enum('enero', 'febrero', 'marzo', 'abril', 'mayo', 'junio', 'julio', 'agosto',  
'septiembre', 'octubre', 'noviembre', 'diciembre'),  
    ultima_repeticion DateTime,  
    Primary key(id_tarea)
```

);

```
create table Periodicidad_Reunion_Frec_Fija(  
    id_Reunion int references Reunion(Id_Reunion),  
    Duracion enum('dia', 'mes', 'ano') not null,  
    ultima_repeticion DateTime,  
    Primary key(id_Reunion)
```

);

```
create table Periodicidad_Reunion_Semanalmente(  
    id_Reunion int references Reunion(Id_Reunion),
```

```

        Dia enum('lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado',
'domingo'),
        Intervalo int,
        ultima_repeticion DateTime,
        Primary key(id_Reunion, Dia)
);

create table Periodicidad_Reunion_Mensualmente(
    id_Reunion int references Reunion(Id_Reunion),
    Cardinal enum ('Primer', 'segundo', 'tercer', 'cuarto', 'ultimo'),
    Dia enum('lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado',
'domingo'),
    Intervalo int,
    ultima_repeticion DateTime,
    Primary key(id_Reunion)
);

create table Periodicidad_Reunion_Anualmente(
    id_Reunion int references Reunion(Id_Reunion),
    Cardinal enum ('Primer', 'segundo', 'tercer', 'cuarto', 'ultimo'),
    Dia enum('lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado',
'domingo'),
    Mes enum('enero', 'febrero', 'marzo', 'abril', 'mayo', 'junio', 'julio', 'agosto',
'septiembre', 'octubre', 'noviembre', 'diciembre'),
    ultima_repeticion DateTime,
    Primary key(id_Reunion)
);

```

7. Problemas resueltos

A la hora de programar la aplicación aparecieron una serie de problemas que por su representatividad pueden ayudar a futuros programadores que se basen en las mismas tecnologías. La mayor parte de estos problemas se debieron al desconocimiento de las tecnologías que se estaban usando o al mal uso de éstas.

Se citarán una serie de problemas y las soluciones aportadas por el equipo de desarrollo de Complu – Meeting:

1º Navegación entre páginas : Para desplazarse de una página a otra por la aplicación en algunos casos era necesario almacenar valores de unas páginas a otras. Al principio, se resolvió pasando parámetros de forma tosca entre los métodos "Page_Load()" de cada página que son los primeros eventos de carga de cada página. Sin embargo, daba problemas y se optó por utilizar variables de aplicación. De esta forma se guardaban pares (clave, variable) que permanecían inalterables durante la ejecución de la aplicación. Aunque el problema parecía estar resuelto totalmente se descubrió que no a la hora de hacer pruebas con varios usuarios accediendo a la aplicación simultáneamente desde distintos equipos. El problema era que se confundían las variables porque al guardarse en el servidor eran iguales para todos los clientes. Finalmente se optó por utilizar variables de sesión que realizaban la misma función que las de la aplicación salvo que eran personalizadas para cada sesión en la aplicación, es decir, cuando dos usuarios accedían a la aplicación desde distintos equipos, al tener distintas sesiones abiertas, sus variables de sesión eran distintas y no se interferían mutuamente.

2º Recarga del estado de la página: Como ya se ha visto anteriormente, muchas de las páginas de la aplicación tienen partes fijas y partes dinámicas o cambiantes. Las partes fijas no dan ningún problema a la hora de recargar la página entre cada transferencia del servidor, pero las dinámicas sí, porque dependiendo de la acción a ejecutar tienen uno u otro aspecto. El problema se resolvió inicialmente utilizando funciones que pintasen las partes dinámicas de las páginas dependiendo del estado de la aplicación que se guardaba en variables de sesión. En muchos casos en los que las partes dinámicas eran escasas se optó por este método, pero en otros como la página principal o la página de reunión donde las partes dinámicas eran extensas se adoptó otra medida similar. La medida consistía en crear clases que guardasen atributos de todos los elementos configurables y cargarlas y guardarlas todas de golpe a través del uso de una única variable de sesión.

3º La propiedad AutoPostBack: Esto no es un problema sino una advertencia útil a futuros programadores. Algunos controles de servidor como los menús desplegables, los checkbox y otros, disponen de una propiedad muy importante llamada AutoPostBack. El funcionamiento de la propiedad es el siguiente :

- Si `AutoPostBack = true` se produce una devolución de datos automática al servidor cuando el usuario presiona una de las teclas TAB o ENTRAR mientras está en el control.
- Si `AutoPostBack = false` no se produce la devolución de datos automática.

De esta forma, si se quiere que cada pulsación sobre un control determinado ocasione una recarga automática, es necesario activar a true esta propiedad. En otro caso puede ocurrir que los cambios en el control no se reflejen en futuras recargas del mismo.

8. Bibliografía

- Manual de usuario de ASP.NET.
- Manual de usuario de MySQL 5.0.
- Introducing Microsoft ASP.NET 2.0. Published by Microsoft.
- Professional C# 2nd Edition. Simon Robinson, K. Scott Allen, Ollie Cornes, Jay Glynn, Zach Greenvoss, Burton Harvey, Christian Nagel, Morgan Skinner, Karli Watson.
- Microsoft programing ASP.NET.
- MySQL: the definitve guide to using proraming and administering MySQL 4.1 and 5.0. Paul Dubois. Indianápolis (Indiana): Developers Library 2005
- Web database application with PHP and MySQL. Hugo Williams and David Lane. Sebastopol(California). O'Really 2004

“Autorizamos a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y el prototipo desarrollado”.

**David García Higuera
Pablo Langa Blanco
David Toro Córdoba**