



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicación web para la gestión de ofertas de alojamiento de la Universidad Cardenal Herrera

Proyecto Final de Carrera

ITIG

Código: DISCA-253

Autor: Pashyan, Yelena

Director: Lemus Zúñiga, Lenin Guillermo

Mayo 2014

Tabla de contenidos

1. Introducción	6
1.1 Objetivos	6
1.2 Motivación	6
1.3 Contexto	7
1.4 Estructura de la memoria	8
2. Especificación de requisitos (ERS)	9
2.1 Introducción.....	9
2.1.1 Propósito.....	9
2.1.2 Ámbito.....	9
2.1.3 Definiciones, siglas y abreviaturas	9
2.2 Descripción general	11
2.2.1 Perspectiva del producto.....	11
2.2.2 Funciones del producto	11
2.2.3 Características del usuario	13
2.2.4 Restricciones	13
2.2.5 Supuestos y dependencias	13
2.3 Requisitos específicos	14
2.3.1 Requisitos de interfaces externos.....	14
2.3.2 Requisitos funcionales	15
2.3.3 Requisitos de eficiencia	20
2.3.4 Restricciones de diseño	21
2.4 Atributos.....	21
2.4.1 Seguridad	21
2.4.2 Mantenimiento.....	21
2.4.3 Portabilidad	21
2.5 Otros requerimientos.....	21
3. ASP y .NET	24
3.1 ASP	24
3.2 .NET	25
3.2.1 MSIL, CIL y el código controlado.....	26
3.2.2 Basic Class Library.....	27
3.2.3 Assemblies	28

3.2.4. ViewState	29
3.3 Diferencias entre ASP y .NET.....	29
4. Análisis.....	32
4.1 Casos de uso	32
4.1.1 Caso de uso: Actor anónimo.....	33
4.1.2 Caso de uso: Actor propietario	34
4.1.3 Caso de uso: Actor administrador	35
4.2 Diagrama de clases.....	35
4.3 Diagrama de secuencias.....	37
4.3.1 Inicio de sesión	37
4.3.2 Crear oferta de alojamiento.....	38
4.3.3 Registro de usuarios.....	39
5. Diseño	40
5.1 Capa de presentación	41
5.1.1 Parte pública.....	41
5.1.2 Parte privada	46
5.2 Capa de negocio.....	50
5.3 Capa de datos.....	52
6. Implementación	53
6.1 Tecnologías de implementación web	53
6.2 Descripción de la implementación.....	55
6.2.1 Parte pública	55
6.2.2 Parte privada	59
7. Evaluación	63
7.1 Pruebas de Integración.....	63
7.2 Pruebas de sistema.....	63
8. Conclusiones	67
9. Bibliografía	68



Tabla de figuras

Figura 1: Web Fotocasa.....	7
Figura 2: Interfaz de usuario - Parte pública	14
Figura 3: Interfaz de usuario - Parte privada	15
Figura 4: Estructura ASP	24
Figura 5: Configuración de una aplicación Web con HTML estático	30
Figura 6: Configuración de una aplicación Web con HTML dinámico	30
Figura 7: Familia de tecnologías de la plataforma .NET	30
Figura 8: Funcionamiento de los eventos en la plataforma .NET	31
Figura 9: Actores del diagrama de casos de uso	32
Figura 10: Diagrama casos de uso - Actor anónimo	33
Figura 11: Diagrama casos de uso - Actor propietario	34
Figura 12: Diagrama casos de uso - Actor administrador	35
Figura 13: Diagrama de clases.....	36
Figura 14: Diagrama de secuencias - Inicio de sesión	37
Figura 15: Diagrama de secuencias - Creación oferta de alojamiento.....	38
Figura 16: Diagrama de secuencias - Registro de usuarios	39
Figura 17: Arquitectura de tres capas	40
Figura 18: Capa presentación - Página principal parte pública	42
Figura 19: Cabecera principal UCH-CEU	42
Figura 20: Cabecera página web - Alojamiento	42
Figura 21: Cuerpo de la página web	43
Figura 22: Pie de la página web.....	44
Figura 23: Página web - Oferta Alojamiento	44
Figura 24: Página web principal - Date de alta Gestiona tus ofertas	45
Figura 25: Página web - Date de alta	45
Figura 26: Página web de bienvenida	46
Figura 27: Página de validación - Usuario propietario.....	47
Figura 28: Página de validación - Usuario administrador	47
Figura 29: Intranet usuario propietario - Listado ofertas.....	48
Figura 30: Intranet usuario propietario - Sin ofertas.....	48
Figura 31: Intranet usuario propietario - Crear nueva oferta	49
Figura 32: Intranet usuario propietario - Mi cuenta	50
Figura 33: Capa de negocio	51
Figura 34: Diagrama entidad-relación	52
Figura 35: login.aspx.....	56
Figura 36:login.aspx.cs	56
Figura 37:ofertas.aspx.....	57
Figura 38:ofertas.aspx.cs.....	58
Figura 39:oferta.aspx	59
Figura 40:oferta.aspx.cs	59
Figura 41:oferta_alojamiento.aspx.....	60
Figura 42:oferta_alojamiento.aspx.cs	60
Figura 43:borrador_oferta.aspx	61
Figura 44:borrador_oferta.aspx.cs	62

Figura 45: Validación del código CSS	64
Figura 46: Google Chrome	65
Figura 47: Mozilla Firefox	65
Figura 48: Internet Explorer	66



1. Introducción

Este documento describe el trabajo realizado como Proyecto Final de Carrera de Ingeniería Técnica en Informática de Gestión, cursada en la Escuela Técnica Superior de Informática Aplicada de la Universidad Politécnica de Valencia.

En esta memoria se describen todas las fases del proyecto, desde la idea y la especificación de requisitos hasta la implementación y la ejecución por los usuarios finales.

El proyecto, titulado “Aplicación web para la gestión de ofertas de alojamiento de la Universidad Cardenal Herrera”, ha sido realizado por Yelena Pashyan y dirigido por Lenin Guillermo Lemus Zúñiga.

1.1 Objetivos

El objetivo de este Proyecto Final de Carrera es desarrollar una aplicación web dedicada a la gestión de ofertas de alojamiento, así como un espacio web en la página web de la Universidad CEU Cardenal Herrera dirigido principalmente a sus futuros alumnos y/o a sus alumnos de nuevo ingreso.

1.2 Motivación

En la Universidad CEU Cardenal Herrera, lugar donde se han llevado a cabo las prácticas formativas, en su sitio web dispone de una sección destinada fundamentalmente tanto al futuro alumno como al alumno de nuevo ingreso. Por ello, y pensando principalmente en la movilidad de los alumnos que tienen que acudir al campus de Moncada/Alfara pero atendiendo igualmente a los alumnos que vayan a estudiar a los campus de Valencia capital, Castellón y Elche, se ha pensado en desarrollar un espacio y una aplicación en el contexto de la web actual que permita a los alumnos visualizar las ofertas de alojamiento que mejor se ajusten a sus necesidades. De forma indirecta una de las finalidades del proyecto va a ser facilitarles a particulares que disponen de inmuebles y propiedades y que, al no disponer de un software sencillo y económico para gestionar sus inmuebles, acaban rindiéndose a inmobiliarias para facilitar sus trámites, así como hacer publicidad de estos inmuebles de forma gratuita.

Debo remarcar que el objetivo principal de este proyecto sigue centrándose en los alumnos de la universidad, evitando así la contratación de una inmobiliaria por parte de ellos. También añadir que con este servicio únicamente se pretende poner en contacto a los alumnos y a los propietarios sin hacerse responsable la universidad ni de las condiciones de los alojamientos ni de las condiciones de los contratos.

A nivel académico el objetivo del proyecto es iniciarse en el desarrollo de aplicaciones web.

Se ha pretendido proporcionar un entorno amigable, teniendo en cuenta el amplio abanico de perfiles de usuario, existiendo desde poco experimentados hasta profesionales.

La aplicación se ha implementado con las tecnologías que ofrece Microsoft en su Framework .NET (ASP .NET 4.0 Web Forms), así pues el desarrollo ha sido realizado con el entorno Microsoft Visual Studio 2010, el lenguaje de programación utilizado es C# y el sistema de gestión de bases de datos es Microsoft SQL Server 2008.

Para el uso de la aplicación se requiere una conexión a internet y un navegador web. También será necesario el conocimiento básico a nivel de usuario para el manejo de la aplicación.

1.3 Contexto

Esta aplicación web se ha realizado orientado a un sector inmobiliario, donde los usuarios, en este caso los alumnos de la universidad, podrán buscar pisos de alquiler, pisos compartidos, residencias universitarias, residencias familiares y colegios mayores.

Para lograr los objetivos marcados para la aplicación, se ha pensado en la necesaria existencia de diferentes tipos de usuarios con diversos privilegios, según el papel que tengan en la aplicación. Desde un control casi total sobre la gestión de los recursos de la aplicación para el administrador/moderador, pasando por la gestión de los inmuebles para los clientes, hasta la visibilidad de inmuebles disponibles, en el espacio dedicado a este fin en el sitio web de la universidad, para los usuarios anónimos.

Se ofrecerá una lista de servicios para tratar de cubrir los trámites de gestión de los inmuebles agregados a la aplicación. Entre estos servicios podemos destacar: Alta/baja/edición de usuarios, alta/baja/edición de inmuebles, servicio automático de envío de correos, etc.

Una de las webs inmobiliarias que he tomado como ejemplo es una de las más populares en el mercado: www.fotocasa.es.

A continuación se muestra una imagen de la web Fotocasa.



Figura 1: Web Fotocasa

El portal inmobiliario Fotocasa es muy conocido y tiene un elevado número de visitas.

Lo primero que aparece en el portal son los términos de búsqueda, y una vez definidas las características, se muestran los inmuebles con dichas características.

1.4 Estructura de la memoria

El presente documento está ordenado de forma que se puede resumir cada una de las fases del ciclo de vida de la evolución del proyecto.

- ✚ **Introducción:** Se realiza la presentación del proyecto, se incluyen conceptos como motivación, objetivos, contexto y estructura.
- ✚ **Especificación de requisitos:** Este apartado recoge los requisitos básicos que se demandan antes del inicio del proyecto.
- ✚ **ASP y .NET:** En este capítulo se realizará un breve estudio del framework .NET y de la tecnología ASP. Con este estudio se pretende obtener una perspectiva del funcionamiento global de .NET y ASP, que permitirá abordar el desarrollo de la aplicación web objetivo.
- ✚ **Análisis:** Capítulo destinado a la obtención de un modelo conceptual de la aplicación mediante la ayuda de diagramas UML.
- ✚ **Diseño:** Describirá los diseños realizados para el proyecto así como la metodología utilizada en el proceso.
- ✚ **Implementación:** Se verá una descripción de las tecnologías y herramientas utilizadas durante la fase de implementación. Se describirá el funcionamiento de los componentes implementados en la solución.
- ✚ **Evaluación:** Exposición de las técnicas utilizadas en la evaluación de la aplicación y el resultado obtenido.
- ✚ **Conclusiones:** Valoración personal sobre el desarrollo del proyecto y descripción del trabajo realizado.
- ✚ **Bibliografía:** Documentación utilizada para el desarrollo.

2. Especificación de requisitos (ERS)

En este capítulo se recogen los requisitos básicos que se demandan antes del inicio del proyecto.

2.1 Introducción

2.1.1 Propósito

El propósito de la especificación de requisitos, es establecer las características que requiere la aplicación, desarrollar su funcionalidad y satisfacer las necesidades de los diferentes usuarios.

Esta especificación va dirigida a toda persona interesada en el funcionamiento de la aplicación. Se seguirán las directrices IEEE 830-1998, uno de los estándares más referenciados para la especificación de requisitos en proyectos Web.

2.1.2 Ámbito

El producto software a describir es una aplicación web de gestión. Con todas las funcionalidades implementadas será posible llevar la gestión de inmuebles de propiedad privada a nivel particular y hacer publicidad de ellos, sin tener altos conocimientos ofimáticos. Para facilitar el tratamiento de datos se usará una base de datos relacional.

La aplicación da soporte a tres tipos de usuarios, mencionados anteriormente, y cada uno tiene acceso a una parte diferente. La aplicación consta de dos partes:

- **Parte pública:** Es accesible para todos los usuarios, es decir, tanto para los usuarios registrados como los no registrados, e incluye el listado de las diferentes ofertas de alojamiento que los usuarios registrados han creado previamente y que han sido publicadas por parte del moderador. Los usuarios no registrados sólo pueden acceder a ésta parte.
- **Parte privada:** Es accesible por los usuarios registrados y el usuario administrador. Éstos pueden realizar diferentes acciones según el tipo de usuario.

2.1.3 Definiciones, siglas y abreviaturas

Sitio web: Es un conjunto de documentos y páginas web que tratan de un mismo tema, con una página de inicio. Tiene un dominio y una dirección de internet específicos.

IEEE (Institute of Electrical and Electronics Engineers): Asociación de profesionales norteamericanos que aporta criterios de estandarización de dispositivos eléctricos y electrónicos.

Servidor Web: Es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web.

W3C (World Wide Web Consortium): es un consorcio internacional que produce recomendaciones para la World Wide Web.

Base de Datos: Una base de datos es una colección organizada de información. Ésta contiene una colección de registros que puede buscar, ordenar y analizar rápidamente.

Hardware: Conjunto de todas las partes tangibles de un sistema informático.

Software: Conjunto de los componentes lógicos necesarios de un sistema informático que hacen posible la realización de tareas específicas.

Interfaz: Parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario.

HTML (HyperText Markup Language): Lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes.

Formulario (HTML forms): Servicio que permite al usuario introducir datos los cuales son enviados a un servidor para ser procesados.

Navegador web: Software que permite al usuario recuperar y visualizar documentos de hipertexto desde servidores web a través de Internet.

ASP (Active Server Pages): una solución de software de Microsoft, para la programación en páginas web.

SQL (Structured Query Language): es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

SQL Server: sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

Registro: Es una descripción de un artículo en particular que se almacena en la base de datos. En una base de datos relacional, cada fila de cada tabla es un registro.

Autenticación o autenticación: proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse.

C#: lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

Sistema Operativo o SO (Operative System OS en inglés): Es un conjunto de programas de computadora destinado a permitir una administración eficaz de sus recursos, ejecutándose en modo privilegiado respecto de los restantes programas de aplicación. Comienza a trabajar cuando se enciende la computadora, y gestiona el hardware de la máquina desde los niveles más básicos, permitiendo también la interacción con el usuario.

2.2 Descripción general

2.2.1 Perspectiva del producto

Esta aplicación es independiente del sistema operativo utilizado y para poder acceder a usarla sólo se necesitará un navegador web y una conexión a internet.

Para ello será necesario un servidor web donde alojar la web además de un dominio para su direccionamiento.

2.2.2 Funciones del producto

Como he mencionado en uno de los apartados anteriores, son dos las funciones principales de la aplicación. Ésta se divide en dos partes y cada una de ellas tiene asignado un tipo determinado de usuario. A continuación se va a exponer en qué consisten las funciones que puede realizar cada uno de los usuarios.

Parte pública

Esta parte la puede ver cualquier usuario sin la necesidad de tener que registrarse antes. Las acciones que pueden llevar a cabo son las que se describen a continuación:

Usuario anónimo

- Visualizar ofertas de alojamiento publicadas en la web.
- Buscar inmuebles haciendo uso de los buscadores situados en la propia página web.
- Ponerse en contacto con propietarios de los inmuebles.
- Darse de alta como usuario.

Parte privada

Los usuarios que pueden tener acceso a esta parte de la aplicación tienen los mismos privilegios que el usuario anterior pero, además, los que se citan a continuación:

Usuario registrado

- Iniciar sesión a su espacio personal de usuario.
- Acceder a “Mi cuenta” para ver y/o modificar los datos personales de usuario.
- Darse de baja como usuario. En este caso se borraría de la base de datos tanto como el usuario como todas las ofertas de alojamiento registradas previamente que tenga dicho usuario.
- Visualizar a través de su espacio personal de usuario sus ofertas de alojamiento que han sido publicadas por el administrador.
- Crear nuevas ofertas de alojamiento.
- Modificar/eliminar ofertas de alojamiento si todavía no han sido publicadas por parte del administrador.
- Renovar las ofertas de alojamiento que estén publicadas y a punto de ser caducadas (cuando falten quince días para la fecha de caducidad).
- Cerrar la sesión de usuario saliendo de su espacio personal de usuario.

Usuario administrador

- Iniciar sesión.
- Visualizar el borrador de las ofertas que todavía no han sido publicadas.
- Filtrar/buscar ofertas de alojamiento.
- Publicar ofertas de alojamiento.
- Rechazar ofertas de alojamiento si no están publicadas.
- Eliminar ofertas de alojamiento.
- Cerrar sesión.

Sistema

- Abrir/cerrar sesión de usuario.
- Avisar mediante correo electrónico al usuario cuando éste se de de alta como usuario.
- Comprobar que, al darse de alta un usuario, éste último verifique su cuenta haciendo clic en un enlace que se le adjunta en el correo que se le envía al usuario cuando se da de alta.
- Avisar mediante correo electrónico al moderador cuando se registran ofertas de alojamientos nuevas.
- Avisar mediante correo electrónico al usuario cuando se publica/rechaza/elimina una oferta de alojamiento suya.
- Avisar al moderador cuando el usuario modifique una oferta rechazada.
- Comprobar si una oferta rechazada se modifica durante los cinco días siguientes al rechazo por parte del administrador. Si no se realiza la modificación durante ese plazo el sistema se encarga de eliminar la oferta correspondiente automáticamente.
- Al darse de baja un usuario realizar el borrado automático de todas las ofertas de alojamiento creadas por dicho usuario.
- Avisar al usuario mediante correo electrónico cuando falten cinco días y un día para que caduque alguna de sus ofertas de alojamiento.

- Eliminar automáticamente cuando una oferta de alojamiento caduca y avisar a dicho propietario mediante correo electrónico de que su oferta ha sido eliminada por haber caducado.

2.2.3 Características del usuario

Como se ha mencionado anteriormente, existen tres tipos de usuarios:

- **Usuario anónimo/no registrado:** Pueden ver y buscar ofertas de alojamiento creadas y publicadas en la web, contactar con propietarios y registrarse. Sólo tienen acceso a la parte pública de la aplicación.
- **Usuario registrado:** Estos usuarios tienen todos los privilegios del usuario no registrado y, además, tienen la posibilidad de modificar o consultar su propio perfil, crear/modificar/eliminar/renovar ofertas de alojamiento y visualizar sus ofertas de alojamientos ya publicadas a través de su espacio personal. Accede a la parte privada de la aplicación.
- **Usuario administrador:** Tiene los privilegios del usuario no registrado además de poder publicar/rechazar/eliminar las ofertas de alojamiento registradas por los propietarios y visualizarlas en forma de borrador antes de publicarlas. Cuenta con su propio espacio personal. Accede a la parte privada de la aplicación.

La aplicación ofrecerá unos servicios que por su complejidad podrán ser utilizados por cualquier persona, teniendo unos conocimientos mínimos de informática y de navegación por la web para poder manejar al sitio web.

2.2.4 Restricciones

Para acceder al portal web se deberá disponer del hardware necesario para conectarse a internet, siendo posible vía ordenador, PDA, móvil, Smartphone, etc.

Cualquier persona podrá visitar la página mediante un navegador web, pudiendo consultar información o acceder a las distintas funciones dependiendo del tipo de rol que tenga dicho usuario.

2.2.5 Supuestos y dependencias

La aplicación web requiere que los usuarios utilicen un navegador web para acceder a ella. No será necesario el uso de un navegador específico, pero estos navegadores deberán ser compatibles con el protocolo HTTP. Es aquí donde encontramos la única dependencia, con el servidor

El servidor donde se hospede nuestro sistema deberá admitir:

- Ejecución de código ASP .NET.
- Procesos transaccionales con bases de datos SQL Server.

2.3 Requisitos específicos

2.3.1 Requisitos de interfaces externos

- *Interfaces de usuario*

La aplicación tiene un diseño único para todos los usuarios, tal como se ve en las siguientes imágenes.

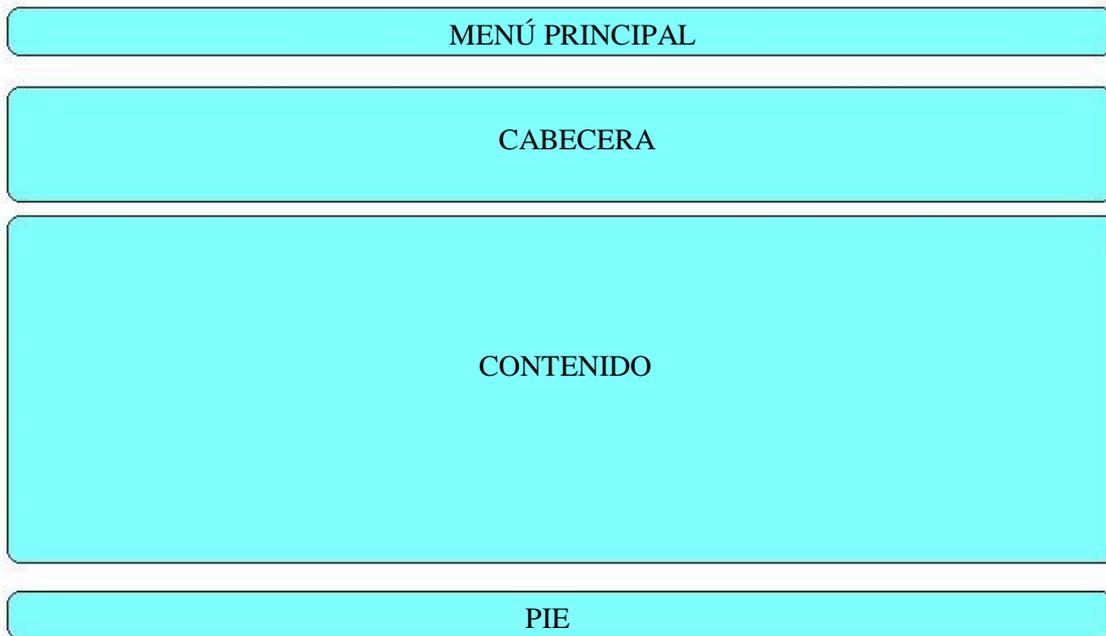


Figura 2: Interfaz de usuario - Parte pública

La parte pública y la parte privada difieren en cuanto al diseño.

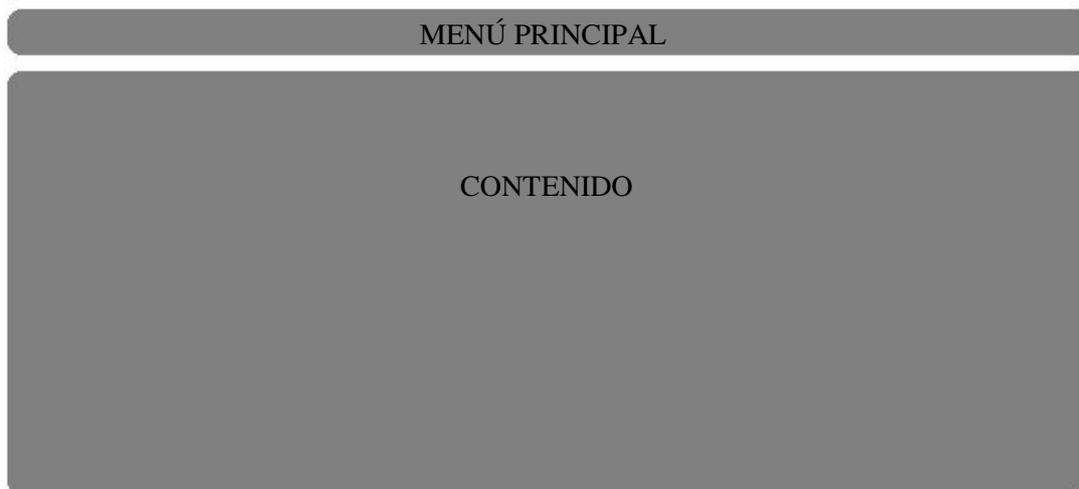


Figura 3: Interfaz de usuario - Parte privada

- *Interfaces de hardware*

Los usuarios deberán disponer de acceso a la red para poder acceder a la aplicación, además de un equipo que permita la ejecución de un navegador web.

Para el alojamiento de datos y de la aplicación se contará con un servicio de almacenamiento externo.

- *Interfaces de software*

La aplicación ha sido desarrollada para ejecutarse en cualquier sistema operativo y navegador web estándar.

- *Interfaces de comunicaciones*

Los interfaces de comunicación entre cliente y servidor son los estándares TCP/IP.

2.3.2 Requisitos funcionales

Usuario anónimo/no registrado

-Dar de alta como usuario:

- ✓ **Propósito:** Crear perfil de usuario.
- ✓ **Entrada:** Nombre, apellidos, teléfono, forma de contactar, correo electrónico, contraseña, código de comprobación.
- ✓ **Proceso:** Se almacenan los datos del usuario.

- ✓ **Salida:** Se visualiza un mensaje de agradecimiento durante diez segundos y posteriormente se visualiza el listado de ofertas de alojamiento publicadas en la web. Correo electrónico al usuario.

-Visualizar inmuebles:

- ✓ **Propósito:** Ver ofertas de alojamiento.
- ✓ **Entrada:** -
- ✓ **Proceso:** Se carga la web.
- ✓ **Salida:** Se visualizan todas las ofertas de alojamiento publicadas por otros propietarios y, si no las hay, se muestra un mensaje de que no hay anuncios.

-Buscar inmuebles:

- ✓ **Propósito:** Buscar inmuebles con la ayuda de los buscadores que hay en la página web.
- ✓ **Entrada:** Tipo de alojamiento, precio, provincia, localidad, filtro de ordenar por fecha de publicación/precio ascendente/precio descendente.
- ✓ **Proceso:** Se cargan los alojamientos que cumplen con las condiciones de la búsqueda.
- ✓ **Salida:** Se visualiza el resultado de la búsqueda.

-Contactar con propietarios:

- ✓ **Propósito:** Ponerse en contacto con propietarios de inmuebles.
- ✓ **Entrada:** Si el propietario del inmueble prefiere que contacten con él mediante correo electrónico, el usuario deberá introducir su nombre, correo electrónico, teléfono y el mensaje que quiere enviar al propietario junto con el código de comprobación que aparece en el formulario. Si el propietario prefiere que contacten con él mediante llamadas al número del teléfono que aparece en su anuncio, el usuario podrá ver el número de teléfono del propietario y hacerle una llamada. Si al propietario le es indiferente la manera en la que los interesados se pongan en contacto con él, el usuario puede elegir entre mandarle un correo electrónico introduciendo los datos que he mencionado anteriormente o hacerle una llamada telefónica.
- ✓ **Proceso:** En caso de que el propietario prefiera que contacten con él mediante llamadas telefónicas, se visualiza el número de teléfono del propietario. Si el propietario prefiere que le manden un correo electrónico, se recogen los datos del usuario introducidos en el formulario y se le manda un correo electrónico al propietario del anuncio.
- ✓ **Salida:** Dependiendo de la forma de contactar, le llega un correo electrónico al propietario del anuncio o una llamada telefónica por parte del usuario interesado.

Usuario registrado

Este usuario tiene los mismos privilegios del usuario anterior (a excepción registrarse, porque ya está registrado) y, además, los siguientes privilegios:

-Iniciar sesión:

- ✓ **Propósito:** Identificarse en la aplicación web.
- ✓ **Entrada:** Email, contraseña.
- ✓ **Proceso:** Verificar los credenciales introducidos.
- ✓ **Salida:** Si la verificación ha ido correctamente, ir al espacio personal del usuario.

-Consultar/modificar perfil:

- ✓ **Propósito:** Consultar/modificar los datos personales del usuario.
- ✓ **Entrada:** Datos personales.
- ✓ **Proceso:** Se recuperan los datos personales de la base de datos/se modifican los datos existentes por los nuevos introducidos.
- ✓ **Salida:** Se muestran los datos personales del usuario/se modifican los datos correctamente o falla la modificación.

-Darse de baja como usuario:

- ✓ **Propósito:** Eliminar perfil de usuario.
- ✓ **Entrada:** Identificador del usuario a eliminar.
- ✓ **Proceso:** Se eliminan de la base de datos tanto los datos del usuario como las ofertas de alojamiento que tenga creadas dicho usuario.
- ✓ **Salida:** Mensaje de confirmación o de error. Correo electrónico al usuario.

-Crear nuevas ofertas de alojamiento:

- ✓ **Propósito:** Crear ofertas de alojamiento.
- ✓ **Entrada:** Datos del inmueble introducidos en el formulario.
- ✓ **Proceso:** Los datos de la nueva oferta de alojamiento se guardan en la base de datos.
- ✓ **Salida:** Mensaje de confirmación o de error, dependiendo del resultado de la operación. Correo electrónico tanto al propietario del inmueble como al administrador.

-Visualizar ofertas de alojamiento que han sido publicadas a través del espacio personal del usuario:

- ✓ **Propósito:** Visualizar las ofertas publicadas.
- ✓ **Entrada:** Clic en el botón de visualizar en la parte de acciones. El botón se hace visible sólo cuando la oferta se publica, en caso contrario no se ve en la lista de acciones.
- ✓ **Proceso:** Identificar el inmueble a visualizar, comprobar que su estado es “Publicada”.
- ✓ **Salida:** Una vez que la comprobación del punto anterior se lleva a cabo con éxito, visualizar el inmueble.

-Modificar/eliminar ofertas de alojamiento que todavía no han sido publicadas:

- ✓ **Propósito:** Modificar los datos de los inmuebles/eliminar los inmuebles.
- ✓ **Entrada:** Nuevos datos del inmueble/identificador del inmueble a eliminar.
- ✓ **Proceso:** Los datos del inmueble se actualizan en la base de datos/los datos se eliminan de la base de datos.
- ✓ **Salida:** Mensaje de confirmación o de error, dependiendo del resultado de la operación.

-Renovar ofertas de alojamiento que están a punto de ser caducadas:

- ✓ **Propósito:** Modificar la fecha de caducidad de las ofertas que caducarán próximamente.
- ✓ **Entrada:** Clic en el botón de “Renovar” en la parte de acciones. El botón se hace visible sólo cuando faltan quince días de la fecha de caducidad del inmueble.
- ✓ **Proceso:** Actualizar la fecha de caducidad del alojamiento en la base de datos (se le añaden noventa días a la fecha de caducidad).
- ✓ **Salida:** Mensaje de confirmación o de error, dependiendo del resultado de la operación.

-Cerrar sesión:

- ✓ **Propósito:** Salir de la aplicación web.
- ✓ **Entrada:** Clic en el botón “Salir”/identificador de sesión.
- ✓ **Proceso:** Se eliminará el identificador de sesión. Se redirigirá a la página de visualización de las ofertas de alojamiento publicadas.
- ✓ **Salida:** Salir del espacio personal del usuario y ver alojamientos en el porta web.

Usuario administrador

El usuario administrador tiene los mismos privilegios del usuario no registrado y, además, los siguientes privilegios:

-Iniciar sesión:

- ✓ **Propósito:** Identificarse en la aplicación web.
- ✓ **Entrada:** NIF, contraseña.
- ✓ **Proceso:** Verificar los credenciales introducidos.
- ✓ **Salida:** Si la verificación ha ido correctamente, ir al espacio personal del administrador.

-Visualizar ofertas de alojamiento que no están publicadas en forma de borrador:

- ✓ **Propósito:** Visualizar borradores de ofertas de alojamiento que están por moderar o rechazadas.
- ✓ **Entrada:** Clic en el botón “Visualizar” en la parte de acciones/identificador de inmueble a visualizar.
- ✓ **Proceso:** Identificar el registro del inmueble a visualizar.
- ✓ **Salida:** Una vez realizado el paso anterior correctamente, visualizar el inmueble en forma de borrador.

-Filtrar/buscar inmuebles:

- ✓ **Propósito:** Buscar inmuebles con la ayuda de los buscadores que dispone el administrador en su intranet junto al listado de ofertas de alojamiento registradas.
- ✓ **Entrada:** Título de la oferta, tipo de alojamiento, localidad y estado.
- ✓ **Proceso:** Se cargan los alojamientos que cumplen con las condiciones de la búsqueda.
- ✓ **Salida:** Se visualiza el resultado de la búsqueda.

-Publicar ofertas de alojamiento:

- ✓ **Propósito:** Publicar ofertas.
- ✓ **Entrada:** Clic en el botón “Publicar” en la parte de acciones/identificador de inmueble a publicar.
- ✓ **Proceso:** Identificar el registro del inmueble a publicar y actualizar el estado de la oferta a “Publicada” en la base de datos.

- ✓ **Salida:** La oferta de alojamiento que ha sido publicada se visualiza en el listado general de inmuebles en el portal web. Correo electrónico al propietario del inmueble.

-Rechazar ofertas de alojamiento:

- ✓ **Propósito:** Publicar ofertas de alojamiento que no cumplan con las condiciones.
- ✓ **Entrada:** Clic en el botón “Rechazar” en la parte de acciones/identificador de inmueble a rechazar. Antes de rechazar la oferta habrá que explicar brevemente el motivo del rechazo.
- ✓ **Proceso:** Identificar el registro del inmueble a rechazar y actualizar el estado de la oferta a “Rechazada” en la base de datos.
- ✓ **Salida:** Mensaje de confirmación. Correo electrónico al propietario del inmueble.

-Eliminar ofertas de alojamiento:

- ✓ **Propósito:** Eliminar ofertas de alojamiento permanentemente.
- ✓ **Entrada:** Clic en el botón “Eliminar” en la parte de acciones/identificador de inmueble a eliminar.
- ✓ **Proceso:** Identificar el registro del inmueble a eliminar y eliminar el registro de la base de datos.
- ✓ **Salida:** Mensaje de confirmación o error, dependiendo del resultado. Correo electrónico al propietario del inmueble.

-Cerrar sesión:

- ✓ **Propósito:** Salir de la aplicación web.
- ✓ **Entrada:** Clic en el botón “Salir”/identificador de sesión.
- ✓ **Proceso:** Se eliminará el identificador de sesión. Se redirigirá a la página de login.
- ✓ **Salida:** Salir del espacio personal del administrador.

2.3.3 Requisitos de eficiencia

Ya que el sistema dependerá de la velocidad de acceso a la base de datos alojada en un servidor externo, este deberá proveer a la aplicación los servicios requeridos permitiendo un acceso fluido a todos los usuarios de la aplicación.

2.3.4 Restricciones de diseño

-Estándares de diseño

No existen restricciones en cuanto a estándares, aunque se ha procurado cumplir los propuestos por la W3C, por ser los más internacionales. Además, tiene acceso seguro, así sólo pueden acceder a la parte privada los usuarios con permiso mediante un proceso de autenticación.

-Limitaciones hardware

Al ser una aplicación web, no necesita un hardware específico. Por tanto, no existen limitaciones hardware.

2.4 Atributos

2.4.1 Seguridad

Esta aplicación cumple con la Ley Orgánica de Protección de Datos. Los datos confidenciales de la cuenta de un cliente registrado solo serán accesibles para el propietario de la cuenta y los administradores. Un usuario solo podrá acceder a sus datos tras validar su cuenta después del registro y realizar el proceso de autenticación.

2.4.2 Mantenimiento

El mantenimiento de la aplicación se llevará a cabo por el programador web del departamento en el que se ha realizado el proyecto, entendiéndose como tal las tareas de mantener actualizado y al día los servicios que ofrece. Así como cualquier cambio y/o mejora que se deseara introducir y requiriese de la modificación de la base de datos.

2.4.3 Portabilidad

Como se ha comentado con anterioridad, esta aplicación web es compatible con cualquier sistema que siga los estándares convencionales de navegación web. Esto significa que la aplicación podrá ser soportada por cualquier plataforma y sistema operativo.

2.5 Otros requerimientos

Se trabaja sobre una base de datos SQL Server donde se almacenará toda la información necesaria para el funcionamiento de la aplicación web. Mediante ASP .Net y su

API de acceso a base de datos SQL Server, el servidor web hace las consultas sobre la base de datos. Para poder introducir datos es necesario ser un usuario registrado y tener los permisos necesarios para ello. A continuación se detallan las especificaciones del servidor y del ordenador en los que se ha trabajado durante la realización de la aplicación.

Especificaciones del servidor

Hardware

- **Procesador:** Dual-Core AMD Opteron(tm) Processor 2216 HE 2.40 Ghz (4 procesadores).
- **Memoria instalada (RAM):** 4,00 GB.
- **Tipo Sistema:** Sistema operativo de 64 bits.
- **Disco duro:** Dispositivo SCSI virtualizado con VMWare, 80 GB.

Software:

- **Sistema Operativo:** Windows Server 2008 R2 Enterprise Service Pack 1.
- **Servidor Aplicaciones/Web:** Internet Information Service 7.
- **Sistema Gestor de Bases de Datos:** SQL Server 2008 R2.
- **Plataforma web:** .NET Framework 4.

Especificaciones del PC

Hardware:

- **Procesador:** Intel(R) Pentium(R) Dual CPU E2220 @2,40GHz 2.40 GHz.
- **Memoria instalada (RAM):** 4,00 GB (3,25 GB utilizable).
- **Tipo sistema:** Sistema operativo de 32 bits.
- **Disco duro:** 300 GB.

Software:

- **Sistema Operativo:** Windows 7 Profesional Service Pack 1.
- **Entorno de Desarrollo de bases de datos:** SQL Server Management Studio 2008 R2.
- **Entorno de Desarrollo web:** Visual Studio 2010.
- **Tecnologías de desarrollo en Capa de Presentación:** [ASP.NET](#) 4 Web Forms, [ASP.NET](#) Ajax, JQuery, HTML, CSS, Javascript.
- **Tecnologías de desarrollo en Capa de Negocio:** C#.
- **Tecnologías de desarrollo en Capa de Acceso a Datos:**

- *SqlDataSource*: Esto es para conectar directamente con las tablas SQL Server, lo usamos a veces para obtener contenido dinámico para los desplegables.

- *ADO.NET*: Esto es para conectar con las bases de datos a través de objetos que referencian a nuestras clases (como Grado.cs). Dentro de [ADO.NET](#) puedes distinguir entre dos librerías:

- *System.SqlClient*: Para conectar con los procedimientos almacenados de SQL Server.

- **Tecnologías de desarrollo en Capa de Datos: SQL**

3. ASP y .NET

En este capítulo se realizará un breve estudio del framework .NET y de la tecnología ASP. Con este estudio se pretende obtener una perspectiva del funcionamiento global de .NET y ASP, que permitirá abordar el desarrollo del servicio web objetivo.

3.1 ASP

ASP es una tecnología que lanzó Microsoft del tipo lado del servidor para la generación de páginas web dinámicamente. A esta tecnología se le conoce actualmente como ASP clásico, y es el precursor del framework .NET.

Al ser una tecnología del tipo lado del servidor, ASP se ejecuta en el servidor web, y una vez finalizado, se envía el resultado al cliente a través de Internet en forma de página web. El documento que recibe el cliente es una página en HTML, lo que hace posible su compatibilidad con todos los navegadores y sistemas operativos. Por otra parte, en el lado del servidor, las páginas pueden realizar accesos a bases de datos, conexiones de red, etc. Esto se consigue mediante el uso de scripts, usando Visual Basic Script o JScript (Javascript de Microsoft).

A la hora de usar ASP disponemos de seis objetos nativos, que a diferencia de otros modelos de objetos, no forman una jerarquía, sino que se relacionan entre sí de forma lógica.

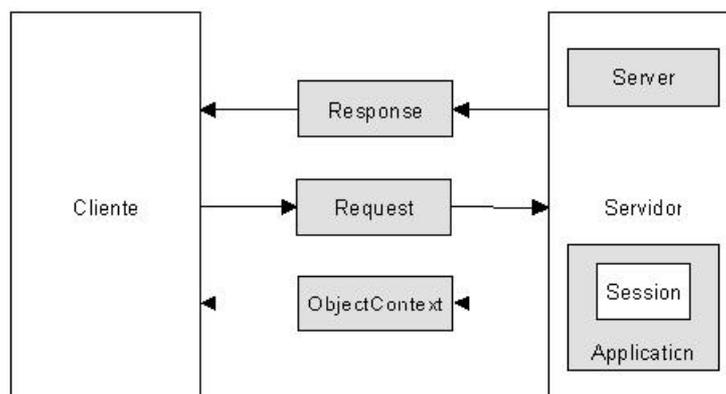


Figura 4: Estructura ASP

A continuación se procede a mostrar las características más importantes de estos objetos:

- **Application:** Su principal función es compartir la información de la aplicación entre todos los usuarios conectados a ella. Únicamente existe un objeto *Application* común a todos los usuarios.

- **ObjectContext:** Se utiliza para la gestión de las transacciones.
- **Request:** Este objeto se encarga de facilitarnos toda la información que un usuario nos manda a través de la cabecera HTTP. Esta información puede llegar en forma de petición GET o POST, en forma de cookies, etc.
- **Response:** Es el encargado de enviar toda la información del servidor al cliente, como por ejemplo el envío de información directamente al navegador del cliente o el manejo de los valores de las cookies.
- **Server:** Su función es proporcionar métodos relacionados con el servidor dónde se ejecuta la aplicación. Su principal uso es para crear una instancia de un componente.
- **Session:** Al acceder a la aplicación, a cada usuario se le asigna un objeto session, siendo esta información diferente para cada usuario. El tiempo de vida de este objeto es de veinte minutos, aunque se puede modificar, y se crea la primera vez que un usuario solicita una página.

La tecnología ASP, aparte del gran avance que supuso a la hora de desarrollar aplicaciones web, también contenía un gran número de inconvenientes que se han subsanado con la aparición de .NET. Entre ellos se puede destacar la dificultad para mantener y actualizar el código escrito debido a que está mezclado con el HTML de la página, complicando la tarea del programador a la hora de introducir novedades, ya que el fragmento de código ha de situarse en el punto exacto donde se quiere que su salida aparezca. Otro de los inconvenientes sería la necesidad de escribir código para mantener el estado de la página ASP, cuando por ejemplo, el usuario quiera volver a ella. Esto provoca la pérdida de información si hubiese algún error al, por ejemplo, rellenar un formulario.

Pero los inconvenientes de ASP no quedan aquí, también es conocido la ineficacia de esta tecnología a la hora de implantar sistemas de cierta envergadura. Debido a sus características se presentan graves problemas de eficiencia cuando, por ejemplo, llegan más peticiones al servidor de las que éste puede atender, aparte de los problemas de depuración que se pueden presentar. Aunque en realidad, muchos de estos problemas provienen de los lenguajes utilizados en los fragmentos de código en las páginas ASP. En la actualidad, debido a la creciente interoperabilidad de múltiples componentes y la manipulación de datos empleando numerosas fuentes de información se hace más complicado las aplicaciones de esta naturaleza.

Para solucionar este problema aparecieron los servicios web, cuya principal función es atender a diferentes tipos de usuarios y además se introdujo la posibilidad de que los servicios pueden residir en diferentes servidores Web. Esta tecnología fue avanzando hasta la llegada de la tecnología .NET de Microsoft, que se fundamenta en los servicios web para la construcción de aplicaciones basadas en tecnologías de Internet.

3.2 .NET

Microsoft .NET es un framework cuya principal característica es la transparencia de redes, independiente del hardware que se utilice, y que permite un rápido diseño de aplicaciones.



El entorno .NET se puede clasificar en las siguientes partes:

- **.NET Framework:** Es el entorno de trabajo de la plataforma .NET y que la engloba completamente. Toda la plataforma .NET forma parte de .NET Framework.
- **Lenguajes .NET:** Soporta más de 20 lenguajes, haciendo posible desarrollar cualquier tipo de aplicación con cualquiera de ellos. Los más usados son C#, Visual Basic .NET, Delphi, etc.
- **Common Runtime Language (CRL):** Es el entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes.
- **Microsoft Intermedial language (MSIL):** Se encarga de transformar código nativo intermedio de alto nivel independientemente de la plataforma que lo ejecuta a código de máquina del dispositivo que lo ejecuta.
- **Common Language Specification (CLS):** Su función consiste en determinar las reglas para crear el código MSIL .
- **ASP.NET:** Es la evolución del ASP clásico. Su principal característica es su facilidad para la creación de páginas Web dinámicas. Además se encuentra integrado en el entorno .NET.
- **Biblioteca de clases .NET:** Maneja la mayoría de las operaciones básicas que se encuentran involucradas en el desarrollo de aplicaciones.

3.2.1 MSIL, CRL y el código controlado

A la hora de escribir un programa se puede hacer en diferentes lenguajes de programación, lenguajes que, aunque no se puedan leer como tal, podemos entenderlos con ayuda de los conocimientos adquiridos, mediante un editor de texto. Pero cuando se quiere ejecutar un programa, un ordenador no es capaz de entender este lenguaje, por lo que se tiene que traducir a lenguaje máquina, es decir, el lenguaje que las máquinas son capaces de entender. Esto se hace a través de la compilación, cuya función es traducir el programa del lenguaje que nosotros hemos escrito al lenguaje que la máquina pueda entender, dando como resultado un ejecutable, que si bien la máquina ya puede entender, nosotros no podemos interpretar.

Este lenguaje máquina es diferente dependiendo del sistema operativo, arquitectura del procesador, etc, que se utilice, solo funcionara en la plataforma para la que fue diseñado.

A la hora de desarrollar .NET, se hizo especial hincapié en este tema, ya que el objetivo a conseguir era dejar atrás los problemas de compatibilidad del software dependiendo de la plataforma que se utilizase. Para ello se creó un lenguaje intermedio llamado MSIL. Cuando se programa en .NET, independientemente del lenguaje que se esté utilizando, siempre se compila hacia MSIL (Microsoft Intermediate Language). Una vez que el programa ha sido compilado será interpretado por un intérprete, el CRL. Con esto se consigue, obteniendo el intérprete apropiado, que cualquier programa escrito en .NET funcione en cualquier plataforma.

Cuando se compila a MSIL, el procesador no puede interpretar el programa ya que no está escrito en lenguaje máquina, para ello se utiliza el CRL, cuya función es hacer de intermediario entre el lenguaje de MSIL y el lenguaje máquina.

El CRL es muy parecido al *ByteCode* de Java, aunque CRL es mejor, además que una de las características de CRL, al contrario que *ByteCode*, únicamente compila la parte necesario del código en cada momento durante la ejecución.

Así podemos ejecutar nuestro programa sobre cualquier máquina, siempre y cuando exista una versión del .Net Framework y del CRL apropiada. Al código que se ejecuta bajo la batuta del CRL se le conoce como código manejado.

La aparición de JAVA en el mercado supuso un gran adelanto, debido a la aparición del concepto de máquina virtual, en el que un lenguaje era compilado a un lenguaje intermedio, que con ayuda de una máquina virtual podía ejecutarse en toda clase de máquinas. Microsoft adopta esta idea en .NET creando CRL, con la diferencia de que no se limita a un único lenguaje. La estrategia de Microsoft en este ámbito es intentar acoger al mayor número de desarrolladores, independientemente del lenguaje que utilicen. De esta manera se asegura que su tecnología .NET pueda ser usada por cualquier desarrollador.

CRL se caracteriza por ser independiente del sistema operativo por tanto se aprecia como Microsoft ve nuevos frentes donde expandirse o protegerse según se mire. El mercado mundial de sistemas operativos orientado al sector de microinformática está dominado por la serie Windows. En años recientes un nuevo competidor GNU ha llegado para quedarse, Linux, una reciente versión de UNIX está comiendo mercado debido a sus ventajas heredadas de UNIX y a las suyas propias. Se puede apreciar como Microsoft podría desembarcar en Linux empleando .NET o bien podría ser un signo de debilidad del sistema Windows ante la llegada de Linux. CRL puede en un futuro implantarse en otros sistemas operativos distintos de la serie Windows como base para el desarrollo de aplicaciones de escritorio. Como se aprecia, esta característica es nueva con respecto a ASP que no da soporte al desarrollo de aplicaciones de escritorio.

3.2.2 Basic Class Library

La librería de clase base es una serie de librerías contenidas en el framework .Net diseñadas para ayudar a los desarrolladores a construir sus aplicaciones. Está formada por cientos de tipos de datos cuya principal función es acceder a los servicios que ofrece el CRL así como el acceso a las funcionalidades más habituales en lo que a escribir programas se refiere. También el desarrollador puede crear clases nuevas que mediante herencia extiendan su funcionalidad.

La ventaja de esta librería es que está escrita en MSIL y por lo tanto se puede usar desde cualquier lenguaje que genere MSIL, es decir, todos los lenguajes que contiene .Net. Una de las particularidades de C# es que carece de librería de clases propia, basándose solo en la BCL.

Debido al gran tamaño de la BCL, se han organizado las clases en espacios de nombres, agrupando las clases con funcionalidades similares. Un ejemplo son:

- **System:** Contiene clases fundamentales y clases base que definen los tipos de datos de valor y de referencia, los eventos y controladores de eventos, los interfaces, los atributos y las excepciones de procesamiento más frecuentes. Proporciona servicios que admiten



la conversión de tipos de datos, la manipulación de los parámetros de métodos, matemáticas, la invocación de programas remotos y locales, la administración del entorno de las aplicaciones y la supervisión de aplicaciones administradas y no administradas.

- **System.Collections:** Contiene interfaces y clases que definen varias colecciones de objetos, como listas, colas, matrices, tablas hash y diccionarios. Incluye el espacio de nombres Specialized para colecciones con establecimiento inflexible de tipos.
- **System.Data:** Manipulación de bases de datos. Forman la denominada arquitectura ADO.NET.
- **System.IO:** Contiene tipos que permiten la lectura y escritura sincrónicas y asincrónicas en archivos y secuencias de datos.
- **System.NET:** Proporciona una interfaz de programación sencilla para un gran número de protocolos utilizados actualmente en la red. Las clases WebRequest y WebResponse constituyen la base de los “protocolos conectables”, una implementación de servicios de red que permite desarrollar aplicaciones que utilizan recursos de Internet sin tener que preocuparse de las particularidades del protocolo utilizado.
- **System.Reflection:** Contiene clases e interfaces que proporcionan una vista administrada de los tipos, métodos y campos cargados, permitiendo la creación y la invocación dinámicas de los tipos.
- **System.Runtime.Remoting:** Proporciona clases e interfaces que permiten a los programadores crear y configurar aplicaciones distribuidas de correspondencia estricta o imprecisa.
- **System.Security:** Proporciona la estructura subyacente del sistema de seguridad del motor de tiempo de ejecución, incluidas en las clases base para los permisos.
- **System.Threading:** Proporciona clases e interfaces que permiten la programación multiproceso. También proporciona clases para la programación de los subprocesos, las notificaciones de espera y la resolución de bloqueo.
- **System.Web.UI:** Proporciona clases e interfaces que permiten crear controles y páginas que constituyen la interfaz de usuario para las aplicaciones Web. Proporciona asimismo clases que incluyen la función de enlace de datos a controles de servidor de formularios Web Forms, permiten guardar el estado de vista de un control o una página determinados y analizar la funcionalidad de los controles programables y Literal.
- **System.Windows.Forms:** Creación de interfaces de usuario y aplicaciones basadas en Windows para aplicaciones estándar.
- **System.XML:** Acceso a datos en formato XML.

3.2.3 Assemblies



Cuando se genera un proyecto .NET no se genera un ejecutable, sino que se genera un *assemblies*. Un *assembly* es la unidad ejecutable de cualquier programa .NET, pero no es solo eso, también incluye el manifiesto. Un manifiesto es un listado de las librerías y los controles necesarios para que la aplicación funcione. Incluye, entre otras cosas, el número de versión necesario para que la aplicación funcione.

3.2.4. ViewState

El ViewState nos facilita el almacenar la información de estado en las páginas Web ASP.NET entre devoluciones de datos y Post Back. Esta información se serializa en XML para luego codificarla en Base64. Su funcionamiento se limita al procesamiento de la página, a la codificación del estado actual de la página guardando en la página la cadena que se obtiene de forma oculta.

El ViewState es una de las diferencias entre ASP y .NET, ya que permite mantener el estado de los controles de una misma página entre una ida y venida al servidor. Además tiene muchas otras funciones como poder conocer si es la primera vez que se ejecuta la página, o guardar las variables que se quieren accediendo desde el código.

El problema de ViewState es que al serializar la información en XML y luego codificarla en Base64, puede generar grandes cantidades de información que podría llegar a afectar al rendimiento de la página. Para solucionar este problema se puede deshabilitar el ViewState para los controles de datos como GridView, DataList, etc. Para ello se utilizaría la expresión: `EnableViewState="false"`.

3.3 Diferencias entre ASP y .NET

Con la llegada de .NET, Microsoft quiso dar otro enfoque para el desarrollo de software. No se limitó a actualizar ASP, sino que desarrolló una tecnología totalmente nueva para el desarrollo de aplicaciones Web. Las principales prioridades de Microsoft a la hora de desarrollar .NET fueron usar XML como vehículo transmisor de información, incrementar la presencia de modos de tratamiento de la información mediante la librería de datos y utilizar una nueva filosofía de construcción de formularios Web. El lanzamiento de la tecnología .Net por parte de Microsoft supuso un duro golpe para J2EE ya que lo superaba en varios aspectos. Uno de ellos es el acceso más sencillo y la facilidad de poder emplear una gran variedad de lenguajes de codificación, pero sin perder potencia de diseño ya que los principales lenguajes de programación son orientados a objetos.



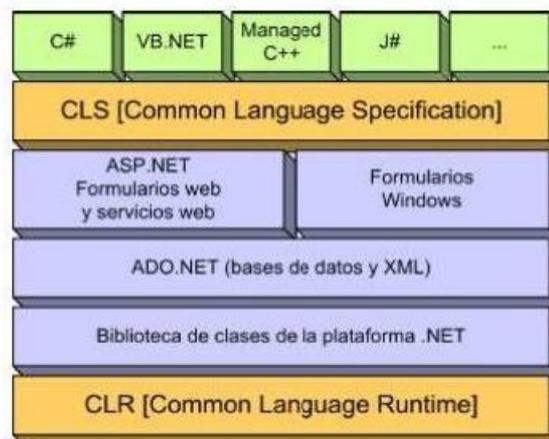
Figura 5: Configuración de una aplicación Web con HTML estático

Antes de la aparición de la Web dinámica, la mayoría de páginas Web se basaban en utilizar documentos HTML estáticos que ofrecían siempre la misma información (Figura 4), que aunque puede ser la solución más adecuada cuando nuestra página web siempre ofrezca la misma información, la evolución de Internet ha llevado a los desarrolladores a crear aplicaciones Web que generan dinámicamente el contenido que finalmente se ofrece a los usuarios. De esta manera podemos seleccionar, filtrar, ordenar y presentar la información de la forma más adecuada en función de las necesidades de cada momento. Aunque esto se podría conseguir con páginas de HTML estático se necesitaría un gran espacio disponible en el disco duro, por lo que la incursión de las aplicaciones Web ha sido fundamental en la evolución de Internet tal y como lo conocemos hoy.

Para ello, la creación de aplicaciones Web requiere la existencia de software ejecutándose en el servidor, cuya función será la generación de los ficheros HTML que le llegan al usuario. Para el usuario no ha supuesto un gran cambio, ya que él sigue realizando la conexión con el servidor a través del protocolo HTTP y recibiendo páginas HTML, que si bien estas han sido creadas dinámicamente, el usuario final no lo puede distinguir. La principal diferencia consiste en que, ahora, el servidor HTTP delega en otros módulos la generación dinámica de las páginas HTML que se envían al cliente (Figura 5).



Figura 6: Configuración de una aplicación Web con HTML dinámico



Para la construcción de interfaces de usuario en .NET se dispone de dos herramientas, los formularios Web y los formularios ASP.NET. El problema de estas dos herramientas es que

Figura 7: Familia de tecnologías de la plataforma .NET

no son intercambiables, es decir, no existe una forma estándar de crear una interfaz de usuario que funcione tanto para aplicaciones Windows como para aplicaciones Web.

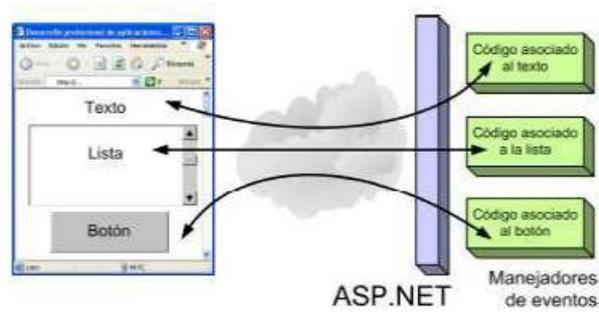


Figura 8: Funcionamiento de los eventos en la plataforma .NET

Una de las diferencias de ASP y ASP.NET es que éste último en vez de aceptar datos de entrada y generar su salida en HTML, implementa su funcionalidad en fragmentos de código que se ejecutan como respuesta a eventos asociados a los controles de la interfaz con los que puede interactuar el usuario. De esta manera se consigue un mayor nivel de abstracción, necesita menos código y permite crear aplicaciones más modulares, legibles y fáciles de mantener.

4. Análisis

En esta fase, la de análisis del proyecto, se va a analizar toda la aplicación a desarrollar y a describir su estructura y funcionalidad mediante diagramas que ayudarán a comprender cómo funciona la aplicación.

Con estos diagramas voy a describir, de forma gráfica, cada una de las funcionalidades que lleva a cabo la aplicación (casos de uso) y cada uno de los objetos de los que se compone la misma (diagrama de clases, diagrama de secuencia), utilizando en ambos casos la notación UML (Unificado de Modelado). En resumen, me voy a centrar en tres tipos de diagramas UML:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de secuencia.

4.1 Casos de uso

-Relación: Conexión entre los elementos del modelo. Un ejemplo es la especialización y/o la generalización.

-Actor: Toda entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos así como a entidades abstractas como el tiempo.

Mediante el diagrama de casos de uso se puede describir la funcionalidad de la fase de análisis previa a la construcción de la aplicación. Se puede obtener una representación gráfica de las interacciones entre actores y casos de uso. Ésta última describe el comportamiento del sistema desde el punto de vista de los usuarios.

En la siguiente figura se puede observar el diagrama que describe los distintos usuarios de la aplicación y la relación que existe entre ellos.

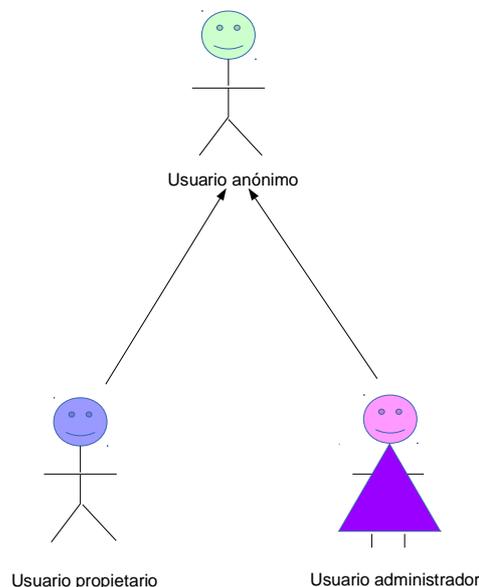


Figura 9: Actores del diagrama de casos de uso

Los usuarios podrán realizar diferentes tareas dependiendo del tipo al que pertenezcan y los privilegios que tengan. En los siguientes apartados se pueden ver los diagramas de casos de uso clasificados por actor.

4.1.1 Caso de uso: Actor anónimo

En la siguiente figura se pueden ver los casos de uso a los que puede acceder el actor anónimo. Puesto que éste no es un usuario registrado, las acciones que podrá realizar serán, en consecuencia, muy limitadas.

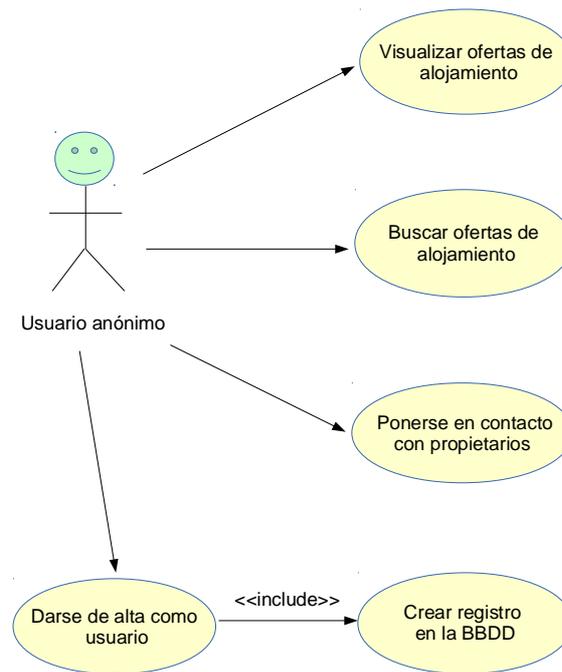


Figura 10: Diagrama casos de uso - Actor anónimo

4.1.2 Caso de uso: Actor propietario

Como se puede observar en la siguiente figura, este actor dispone de muchos casos de uso, pues se trata de un tipo de actor con privilegios (incluidos los del actor anónimo) en comparación con el anteriormente expuesto.

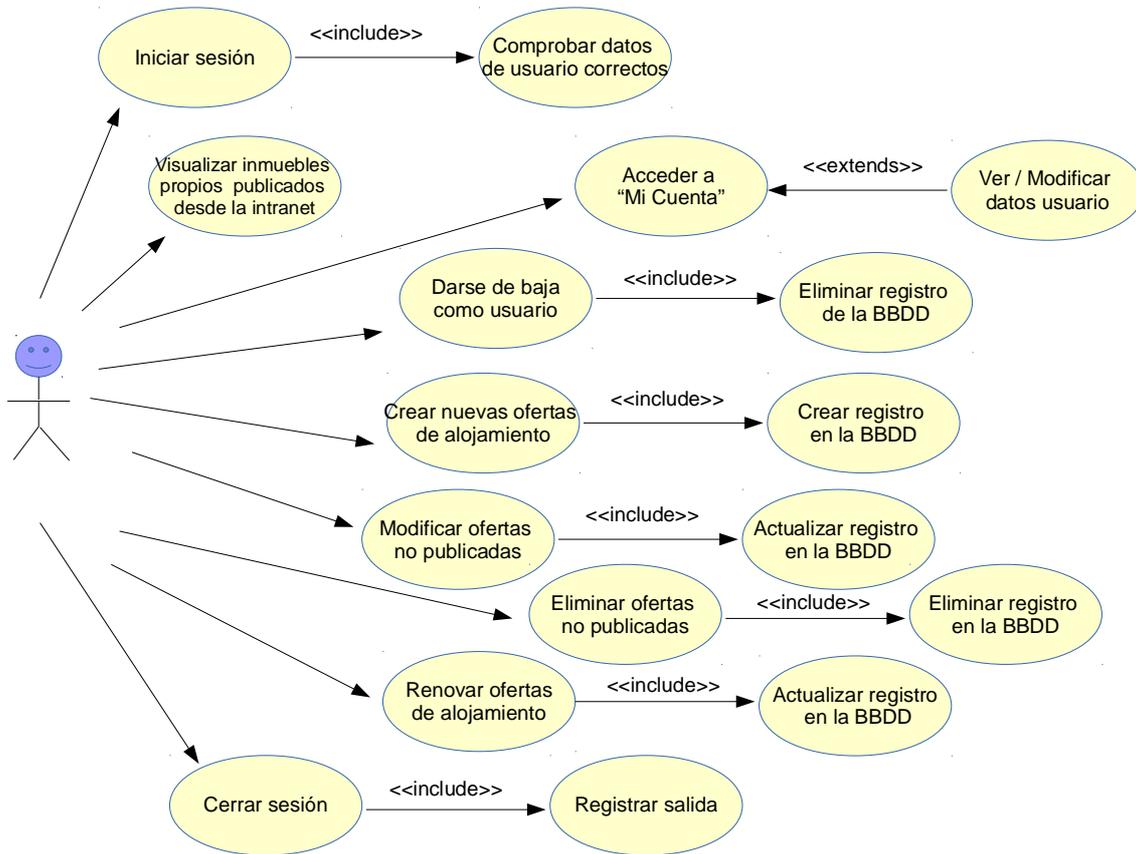


Figura 11: Diagrama casos de uso - Actor propietario

4.1.3 Caso de uso: Actor administrador

En la figura que se observa a continuación se puede observar los casos de uso del actor administrador. Posee las funcionalidades del actor anónimo, algunas del actor propietario y otras propias de este actor.

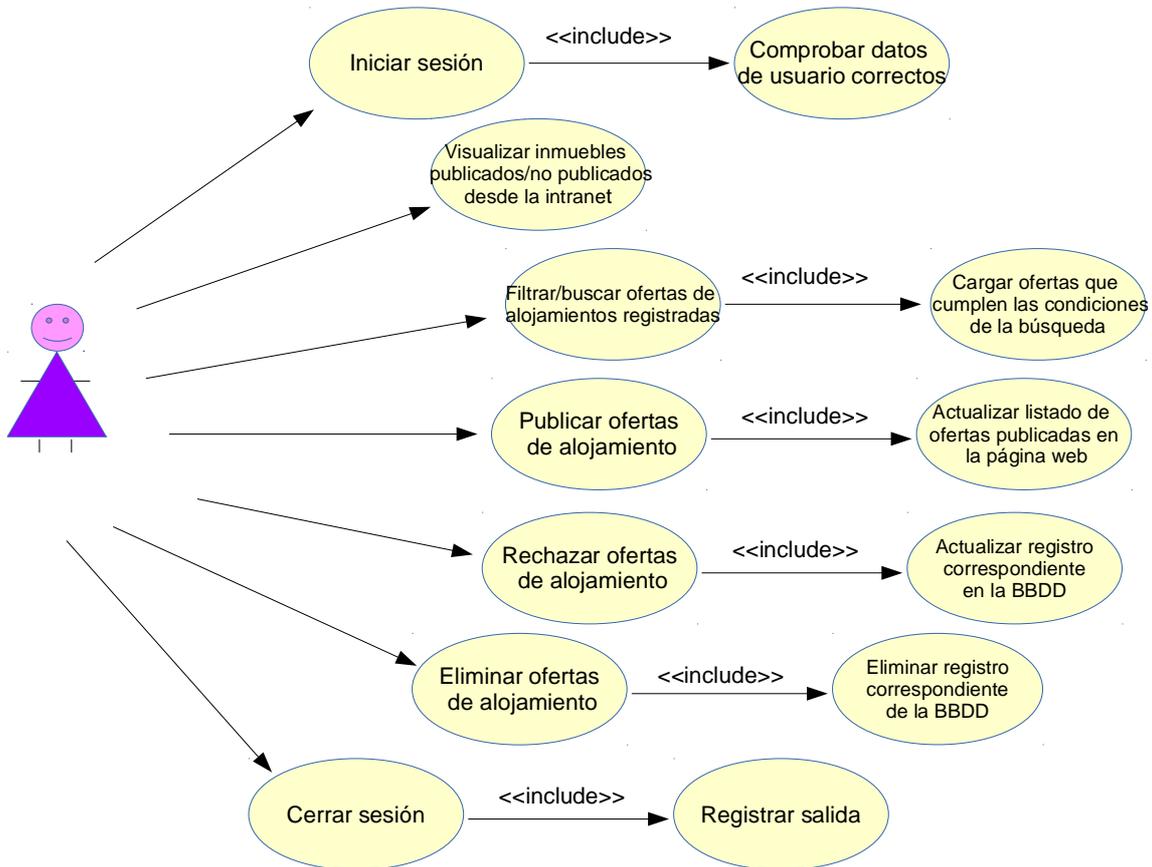


Figura 12: Diagrama casos de uso - Actor administrador

4.2 Diagrama de clases

El diagrama de clases que se observa a continuación describe la estructura de la aplicación mostrando sus clases, atributos y las relaciones entre ellos, es decir, los componentes que se encargarán del funcionamiento de la aplicación y la relación entre uno y otro.

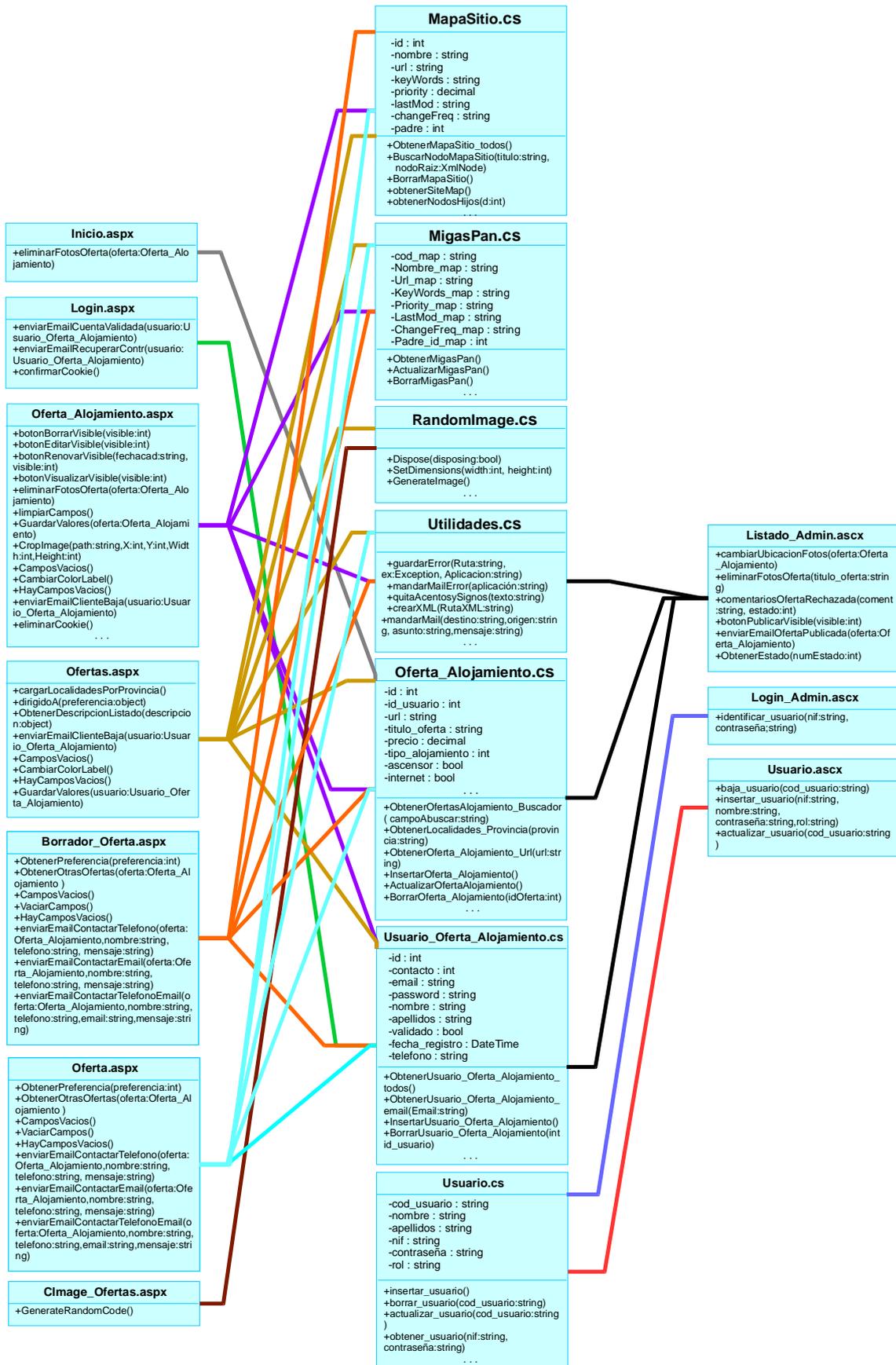


Figura 13: Diagrama de clases

4.3 Diagrama de secuencias

Es un tipo de diagrama de interacción en UML que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones pero no están pensados para mostrar lógicas de procedimientos complejos.

A continuación se pueden ver algunos de los diagramas de secuencias más usados correspondientes a esta aplicación.

4.3.1 Inicio de sesión

Este diagrama de secuencias representa el inicio de sesión de los usuarios registrados, necesario para poder acceder a la parte privada de la aplicación.

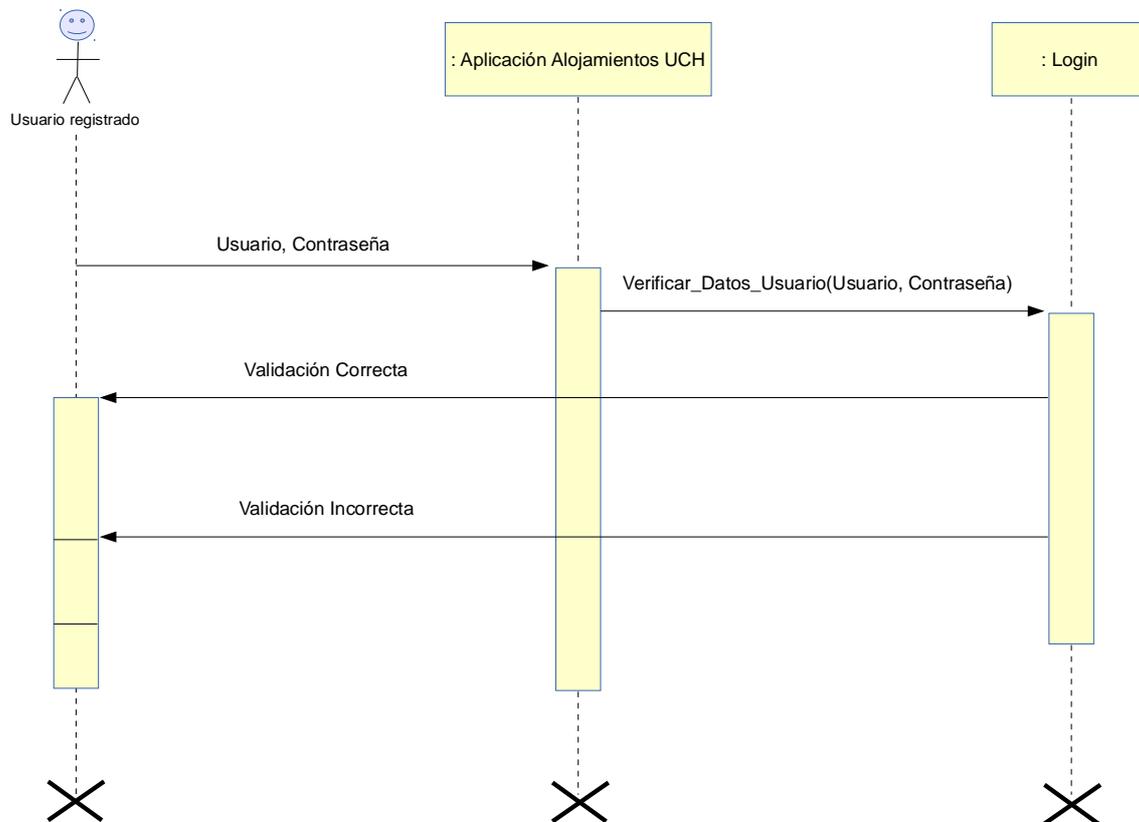


Figura 14: Diagrama de secuencias - Inicio de sesión

4.3.2 Crear oferta de alojamiento

En la siguiente figura se puede observar cómo los usuarios registrados pueden crear ofertas de alojamientos accediendo a la aplicación e introduciendo los datos de la oferta que se quiere crear.

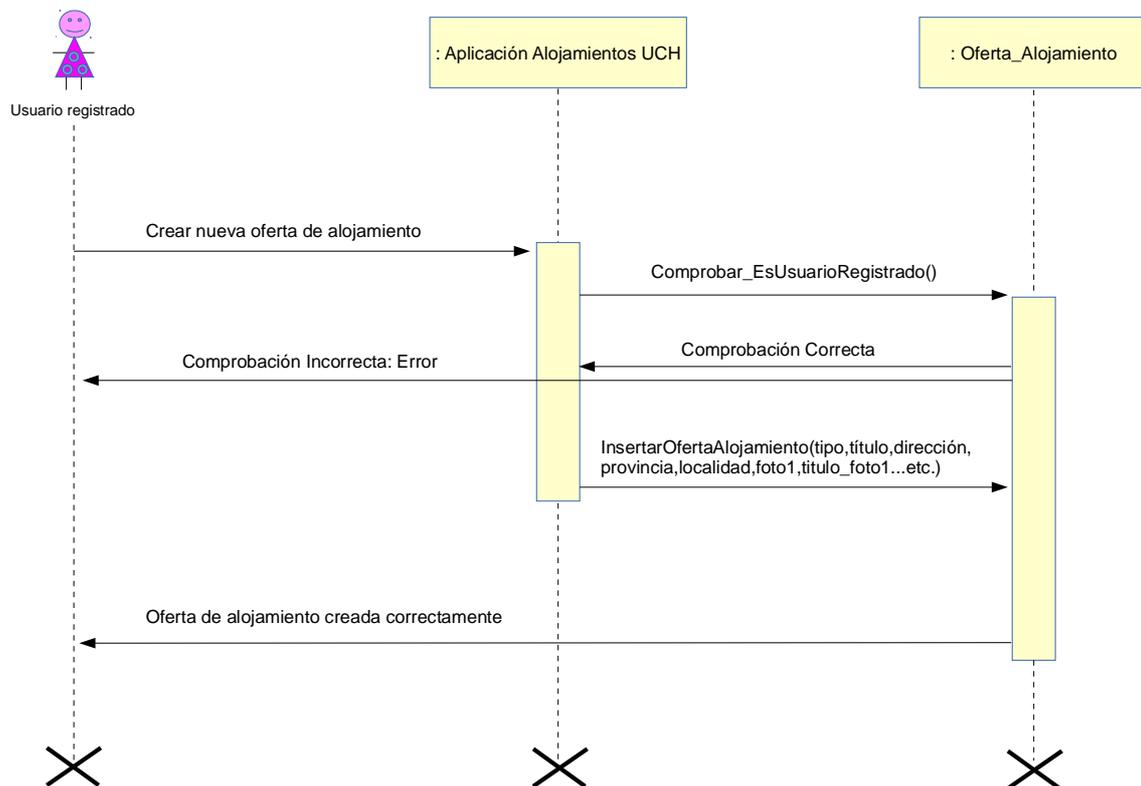


Figura 15: Diagrama de secuencias - Creación oferta de alojamiento

4.3.3 Registro de usuarios

El proceso del registro de los usuarios en la aplicación se refleja en el siguiente diagrama de secuencias.

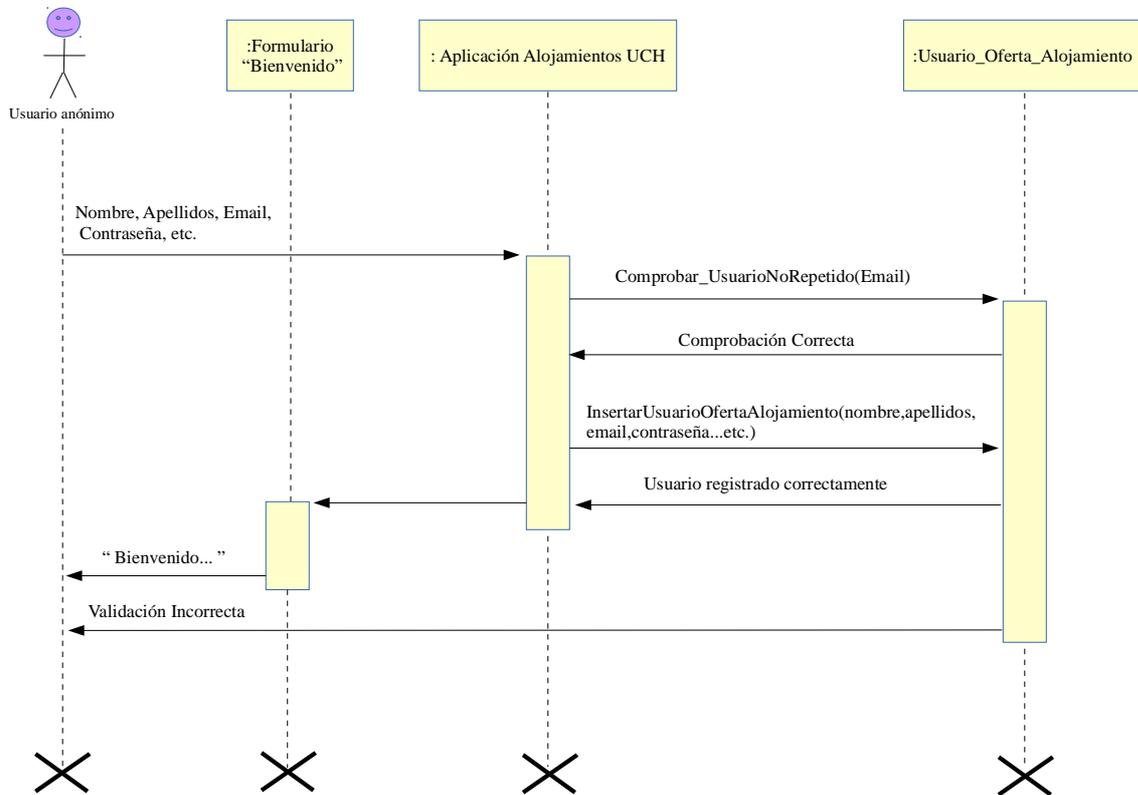


Figura 16: Diagrama de secuencias - Registro de usuarios

5. Diseño

El diseño de esta aplicación sigue la programación por capas, una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la capa de presentación, la de lógica de negocio y la de persistencia de datos.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

Cada una de las capas es totalmente independiente al resto. Esto se hace para separar las tareas a realizar por cada una de ellas y dividir la solución final del proyecto, posibilitando avanzar en el proyecto en varios frentes a la vez. Será necesaria una comunicación entre las capas con la información requerida por cada una.

La técnica de programación en tres capas es óptima de cara a conseguir la mejor calidad acoplamiento-cohesión para facilitar el desarrollo y mantenimiento de la aplicación. Ambos parámetros van encaminados a estructurar una relación ordenada entre las distintas capas y asegurarnos que se implementan con la mejor autonomía posible entre ellas. Lo ideal es plantear un sistema donde se produzca un bajo acoplamiento y una alta cohesión entre sus partes.

A continuación se puede ver la representación gráfica de la arquitectura de tres capas.

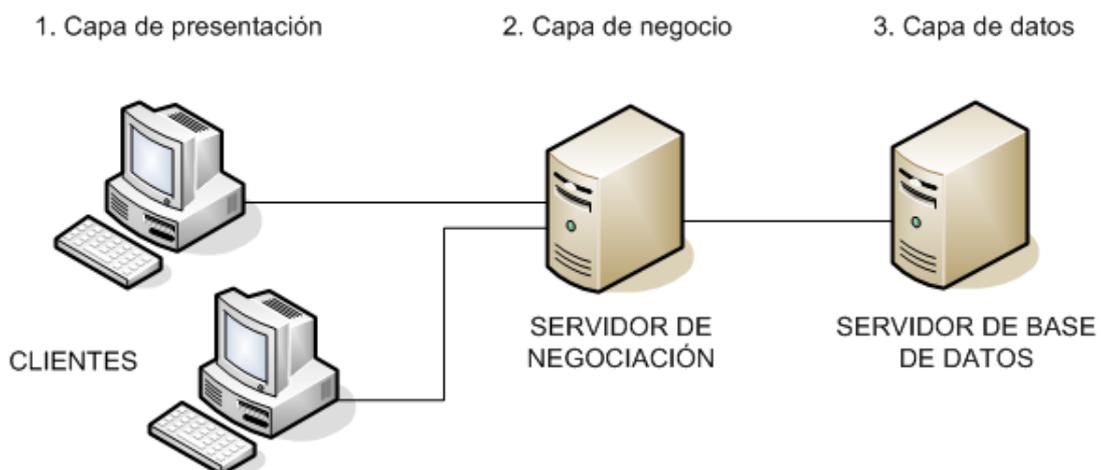


Figura 17: Arquitectura de tres capas

1. Capa de presentación: es la que ve el usuario (también denominada “capa de usuario”), presenta el sistema de usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser “amigable” (entendible y fácil de usar) para el usuario.

2. Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen 32 todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

3. Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

5.1 Capa de presentación

Es la capa encargada de interactuar con el usuario y en el caso concreto de este proyecto está formado por el conjunto de formularios y las páginas web a través de las cuales se muestra información al usuario.

Contiene todas las funcionalidades descritas en los apartados anteriores, accediendo a la información a través de la capa de lógica de negocio. Tendrá un diseño sencillo para facilitar al usuario la usabilidad de la aplicación. El diseño de la capa de presentación dispondrá de un menú superior y la zona central para mostrar la información.

Para implementar la Interfaz de Usuario se ha utilizado el lenguaje C# .Net y framework 4.0 acompañado de algún componente externo para la representación de datos de las páginas web de la aplicación.

5.1.1 Parte pública

A la parte pública de la web puede acceder cualquier tipo de usuario. Al acceder a la página principal del sitio web dedicado a las ofertas de alojamiento (www.uchceu.es/alojamiento) se puede ver el listado de todas las ofertas de alojamiento publicadas.

A continuación se puede ver la página principal de la parte pública.



Figura 18: Capa presentación - Página principal parte pública

La pantalla principal del sitio web se compone de cuatro partes:

- **Cabecera principal:** se incluye en todas las páginas del sitio web de la Universidad Cardenal Herrera. Incluye el logo de la universidad, una opción para cambiar de idioma y dos menús.



Figura 19: Cabecera principal UCH-CEU

- **Cabecera personalizada:** varía dependiendo de la página del sitio web de la universidad. En este caso es la que se muestra en la siguiente figura.



Figura 20: Cabecera página web - Alojamiento

- Cuerpo de la página:** en esta parte central de la página se haya la zona de información (Figura 21), donde se publicará el contenido de las ofertas de alojamiento disponibles ordenadas por la fecha de publicación (empezando por las más recientes). Se mostrarán siete inmuebles por página y la primera del listado será el alojamiento destacado y se diferenciará de los demás por el color de fondo. Además, en la parte superior del cuerpo de la página se puede apreciar la presencia de una serie de filtros que permiten realizar una búsqueda rápida y personalizada de inmuebles.

Futuro Alumno **Ofertas Alojamiento**

- Colegios Mayores conveniados con la CEU-UCH
 - Resto de Colegios Mayores y Residencias Universitarias

Publica tu alojamiento gratis: [Date de alta](#) | [Gestiona tus ofertas](#)

Tipo: Todos
Precio: Todos
Provincia: Todas
Localidad: Todas
Ordenar por: Fecha de publicación

COLEGIO MAYOR AUSIAS MARCH

Tipo: Colegio mayor
Ausias March, el Colegio Mayor de Valencia. BUS PRIVADO de puerta a puerta con servicios de mañana y tarde (14 viajes al día). PENSIÓN COMPLETA FINES DE SEMANA Y FESTIVOS INCLUIDOS sin coste adicional.

687,50 euros/mes

15 m²
100 Hab.

Alojamiento destacado

Situado en: VALENCIA
Dirigido a: chicos y/o chicas

[Ver información y fotos >>](#)

ENORME PISO CON TERRAZA EN BENIMACLET

Tipo: Piso compartido
HAB A:310/360€* (doble,20m2, cama 135, balcón) *1persona/2persona
HAB B:200€ (individual,6m2, cama 90 con canape) HAB C:250/300€ (doble,10m2, cama 135) HAB D:240€ (individual, cama 105) PRECIOS ...

200 euros/mes

100 m²
4 Hab.

Situado en: VALENCIA
Dirigido a: Solo chicos

[Ver información y fotos >>](#)

PISO IDEAL ESTUDIANTE SBENIMACLET

Tipo: Piso compartido
hab a: 230 € (individual, balcón, cama 90) hab b: 250€ (individual, cama 105) hab c: 230 € (individual, camam 90) alquiler de 3 habitaciones individuales en piso situado en 1ª planta edificio s...

230 euros/mes

65 m²
3 Hab.

Situado en: VALENCIA
Dirigido a: Solo chicos

[Ver información y fotos >>](#)

PISO COMPARTIDO EN BENIMACLET

Tipo: Piso compartido
Alquiler de habitaciones en piso compartido en benimaclet. Situado en la 3ª planta sin ascensor posee unas magnificas visats de todo el barrio, techos muy altos y gran luminosidad. Tiene 3 habitacion...

220 euros/mes

75 m²
3 Hab.

Situado en: VALENCIA
Dirigido a: Solo chicos

[Ver información y fotos >>](#)

PISO EN MONCADA CERCA DEL CEU

Tipo: Piso
Primer piso todo exterior, amplio salon con terraza, 3 habitaciones confortables con armarios empotrados y amueblada para estudiantes.Doble ventana climalit. 2 WC con bañera. A 5 minutos a a pie del ...

495 euros/mes

90 m²
3 Hab.

Situado en: MONCADA
Dirigido a: chicos y/o chicas

[Ver información y fotos >>](#)

HABITACION CON BANO PRIVADO PARA CHICA DEL CEU, ON PARLE FRANCAIS

Tipo: Piso compartido
buscamos una chica que estudie en el CEU para completar piso con otras dos estudiantes de veterinaria, el piso tiene 4 anos y dispone de todas las comodidades, lavadora, utensilios de cocina etc, la ...

220 euros/mes

16 m²
3 Hab.

Situado en: ALFARA DEL PATRIARCA
Dirigido a: Solo chicas

[Ver información y fotos >>](#)

FANTASTICO PISO PARA ALQUILAR

Tipo: Piso
Vivienda casi nueva totalmente amueblada y preparada para 3 estudiantes para vivir con todo tipo de comodidades. Cerca de Supermercados, Centro de Salud, Bancos, Zona de Ocio, etc.

650 euros/mes

100 m²
3 Hab.

Situado en: MONCADA
Dirigido a: chicos y/o chicas

[Ver información y fotos >>](#)

« Primera » « Anterior » 2 / 11 « Siguiente » « Última »

ADVERTENCIA: La Universidad no se hace responsable ni de las condiciones del alojamiento ni de las condiciones del contrato de alquiler. Tampoco del comportamiento de los inquilinos ni del cumplimiento de los pagos. Este servicio pretende únicamente poner en contacto a los alumnos y a los propietarios por lo que no se admitirán reclamaciones relacionadas con estos asuntos.

Compártelo:

Me gusta 4
 Twitter 6
 1
 Share 2

Figura 21: Cuerpo de la página web

43

- *Pie de la página:* igual que la cabecera principal, aparece en todas las páginas web del sitio web de la Universidad Cardenal Herrera (Figura 22).



Figura 22: Pie de la página web

En el listado de las ofertas de alojamiento para ver más detalles de una oferta es necesario hacer clic en la imagen, en el título o en el botón azul “Ver información y fotos” que aparece en la parte izquierda de la oferta correspondiente. Así pues, se mostrará una página con todos los detalles del inmueble (en la Figura 23 se puede ver un ejemplo).

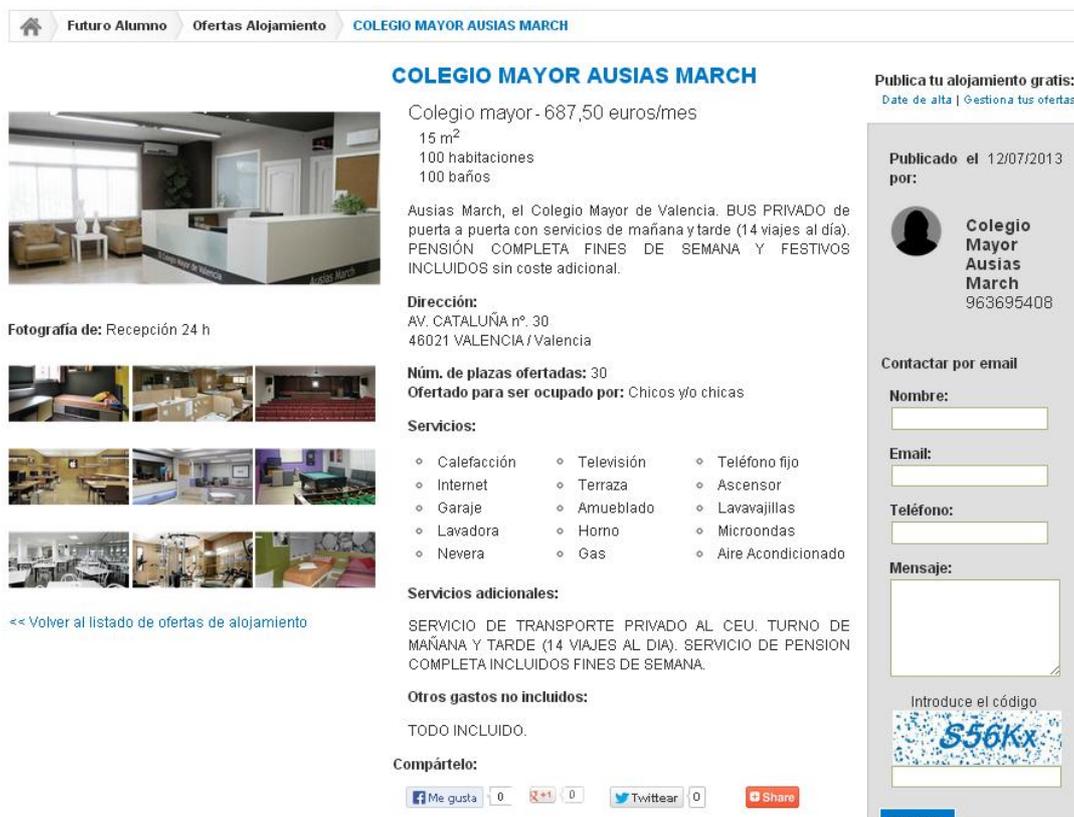


Figura 23: Página web - Oferta Alojamiento

En la parte superior derecha del cuerpo de la página aparece la opción de darse de alta como usuario a los que no están registrados y la de gestionar las ofertas a los usuarios que ya están registrados (véase Figura 24).



Figura 24: Página web principal - Date de alta | Gestiona tus ofertas

Al hacer clic en la opción “Date de alta” se visualiza la página web con el formulario que contiene los datos necesarios para rellenar para registrarse como usuario de la aplicación (Figura 25).

The image shows a registration form titled 'DATE DE ALTA'. At the top, there is a banner with the word 'Alojamiento' and an illustration of houses and a magnifying glass. Below the banner, there is a navigation bar with 'Futuro Alumno' and 'Ofertas Alojamiento'. The form itself has several input fields: 'Nombre:', 'Apellidos:', 'Teléfono:', 'Correo electrónico:', and 'Definir contraseña:'. There are also checkboxes for '¿Cómo quieres que contacten contigo?' with options for 'Teléfono' and 'Email'. Below the form fields, there is a CAPTCHA image with the text 'Introduce el código que ves en la imagen' and the word 'Guvej' in a stylized font. At the bottom of the form, there is a checkbox for 'Acepto que mis datos de contacto (nombre, apellidos y teléfono y/o e-mail) sean visibles a través de la web de la Universidad CEU Cardenal Herrera.' and a green button labeled 'Enviar Formulario'.

Figura 25: Página web - Date de alta

Si todos los campos del formulario se rellenan correctamente, el usuario se registra en la aplicación y se visualiza la siguiente página web de bienvenida durante los siguientes diez minutos y, a continuación, se redirige automáticamente a la página web principal del listado de ofertas de alojamiento (para poder mostrar un ejemplo, me he registrado en la aplicación previamente):



Figura 26: Página web de bienvenida

5.1.2 Parte privada

A la parte privada de la web solo tienen acceso los usuarios registrados y el administrador, aunque las acciones de estos usuarios son diferentes entre sí.

Si se hace clic en la opción "Gestiona tus ofertas" en la página web principal de alojamientos (Figura 24) como se ha mencionado anteriormente, el usuario será redirigido a la página de validación al panel de control (Figura 27).

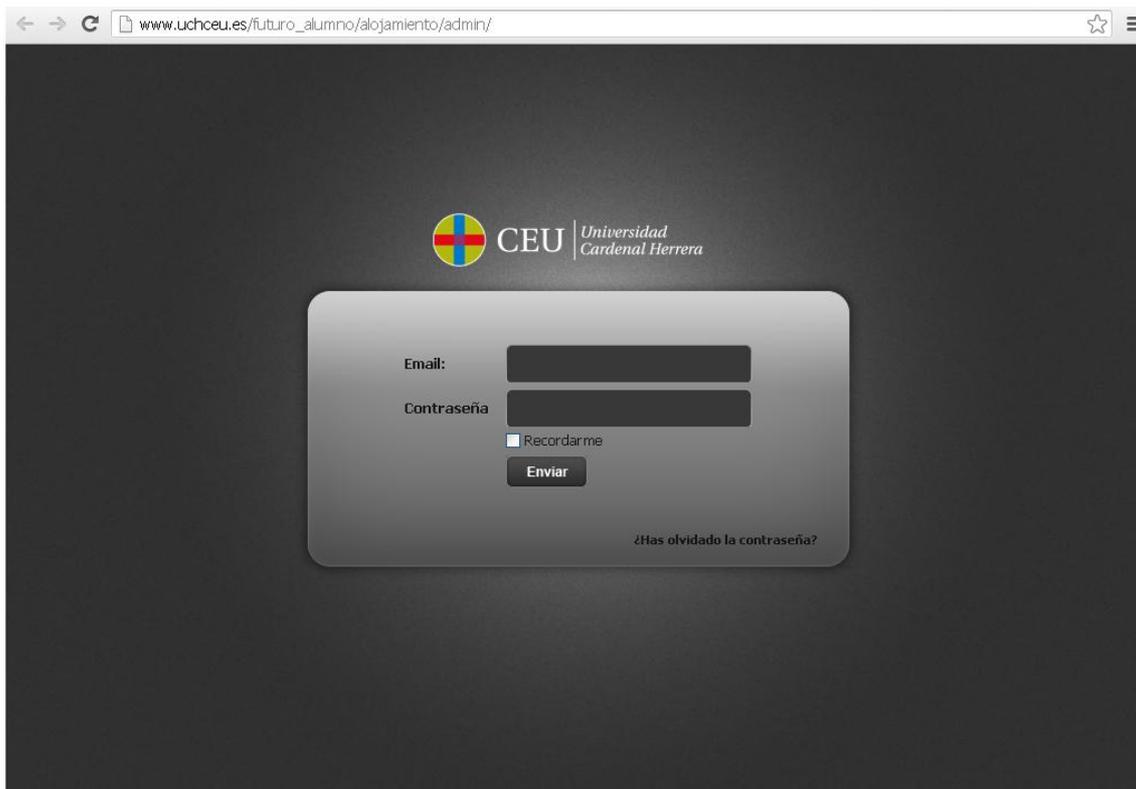


Figura 27: Página de validación - Usuario propietario

A la página de validación al panel de control para el usuario administrador se accede a través del siguiente enlace: <http://www.uchceu.es/admin> .

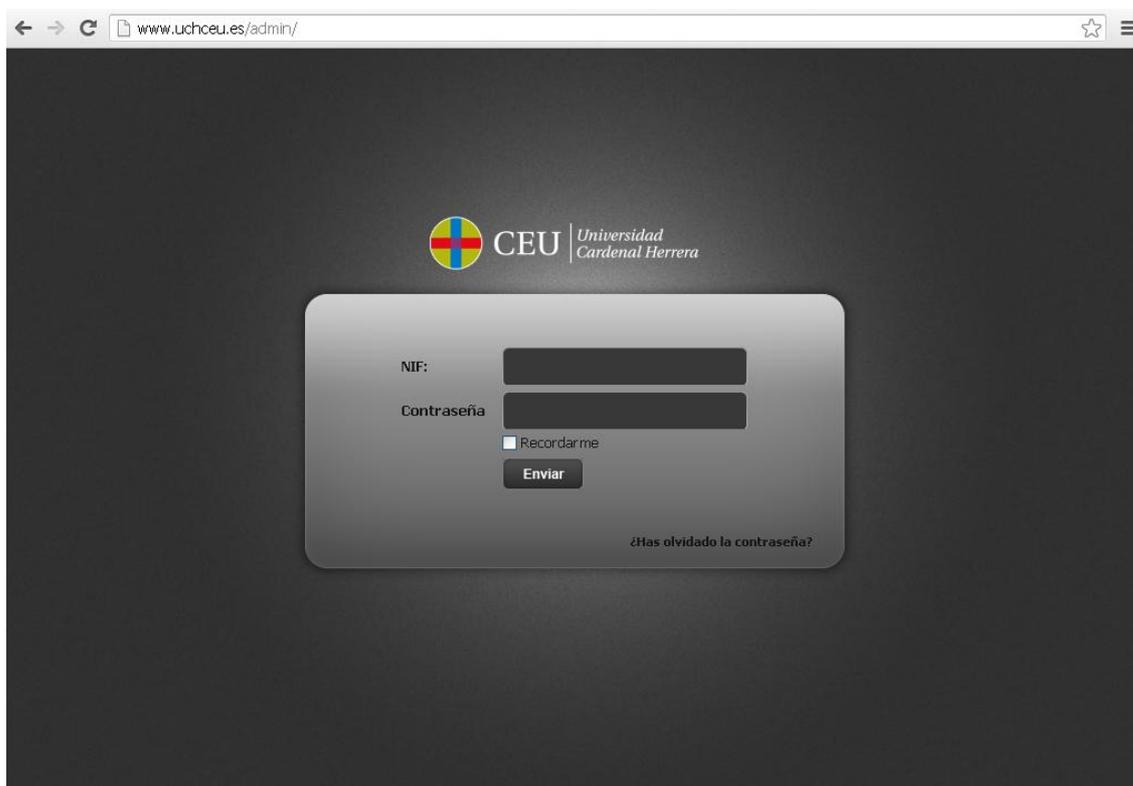


Figura 28: Página de validación - Usuario administrador

Aplicación web para la gestión de ofertas de alojamiento de la Universidad Cardenal Herrera

En la parte privada del usuario registrado se visualizan cuatro pestañas: “Crear nueva oferta”, “Listado ofertas”, “Mi cuenta”, “Salir”. La pantalla que se visualiza por defecto es la del listado propio del usuario de los inmuebles. Si dicho usuario no tiene creadas ninguna oferta de alojamiento, se ve lo siguiente:

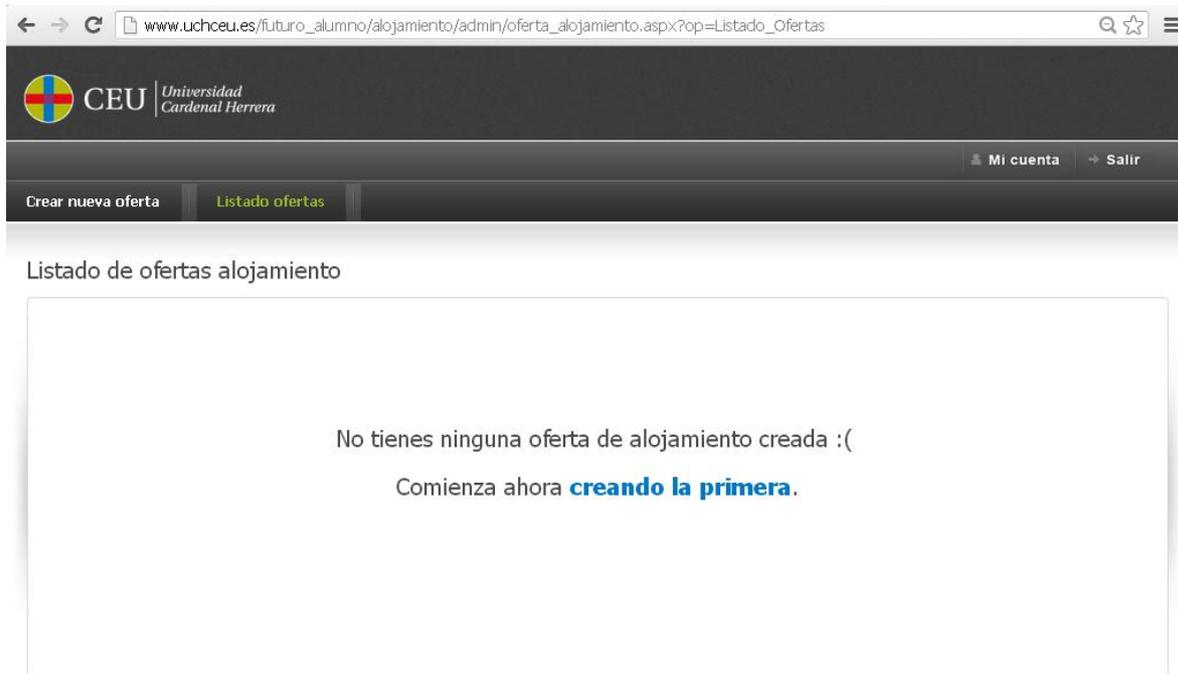


Figura 30: Intranet usuario propietario - Sin ofertas

Una vez que el usuario registrado crear ofertas de alojamiento, en la pestaña “Listado ofertas” se visualiza el listado de todos los inmuebles de dicho usuario (Figura 30).

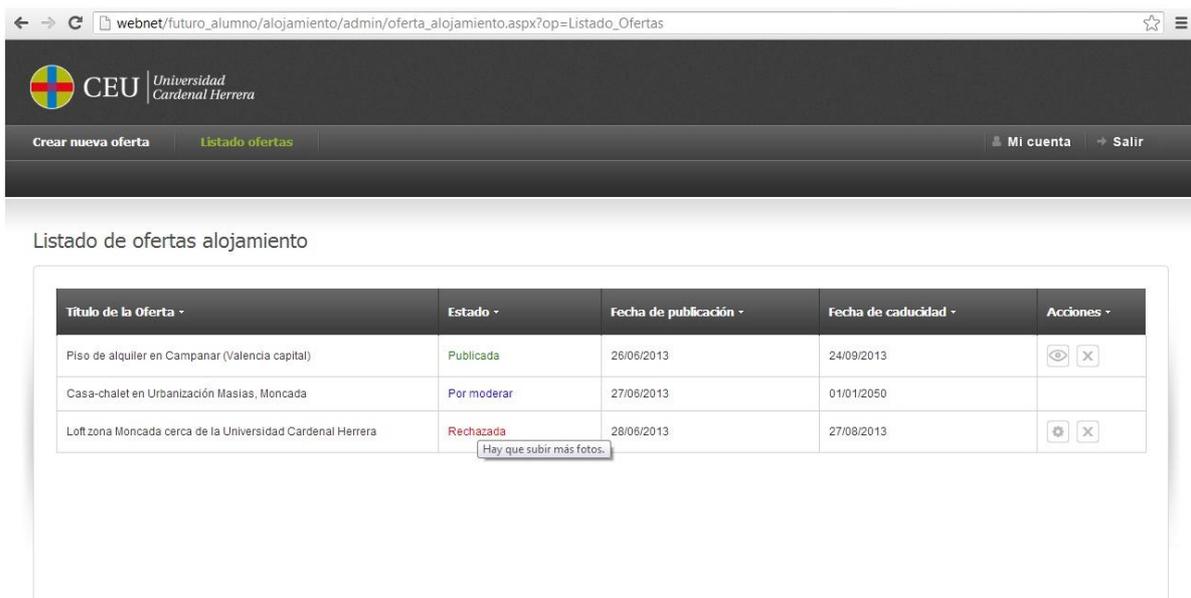


Figura 29: Intranet usuario propietario - Listado ofertas

En la pestaña “Mi cuenta” el propietario puede ver y/o modificar sus datos personales o darse de baja como usuario de la aplicación (Figura 32).

Modificar datos de la cuenta

Nombre:

Apellidos:

Teléfono:

¿Cómo quieres que contacten contigo? Teléfono Email

Correo electrónico:

Contraseña:

Repetir contraseña:

Borrar la cuenta

Puedes darte de baja de nuestro sistema [borrando tu cuenta](#)

Figura 32: Intranet usuario propietario - Mi cuenta

A través del botón “Salir” situado en la parte superior derecha de la página web el usuario puede cerrar la sesión y será redirigido automáticamente a la página web del listado público de inmuebles disponibles.

5.2 Capa de negocio

La capa de negocio guarda toda la funcionalidad de las entidades que se han definido en el sistema y aquellas que hacen de intermediarias entre la capa de presentación y la capa de datos.

Toma información de la base de datos según las necesidades del usuario y del sistema. También recogerá información suministrada por el usuario mediante la capa de presentación y la almacenará en la base de datos.

Los componentes de software que implementan el comportamiento de la solución vienen definidos por el conjunto de archivos que forman la aplicación (Figura 33).

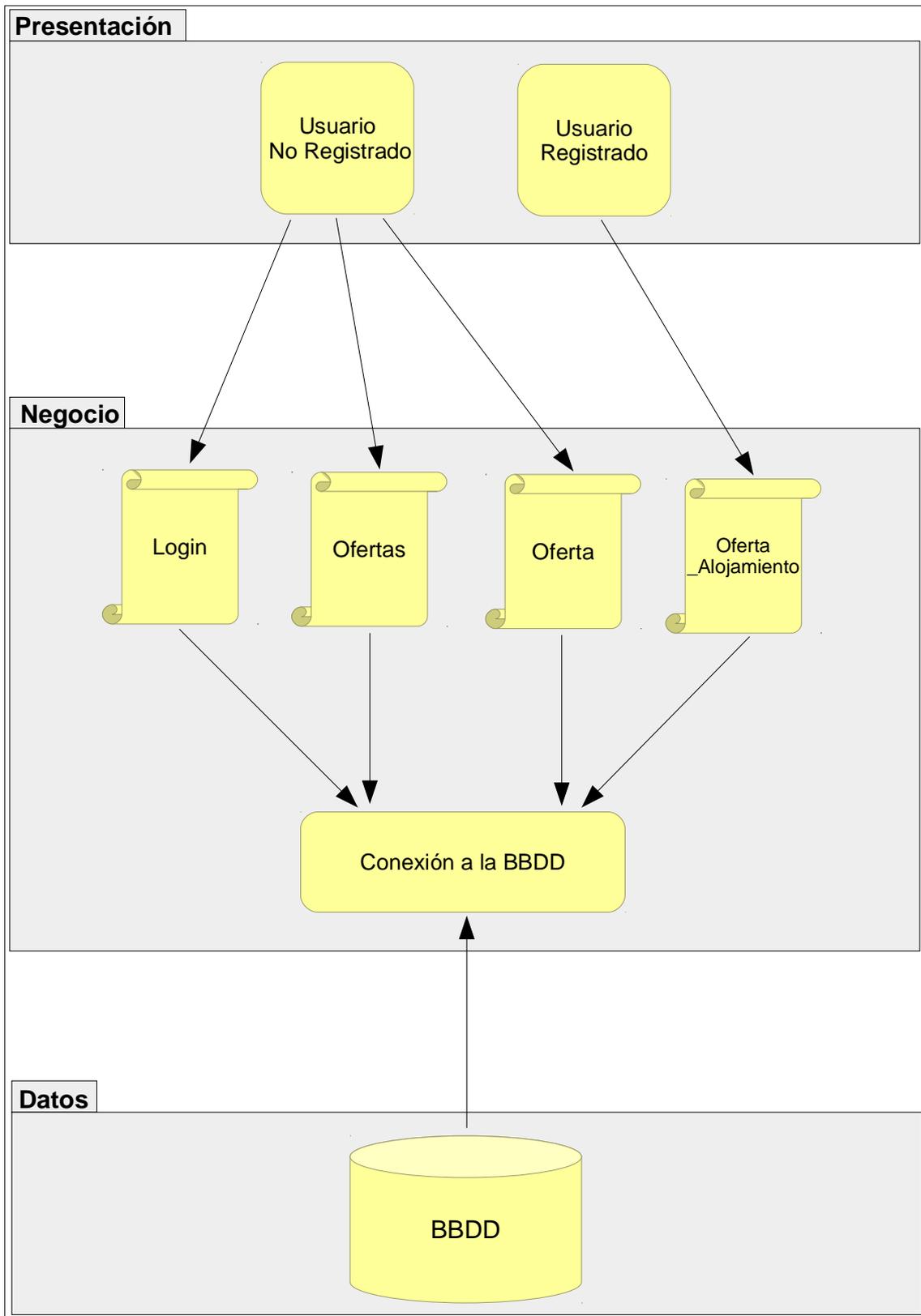


Figura 33: Capa de negocio

El módulo de negocio está organizado en varios componentes (Figura 33).

El analizador de *Login* será el encargado de comprobar si el usuario está registrado correctamente y permitir la carga de información de la base de datos; para luego redirigir al usuario de la zona anónima a su zona privada como usuario registrado.

El componente *Ofertas*, permitirá crear un nuevo usuario para acceder al área privada y, además, visualizar el listado de las ofertas de alojamiento publicadas en la página web.

El componente *Oferta* será el encargado de mostrar todos los datos de una oferta de alojamiento dada.

El módulo *Oferta_Alojamiento* mostrará los inmuebles propios para un usuario registrado. Además permitirá crear nuevas ofertas de alojamiento y ver/modificar los datos personales del usuario.

El módulo *Conexión a la BBDD* incluye las funciones necesarias para realizar la conexión a la base de datos.

5.3 Capa de datos

La capa de datos es aquella donde se almacenan todos los datos de la aplicación. En el caso de este proyecto se trata de una base de datos, pero podría estar formado por archivos de texto, documentos XML, etc.

Seguidamente, se puede ver el diagrama entidad-relación, con diversas relaciones entre las tablas de la base de datos.

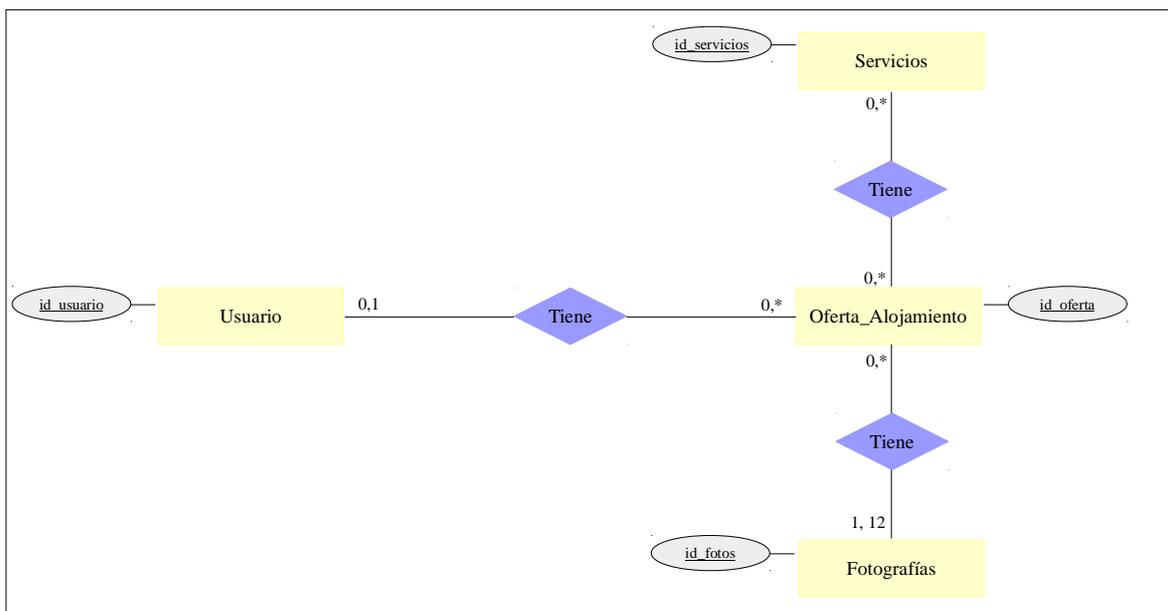


Figura 34: Diagrama entidad-relación

6. Implementación

A lo largo de este proyecto se han utilizado diferentes tecnologías, las cuales van a ser explicadas a continuación.

6.1 Tecnologías de implementación web

-Nivel de presentación

- **HTML** (HyperText Markup Language): es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de “etiquetas”, rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar al comportamiento de navegadores Web y otros procesadores de HTML. HTML también es usado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores). Por convención, los archivos de formato HTML usan la extensión .htm o .html. HTML consiste de varios componentes vitales, incluyendo elementos y sus atributos, tipos de datos y la declaración del tipo de documento.
- **CSS** (Cascading Style Sheets): lenguaje formal usado para definir la presentación de documentos estructurados escritos en HTML, XML y XHTML por extensión. El World Wide Web Consortium (W3C) es el organismo encargado de formular la especificación de las hojas de estilo que servirán como estándar para los navegadores. La idea que se reside tras el desarrollo de CSS es separar la estructura de un documento de su presentación.
- **JavaScript**: es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java. Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien un lenguaje basado en prototipos, a que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM.
- **JQuery**: es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a las páginas web. JQuery es software libre y de código abierto que, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.



- **Ajax** (Asynchronous JavaScript And XML): es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

-Nivel de aplicación

- **C#:** es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes (más notablemente de Delphi y Java). El símbolo # se compone de cuatro signos '+' pegados, indicando así su descendencia de C++. C#, como parte de la plataforma.NET, está normalizado por ECMA desde diciembre de 2001 (ECMA-334 "Especificación del Lenguaje C#"). El 7 de noviembre de 2005 salió la versión 2.0 del lenguaje que incluía mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables. El 19 de noviembre de 2007 salió la versión 3.0 de C# destacando entre las mejoras los tipos implícitos, tipos anónimos y el LINQ (Language Integrated Query). La última versión estable, 5.0, salió el 15 de agosto de 2012. Aunque C# forma parte de la plataforma.NET, ésta es una interfaz de programación de aplicaciones; mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma.
- **ASP .NET:** es un framework para aplicaciones Web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios Web dinámicos, aplicaciones Web y servicios Web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.
- **Visual Studio:** Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros. Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones Web, así como servicios Web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas Web y dispositivos móviles. Para el desarrollo de dicho proyecto se ha utilizado la versión VS2010.

-Nivel de persistencia

- **SQL SERVER:** Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, Sybase ASE, PostgreSQL o MySQL.

6.2 Descripción de la implementación

A continuación se van a exponer algunos fragmentos de código para poner ejemplos de cómo se ha realizado la implementación de la aplicación.

6.2.1 Parte pública

La parte pública de la aplicación está formada por ficheros como ofertas.aspx, oferta.aspx, login.aspx. El resto de ficheros tienen un control de validación para no permitir acceder a usuarios sin privilegios.

login.aspx

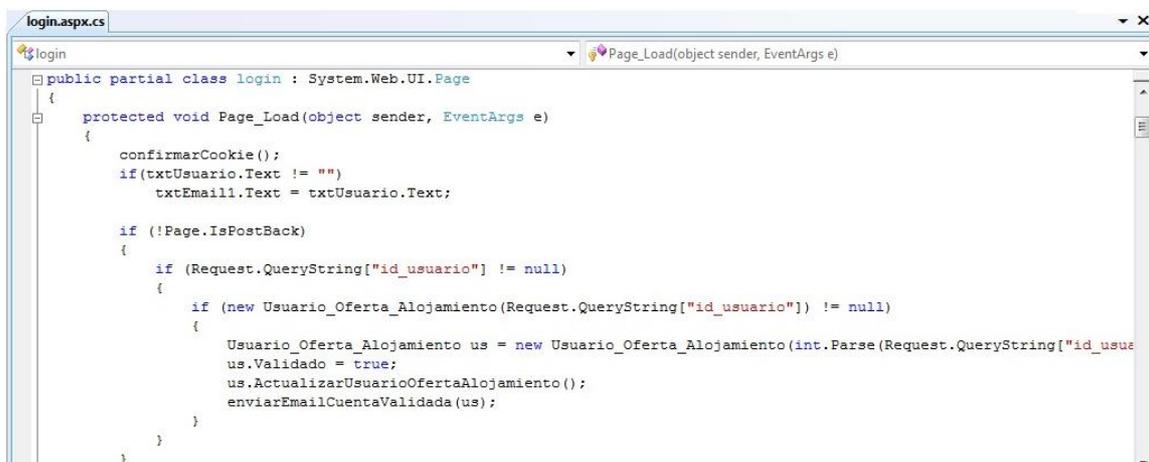
The image displays two screenshots of the login.aspx page for CEU Universidad Cardenal Herrera. The left screenshot shows the login form with fields for Email, Contraseña, and a 'Recordarme' checkbox, along with an 'Enviar' button and a link for forgotten passwords. The right screenshot shows the password reset form with a 'Dirección de correo' field and an 'Enviar' button, along with a 'Volver al login' link.

Es la página encargada de autenticar a los usuarios, abriendo una sesión por usuario internamente, es necesario introducir un email y una contraseña, también están las opciones de “Recordarme” y la de “Recuperar contraseña”.



```
<!-- start login-inner -->
<div id="login-inner">
  <table border="0" cellpadding="0" cellspacing="0">
    <tr>
      <th><asp:Label ID="UserNameLabel" runat="server">Email:</asp:Label></th>
      <td><asp:TextBox ID="txtUsuario" runat="server" CssClass="login-inp"></asp:TextBox></td>
    </tr>
    <tr>
      <th><asp:Label ID="PasswordLabel" runat="server">Contrase&ntilde;a:</asp:Label></th>
      <td><asp:TextBox id="txtPassword" runat="server" TextMode="Password" CssClass="login-inp"></asp:TextBox></td>
    </tr>
    <tr>
      <td colspan="2"><asp:CheckBox id="PersistCookie" runat="server"></asp:CheckBox> Recordarme</td>
    </tr>
    <tr>
      <td colspan="2"><asp:button id="btnConectar" runat="server" onclick="btnConectar_Click" CssClass="submit-login"></asp:button>
    </tr>
  </table>
</div>
```

Figura 35: login.aspx



```
public partial class login : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        confirmarCookie();
        if(txtUsuario.Text != "")
            txtEmail1.Text = txtUsuario.Text;

        if (!Page.IsPostBack)
        {
            if (Request.QueryString["id_usuario"] != null)
            {
                if (new Usuario_Oferta_Alojamiento(Request.QueryString["id_usuario"]) != null)
                {
                    Usuario_Oferta_Alojamiento us = new Usuario_Oferta_Alojamiento(int.Parse(Request.QueryString["id_usue
                    us.Validado = true;
                    us.ActualizarUsuarioOfertaAlojamiento();
                    enviarEmailCuentaValidada(us);
                }
            }
        }
    }
}
```

Figura 36: login.aspx.cs

ofertas.aspx

A continuación se muestra el contenido principal de la aplicación. El contenido muestra siete ofertas de alojamiento, empezando por la destacada seguida de las más recientes, por cada página y las muestra junto con una breve descripción. Es posible acceder a información más

detallada de cada inmueble haciendo clic sobre el alojamiento (fotografía, título o botón de la derecha “Ver información y fotos”).

Además, desde esta página se puede acceder a la página del login arriba mencionada y al formulario de registro de los usuarios que se muestra seguidamente.

Futuro Alumno > Ofertas Alojamiento

Publica tu alojamiento gratis: Dale de alta | Gestiona tus ofertas

Tipos: Todos | Precios: Todos | Provincias: Todas | Localidad: Todas | Ordenar por: Fecha de publicación

COLEGIO MAYOR AUSIAS MARCH
 Tipo: Colegio mayor
 Ausias March, el Colegio Mayor de Valencia. BUS PRIVADO de puerta a puerta con servicios de mañana y tarde (14 viajes al día). PENSIÓN COMPLETA FINES DE SEMANA Y FESTIVOS INCLUIDOS sin coste adicional.
 687,50 euros/mes
 15 m²
 100 Hab.
 Situado en: VALENCIA
 Dirigido a: chicos y/o chicas
[Ver información y fotos >>](#)

ENORME PISO CON TERRAZA EN BENIMACLET
 Tipo: Piso compartido
 HAB A 310/300€ (doble,20m², cama 135, balcon) *1persona2persona
 HAB B 200€ (individual,20m², cama 90 con cama) HAB C 250/300€ (sola,10m², cama 135) HAB D 240€ (individual, cama 105) PRECIOS
 200 euros/mes
 100 m²
 4 Hab.
 Situado en: VALENCIA
 Dirigido a: Solo chicas
[Ver información y fotos >>](#)

PISO IDEAL ESTUDIANTE S BENIMACLET
 Tipo: Piso compartido
 hab a: 230 € (individual, balcon, cama 90) hab b: 250€ (individual, cama 105) hab c: 230 € (individual, cama) alquiler de 3 habitaciones individuales en piso situado en 1ª planta edificio s...
 230 euros/mes
 65 m²
 3 Hab.
 Situado en: VALENCIA
 Dirigido a: Solo chicas
[Ver información y fotos >>](#)

PISO COMPARTIDO EN BENIMACLET
 Tipo: Piso compartido
 Alquiler de habitaciones en piso compartido en benimaclet. Situado en la 3ª planta sin ascensor posee unas magnificas vistas de todo el barrio, techos muy altos y gran luminosidad. Tiene 3 habitaciones...
 220 euros/mes
 75 m²
 3 Hab.
 Situado en: VALENCIA
 Dirigido a: Solo chicas
[Ver información y fotos >>](#)

PISO EN MONCADA CERCA DEL CEU
 Tipo: Piso
 Primer piso todo exterior, amplio salón con terraza, 3 habitaciones confortables con armarios empotrados y amueblada para estudiantes. Doble ventana climatiz. 2 WC con bañera. A 5 minutos a pie del...
 495 euros/mes
 90 m²
 3 Hab.
 Situado en: MONCADA
 Dirigido a: chicos y/o chicas
[Ver información y fotos >>](#)

HABITACION CON BAÑO PRIVADO PARA CHICA DEL CEU, ON PARLE FRANCAIS
 Tipo: Piso compartido
 buscamos una chica que estudie en el CEU para completar piso con otras dos estudiantes de veterania, el piso tiene 4 años y dispone de todas las comodidades, lavadora, utensilios de cocina etc, la...
 220 euros/mes
 16 m²
 3 Hab.
 Situado en: ALFARA DEL PATRIARCA
 Dirigido a: Solo chicas
[Ver información y fotos >>](#)

FANTASTICO PISO PARA ALQUILAR
 Tipo: Piso
 Vivienda casi nueva totalmente amueblada y preparada para 3 estudiantes para vivir con todo tipo de comodidades. Cerca de Supermercados, Centro de Salud, Bancos, Zona de Ocio, etc.
 650 euros/mes
 100 m²
 3 Hab.
 Situado en: MONCADA
 Dirigido a: chicos y/o chicas
[Ver información y fotos >>](#)

«Primera» «Anterior» 2 / 11 «Siguiente» «Última»

ADVERTENCIA: La Universidad no se hace responsable ni de las condiciones de alojamiento ni de las condiciones de contrato de alquiler. Tiempo del comportamiento de los inquilinos ni del cumplimiento de los pagos. Este servicio pretende únicamente poner en contacto a los alumnos y a los propietarios por lo que no se establece ninguna relación con estos usuarios.

Compartir: Facebook | Twitter | Email | Print

Alojamiento

Conscientes del esfuerzo que supone la búsqueda de alojamiento al trasladarte a una nueva ciudad, te ofrecemos una aplicación online para que encuentres fácilmente un hogar en tu nuevo camino.

Futuro Alumno > Ofertas Alojamiento

DATE DE ALTA

Nombre: Apellidos:

Teléfono:

Correo electrónico:

¿Cómo quieres que contacten contigo? Teléfono Email

Definir contraseña:

Introduce el código que ves en la imagen

Acepto que mis datos de contacto (nombre, apellidos y teléfono) y e-mail sean visibles a través de la web de la Universidad CEU Cardenal Herrera.

Enviar Formulario

```

ofertas.aspx
Eventos y objetos de cliente (No hay eventos)
<<%= Page Language="C#" AutoEventWireup="true" MasterPageFile="~/controles/plantillas/plantilla_inivel.master" MaintainSc...
<<%= Register Src="~/futuro_alumno/controles/header.aspx" TagName="header" TagPrefix="uc1" %>
<<%= Register Src="~/controles/redes_sociales.aspx" TagName="redes_sociales" TagPrefix="uc2" %>
<asp:Content ContentPlaceHolderID="head" ID="cHead" runat="server">
  <link rel="stylesheet" href="/futuro_alumno/alojamiento/estilos.css" type="text/css" media="screen" />
</asp:Content>
<asp:Content ID="cont" runat="server" ContentPlaceHolderID="titulo">
  <asp:label runat="server" id="lblTitulo">Ofertas de Alojamiento</asp:label>
  <asp:label runat="server" id="lblDescription">Ofertas de Alojamiento para estudiantes de la Universidad CEU Cardenal
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="Imagenes_Carousel" Runat="Server">
  <uc1:header ID="ctrHeader" runat="server"></uc1:header>
</asp:Content>
<asp:Content ID="Contenedor" ContentPlaceHolderID="Contenedor" runat="server">
  <form id="form1" runat="server">
  <div>
  
```

Figura 37: ofertas.aspx



```

ofertas.aspx
ofertas
dvOfertas

public partial class ofertas : System.Web.UI.Page
{
    public DataView dvOfertas = Oferta_Alojamiento.ObtenerOferta_Alojamiento_todas_visibles();
    public Usuario_Oferta_Alojamiento usuario = new Usuario_Oferta_Alojamiento();
    DataView dvUsuarios = Usuario_Oferta_Alojamiento.ObtenerUsuario_Oferta_Alojamiento_todos();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            LoadDataOfertas(0);

            if (Request.QueryString["seccion"] == "login")
            {
                pnlListado.Visible = false;
                pnlDateDeAlta.Visible = true;
                hlAlta.NavigateUrl = "/futuro_alumno/alojamiento/ofertas.aspx";
                hlAlta.Text = "Listado de ofertas";
                lblPublicaGratis.Visible = false;
            }
            else if (Request.QueryString["seccion"] == "gracias")
        }
    }
}

```

Figura 38:ofertas.aspx.cs

oferta.aspx

Pantalla que muestra todos los detalles de un alojamiento dado. A la derecha hay un formulario para que los interesados puedan ponerse en contacto con el propietario del inmueble y debajo de éste se puede ver otros anuncios del mismo propietario (si es que los tiene).

Inicio
Futuro Alumno
Ofertas Alojamiento
COLEGIO MAYOR AUSIAS MARCH



Fotografía de: Recepción 24 h



[<< Volver al listado de ofertas de alojamiento](#)

COLEGIO MAYOR AUSIAS MARCH

Colegio mayor - 687,50 euros/mes
 15 m²
 100 habitaciones
 100 baños

Ausias March, el Colegio Mayor de Valencia. BUS PRIVADO de puerta a puerta con servicios de mañana y tarde (14 viajes al día). PENSIÓN COMPLETA FINES DE SEMANA Y FESTIVOS INCLUIDOS sin coste adicional.

Dirección:
 AV. CATALUÑA nº. 30
 46021 VALENCIA / Valencia

Núm. de plazas ofertadas: 30
Ofertado para ser ocupado por: Chicos y/o chicas

Servicios:

- Calefacción
- Televisión
- Teléfono fijo
- Internet
- Terraza
- Ascensor
- Garaje
- Amueblado
- Lavavajillas
- Lavadora
- Horno
- Microondas
- Nevera
- Gas
- Aire Acondicionado

Servicios adicionales:
 SERVICIO DE TRANSPORTE PRIVADO AL CEU. TURNO DE MAÑANA Y TARDE (14 VIAJES AL DÍA). SERVICIO DE PENSIÓN COMPLETA INCLUIDOS FINES DE SEMANA.

Otros gastos no incluidos:
 TODO INCLUIDO.

Compártelo:

Me gusta 0
+1 0
Twitter 0
Share

Publica tu alojamiento gratis:
[Date de alta](#) | [Gestiona tus ofertas](#)

Publicado el 12/07/2013 por:



Colegio Mayor Ausias March
 963695408

Contactar por email

Nombre:

Email:

Teléfono:

Mensaje:

Introduce el código

S56Kx

```

oferta.aspx
Eventos y objetos de cliente (No hay eventos)
<div id="columna_dos_430">
  <h1 style="font-size:20px;"><%= lblTitulo.Text %></h1>
  <div style="margin-left:15px;">
    <p><span style="font-size:18px;"><asp:Literal runat="server" ID="lblTipoAlojamiento"></asp:Literal></span> -
  <span style="margin-left:10px;"><asp:Literal runat="server" ID="lblM2"></asp:Literal> m<sup>2</sup></span><br />
  <span style="margin-left:10px;"><asp:Literal runat="server" ID="lblNumeroHab"></asp:Literal> habitaciones</span><br />
  <span style="margin-left:10px;"><asp:Literal runat="server" ID="lblBanyos"></asp:Literal> baños</span></p>
    <p><asp:Literal runat="server" ID="lblDescripcion"></asp:Literal></p>
    <p><strong>Dirección:</strong><br />
  <asp:Literal runat="server" ID="lblDireccion"></asp:Literal> n°. <asp:Literal runat="server" ID="lblNumero"></asp:Literal>
  <asp:Literal runat="server" ID="lblCodigoPostal"></asp:Literal> <asp:Literal runat="server" ID="lblLocalidad"></asp:Lite
  <br />
  <p><strong>Núm. de plazas ofertadas:</strong> <asp:Literal runat="server" ID="lblNumeroPlazas"></asp:Literal><br />

```

Figura 39: oferta.aspx

```

oferta.aspx.cs
futuro_alumno_oferta ObtenerPreferencia(int preferencia)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Web.UI.HtmlControls;
using System.Web.Mail;
using System.Text.RegularExpressions;

public partial class futuro_alumno_oferta : System.Web.UI.Page
{
  private string ObtenerPreferencia(int preferencia)
  {
    string pref = string.Empty;
    switch (preferencia)
    {

```

Figura 40: oferta.aspx.cs

6.2.2 Parte privada

La parte privada de la aplicación está formada por ficheros como oferta_alojamiento.aspx, borrador_oferta.aspx, login_admin.aspx, listado_admin.aspx, etc. Es necesario recordar que es imprescindible estar registrado en la aplicación con antelación para poder acceder a esta zona.

Seguidamente se muestran trozos de código de algunos de los ficheros arriba mencionados.

oferta alojamiento.aspx

Página encargada de la parte privada del usuario propietario. Permite ver el listado de inmuebles del usuario, crear nuevas ofertas de alojamiento, ver/modificar los datos personales del usuario y/o darse de baja como usuario.

Aplicación web para la gestión de ofertas de alojamiento de la Universidad Cardenal Herrera

webnet/futuro_alumno/alojamiento/admin/oferta_alojamiento.aspx?op=Listado_Ofertas

CEU Universidad Cardenal Herrera

Crear nueva oferta Listado ofertas Mi cuenta Salir

Listado de ofertas alojamiento

Titulo de la Oferta	Estado	Fecha de publicación	Fecha de caducidad	Acciones
Piso de alquiler en Campanar (Valencia capital)	Publicada	26/06/2013	24/09/2013	
Casa-chalet en Urbanización Masias, Moncada	Por moderar	27/06/2013	01/01/2050	
Loft zona Moncada cerca de la Universidad Cardenal Herrera	Rechazada	28/06/2013	27/09/2013	

Hay que subir más fotos.

```

oferta_alojamiento.aspx
Eventos y objetos de cliente (No hay eventos)
<p><asp:Label ID="lblTipoAlojamiento" Width="150px" runat="server">Tipo de alojamiento:</asp:Label>
<asp:DropDownList ID="ddlTipoAlojamiento" runat="server"
  AppendDataBoundItems="true" ValidationGroup="Guardar" style="width:203px">
  <asp:ListItem Text="Selecciona una opción" Value="0" Selected="True" ></asp:ListItem>
  <asp:ListItem Value = "Colegio mayor" >Colegio mayor</asp:ListItem>
  <asp:ListItem Value = "Residencia universitaria" >Residencia universitaria</asp:ListItem>
  <asp:ListItem Value = "Residencia familiar" >Residencia familiar</asp:ListItem>
  <asp:ListItem Value = "Piso" >Piso</asp:ListItem>
  <asp:ListItem Value = "Piso compartido" >Piso compartido</asp:ListItem>
</asp:DropDownList>
<asp:RequiredFieldValidator
  ID="cvTipoAlojamiento"
  runat="server"
  ControlToValidate="ddlTipoAlojamiento"
  ErrorMessage="Es obligatorio seleccionar un tipo de alojamiento."
  Display="None"
  InitialValue="0"
  ValidationGroup="Guardar">
</asp:RequiredFieldValidator>

```

Figura 41: oferta_alojamiento.aspx

```

oferta_alojamiento.aspx.cs
futuro_alumno_admin_oferta_alojamiento dvOfertas
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Security;
using System.IO;
using System.Web.Mail;
using System.Collections;
using System.Data;
using System.Drawing;

public partial class futuro_alumno_admin_oferta_alojamiento : System.Web.UI.Page
{
    DataView dvOfertas = Oferta_Alojamiento.ObtenerOferta_Alojamiento_todas();
    public Usuario_Oferta_Alojamiento usuario = new Usuario_Oferta_Alojamiento();
    public Oferta_Alojamiento oferta = new Oferta_Alojamiento();
    DataView dvUsuarios = Usuario_Oferta_Alojamiento.ObtenerUsuario_Oferta_Alojamiento_todos();

    //paginacion

```

Figura 42: oferta_alojamiento.aspx.cs

borrador_oferta.aspx

Es prácticamente igual al fichero oferta.aspx que se ha mencionado en el apartado anterior. La diferencia está en que ésta página la puede ver solo el usuario administrador mientras una oferta de alojamiento no haya sido publicada. Este fichero ha sido creado por razones de seguridad, concretamente para que nadie pueda acceder a la página de una oferta de alojamiento sin que ésta haya sido moderada previamente por el administrador. Los borradores de ofertas se almacenan en una carpeta (llamada borrador) independiente de las demás ofertas de alojamiento. En la siguiente ilustración se puede apreciar la diferencia de la url de una oferta y de la url del borrador de una oferta (en esta última aparece la palabra “borrador” mientras que en la primera no).

www.uchceu.es/futuro_alumno/alojamiento/borrador/oferta_residencia_senoritas

CEU | Universidad Cardenal Herrera

ENGLISH FRANÇAIS

Cómo llegar | Correo | Intranet | Directorio | Empleo

La Universidad | Estudios | Vida Universitaria | Servicios | Relaciones Internacionales | Investigación

Google

Alojamiento

Conscientes del esfuerzo que supone la búsqueda de alojamiento al trasladarte a una nueva ciudad, te ofrecemos una aplicación online para que encuentres fácilmente un hogar en tu nuevo camino.

RESIDENCIA SEÑORITAS

Publica tu alojamiento gratis: [Date de alta](#) | [Gestiona tus ofertas](#)

Publicado el 13/08/2013 por: **Maria Jose Ros**
Palacios
666734266/696

Residencia familiar - 550 euros/mes
100 m²
3 habitaciones
1 baños

Habitaciones grandes individuales. Pensión completa. Céntrico, a 5 min del Ceu. Calefacción central y aire acondicionado.

Dirección:
Calle Algemesi nº. 10
46113 MONCADA / Valencia

Núm. de plazas ofertadas: 2
Ofertado para ser ocupado por: Solo chicos

Servicios:

- Calefacción
- Amueblado
- Horno
- Gas
- Televisión
- Lavavajillas
- Microondas
- Aire Acondicionado
- Internet
- Lavadora
- Nevera

Servicios adicionales:
Servicio comidas (desayuno, comida y cena de lunes a viernes)
Servicio limpieza habitaciones y zonas comunes, ropa de cama y toallas. Apoyo académico puntual GRATUITO!

Contactar por email

Nombre:

Email:

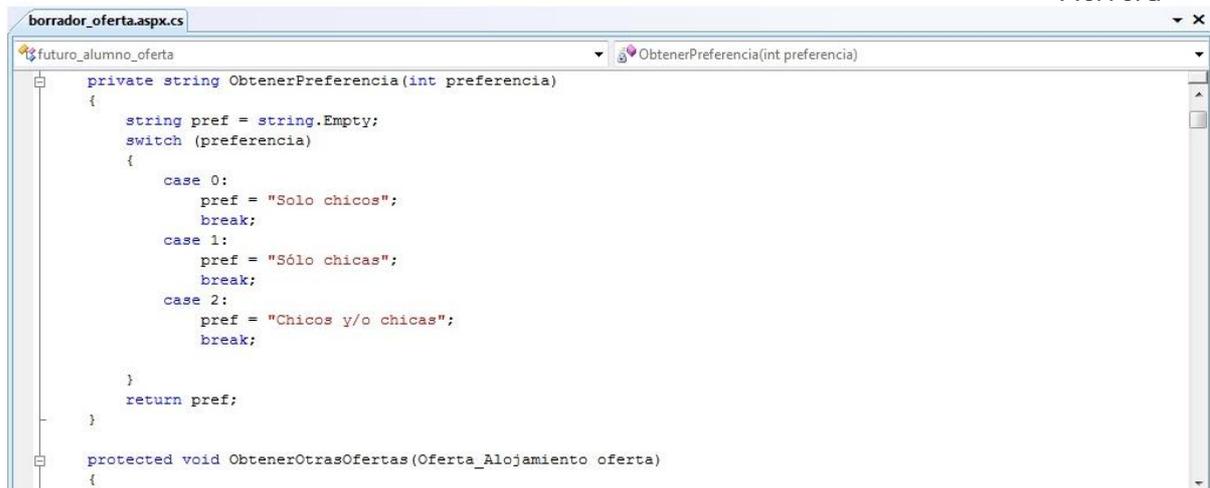
Teléfono:

Mensaje:

```
borrador_oferta.aspx
Eventos y objetos de cliente (No hay eventos)
<div id="columna_dos_430">
  <h1 style="font-size:20px;"><%= lblTitulo.Text %></h1>
  <div style="margin-left:15px;">
    <p><span style="font-size:18px;"><asp:Literal runat="server" ID="lblTipoAlojamiento"></asp:Literal></span> -
    <span style="margin-left:10px;"><asp:Literal runat="server" ID="lblM2"></asp:Literal> m<sup>2</sup></span><br />
    <span style="margin-left:10px;"><asp:Literal runat="server" ID="lblNumeroHab"></asp:Literal> habitaciones</span><br />
    <span style="margin-left:10px;"><asp:Literal runat="server" ID="lblBaños"></asp:Literal> baños</span></p>
    <p><asp:Literal runat="server" ID="lblDescripcion"></asp:Literal></p>
    <p><strong>Dirección:</strong><br />
    <asp:Literal runat="server" ID="lblDireccion"></asp:Literal> nº. <asp:Literal runat="server" ID="lblNumero"></asp:Literal>
    <asp:Literal runat="server" ID="lblCodigoPostal"></asp:Literal> <asp:Literal runat="server" ID="lblLocalidad"></asp:Lite
    <strong>Núm. de plazas ofertadas:</strong> <asp:Literal runat="server" ID="lblNumeroPlazas"></asp:Literal><br />
```

Figura 43: borrador_oferta.aspx





```
borrador_oferta.aspx.cs
futuro_alumno_oferta
ObtenerPreferencia(int preferencia)

private string ObtenerPreferencia(int preferencia)
{
    string pref = string.Empty;
    switch (preferencia)
    {
        case 0:
            pref = "Solo chicos";
            break;
        case 1:
            pref = "Sólo chicas";
            break;
        case 2:
            pref = "Chicos y/o chicas";
            break;
    }
    return pref;
}

protected void ObtenerOtrasOfertas(Oferta_Alojamiento oferta)
{
```

Figura 44: borrar_oferta.aspx.cs

7. Evaluación

En esta fase de evaluación concluye el ciclo de vida del proyecto y lo deja preparado para su uso. La fase de evaluación del proyecto medirá el nivel de calidad que ofrece la solución construida.

Para asegurar el correcto funcionamiento de las aplicaciones que componen este proyecto se han realizado diferentes pruebas, las cuales se van a describir a continuación.

7.1 Pruebas de Integración

En este tipo de pruebas se comprueba la compatibilidad y funcionalidad de los interfaces entre las distintas partes que componen un sistema, éstas pueden ser módulos, aplicaciones individuales, aplicaciones cliente/servidor, etc. Este tipo de pruebas es especialmente relevante en aplicaciones distribuidas.

En el caso concreto del proyecto a tratar, se ha comprobado la funcionalidad correcta de la aplicación mediante pruebas de creación, eliminación, edición de ofertas de alojamiento, registro de usuarios, etc.

7.2 Pruebas de sistema

- **Rendimiento:** determinan los tiempos de respuesta, el espacio que ocupa el módulo en disco o en memoria, el flujo de datos que genera a través de un canal de comunicaciones, etc.

Hay que tener en cuenta que en épocas de gran actividad como los meses de verano, que es cuando los alumnos comienzan a buscar alojamiento, pueden generar cantidad de información que deberá ser almacenada en la base de datos. Esto puede provocar lentitud y un tamaño importante de la misma por lo que quizás se debería contemplar algún sistema de copias de seguridad periódico.

- **Robustez:** determinan la capacidad del programa para soportar entradas incorrectas.

Con el amplio abanico de usuarios que han probado el sistema, se ha comprobado la capacidad del mismo ante entradas incorrectas comprobando, por ejemplo, que el formato de los datos introducidos en los formularios es correcto.

- **Seguridad:** se determinan los niveles de permiso de usuarios, las operaciones de acceso al sistema y acceso a datos.

A la parte pública de la aplicación puede acceder cualquier usuario, en cambio a la parte privada se puede acceder solo si se ha registrado previamente y se ha validado la cuenta de usuario.

El acceso a la base de datos está reservado al personal del departamento de comunicación digital y el departamento de informática de la universidad, por lo que se requiere el conocimiento de un usuario y contraseña dentro del dominio de personal.

- **Pruebas de aceptación:** Son las que hará el cliente, se determina que el sistema cumple con lo deseado y se obtiene la conformidad del cliente.

Tras verificar todas las pruebas técnicas que aseguran el correcto funcionamiento de las aplicaciones que conforman este proyecto se han llevado a cabo una serie de pruebas de validación que demuestren que las aplicaciones sean sencillas de usar y funcionen como se espera.

Para estas pruebas se ha contado con la colaboración de varios usuarios con diferente familiarización con la informática que se han prestado a utilizar las aplicaciones y evaluarlo.

- **Validación del código CSS:** puesto que se ha utilizado una hoja de estilo para la interfaz del portal web, se ha realizado una prueba de validación a través de la página web <http://jigsaw.w3.org/css-validator/> y el resultado ha sido válido (Figura 45).

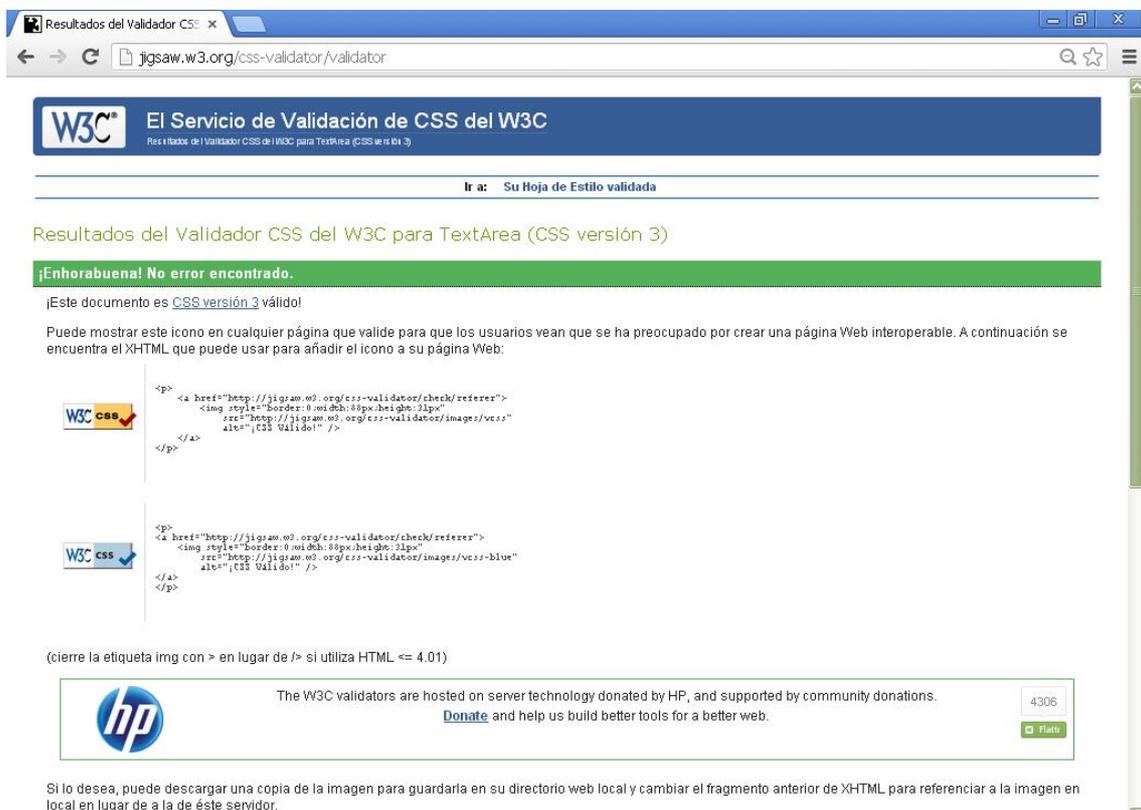


Figura 45: Validación del código CSS

- **Prueba de exploradores y resolución de pantalla:** Se ha comprobado el correcto funcionamiento de la aplicación tanto en Google Chrome como en Mozilla Firefox y Microsoft Explorer (Figuras 46, 47 y 48 respectivamente). El mayor problema existió durante la situación de las capas en sus lugares correspondientes y demás número de elementos y estilos.



Figura 46: Google Chrome

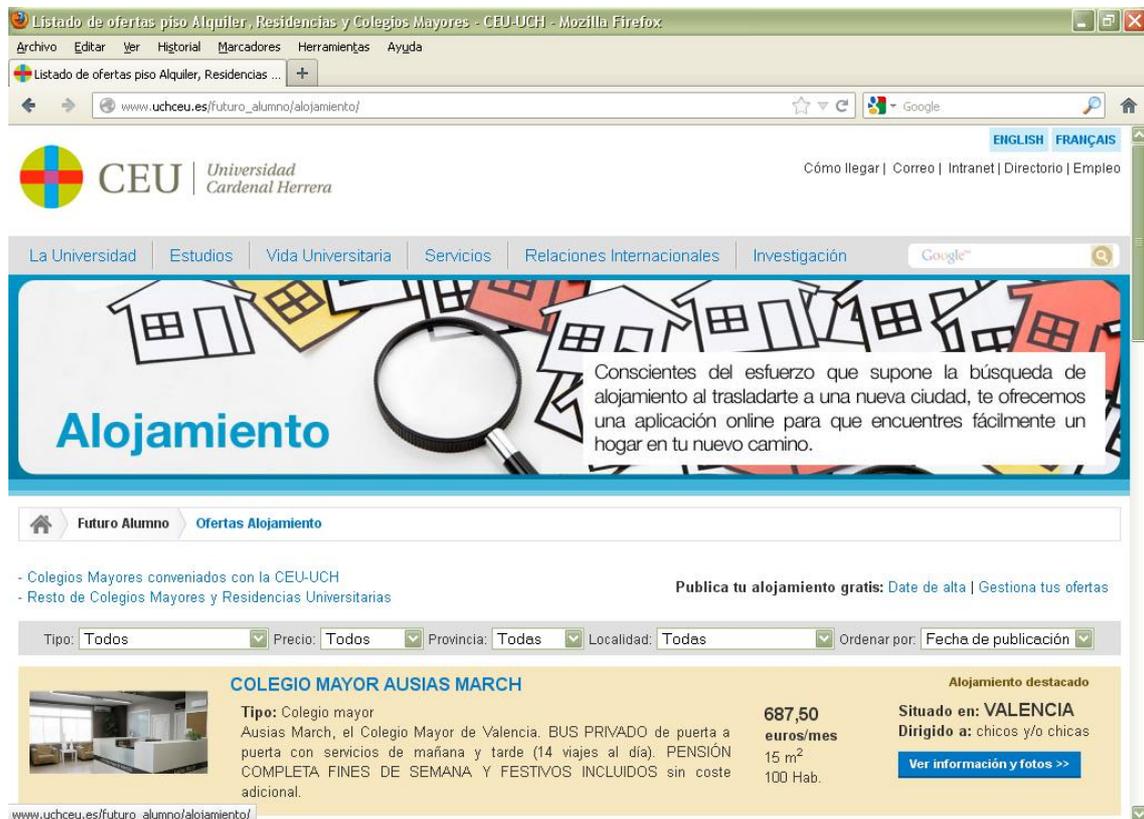


Figura 47: Mozilla Firefox

Aplicación web para la gestión de ofertas de alojamiento de la Universidad Cardenal

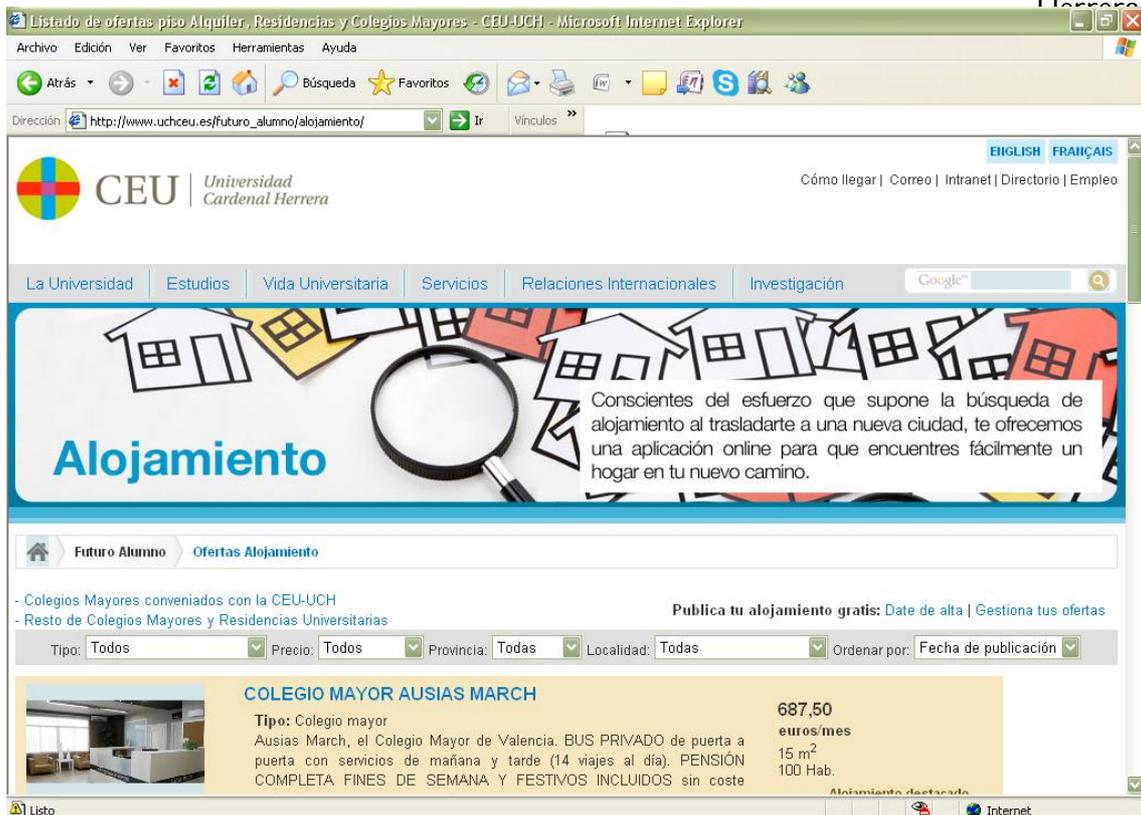


Figura 48: Internet Explorer

Se ha probado la aplicación en diferentes ordenadores a diferentes resoluciones, aunque se encuentra optimizado para 1280 x 1024 píxeles.

8. Conclusiones

En este proyecto final de carrera se ha desarrollado un espacio y una aplicación en el contexto de la web actual de la Universidad Cardenal Herrera que permite a sus alumnos visualizar las ofertas de alojamiento que mejor se ajusten a sus necesidades.

La aplicación responde a las expectativas y requerimientos recogidos en la especificación de requisitos. El trabajo realizado ofrece un alto nivel de cumplimiento de objetivo y es un punto de partida para que propietarios que pagan por la gestión de sus inmuebles, puedan desarrollar esta actividad vía internet con una aplicación gratuita y para facilitarles la búsqueda de alojamientos a los alumnos de la universidad.

La aplicación ofrece al usuario una interfaz intuitiva de fácil manejo y que tiene aspecto cuidado, atractivo y sencillo. La sencillez del proceso de gestión y la rapidez del proceso hacen de esta aplicación un gestor inmobiliario ideal para el usuario inexperto. Pretende atraer usuarios gracias a la sencillez y usabilidad del sistema. Añadir también que no precisa de ninguna instalación y está disponible vía web.

Desde el punto de vista personal decir que ha resultado una experiencia muy interesante haber llevado a cabo este proyecto ya que para terminarlo ha sido necesario aplicar gran parte de los conocimientos adquiridos durante los estudios y es muy gratificante ver la utilidad de los mismos a la hora de desarrollar un proyecto prácticamente real y completo, empezando por recabar la especificación de requisitos, planificación y estudio del proyecto, elección de tecnologías a utilizar, fase de implementación y hasta la fase de pruebas para que el usuario no tenga problema en su uso.

Al realizar el proyecto fin de carrera ha sido interesante poner a prueba todos los conocimientos que he ido adquiriendo a lo largo de la carrera. “Ingeniería del Software” me ha ayudado en la planificación a la hora de esquematizar y simplificar el proyecto en partes más pequeñas, así como en el desarrollo de diagramas UML. “Programación” y “Metodología de la programación” me han ayudado en temas de programación estructurada. Asignaturas relacionadas con la persistencia de información como “Diseño de Bases de Datos” y “Bases de Datos Avanzadas”. Finalmente la asignatura “Creación de documentos de hipertexto” me ha aportado conocimientos acerca de HTML y CSS.

El desarrollo de este proyecto también ha supuesto todo un reto ya que gran parte ha implicado la investigación de nuevas tecnologías que no se habían tratado a lo largo de la carrera tales como el desarrollo Web, el trabajo con librerías externas, el framework .Net y el gestor de bases de datos SQL Server.

Para finalizar, solo cabe mencionar que la realización de éste proyecto de final de carrera ha sido de un inestimable valor. Por falta de tiempo no he podido terminarlo como hubiese querido, pues considero que se pueden hacer mejoras y ampliaciones en los servicios que la web puede aportar a los usuarios. Aun así se han alcanzado los requisitos y cumplido los objetivos con éxito.

La aplicación se puso en marcha el día 23 de junio y al día de hoy cuenta con más de 135000 visitas (se ha comprobado mediante la herramienta Google Analytics) y se han registrado más de 70 ofertas de alojamiento (www.uchceu.es/alojamiento).

9. Bibliografía

- Documentaciones de las asignaturas ISG, PRG, MTP, BDA, DBO, CDH
- BUENDÍA GARCÍA F. 2008. Una guía para la realización y supervisión de proyectos final de carrera (PFC) en el ámbito de la web
- Programación en Visual C# .NET de Francisco Charte Ojeda.
- Memorias de otros proyectos finales de carrera
- msdn.microsoft.com
- www.wikipedia.org
- www.desarrolloweb.com
- www.codeproject.com
- www.microsoft.com

