



UNIVERSIDAD
POLITECNICA
DE VALENCIA

Escuela Técnica Superior de Ingeniería Informática



PROYECTO FINAL DE CARRERA

DESARROLLO DE UN SITIO WEB PARA UNA AGENCIA DE VIAJES TOURALIA

CURSO 2011/2012

Autor: Iván García García

Tutor: Félix Buendía García



*Dedico este proyecto a todas
aquellas personas que me han
apoyado en esta larga y dura
travesía, a mi familia y mis
amigos, sin cuyo apoyo,
esto no habría sido posible.*



Índice

1. Introducción	9
1.1. Objetivos	9
1.2. Contexto.....	10
1.3. Estructura de la memoria del proyecto	11
2. Especificación de requisitos.....	13
2.1. Introducción	13
2.1.1. Ámbito	13
2.1.2. Definiciones, siglas y asignatura	14
2.1.3. Referencias.....	15
2.1.4. Visión Global	15
2.2. Descripción General	16
2.2.1. Perspectiva del producto.....	16
2.2.2. Funciones del Producto	16
2.2.3. Características del usuario.....	18
2.2.4. Restricciones	19
2.3. Requisitos específicos.....	20
2.3.1. Requisitos de interfaz	20
2.3.2. Requisitos funcionales.....	22
2.3.3. Requisitos de contenido	22
3. Análisis	24
3.1. Diagramas de caso de uso	25



3.2. Diagrama de clases	28
3.3. Diagramas de secuencia	30
4. Diseño	34
4.1. Introducción	34
4.2. Arquitectura del sistema	34
4.2.1. Capa de presentación	36
4.2.2. Capa de negocio	39
4.2.3. Capa de bases de datos	41
5. Implementación	47
5.1. Introducción	47
5.2. Tecnologías	47
5.2.1. Capa de presentación	48
5.2.2. Capa Lógica o de negocio	51
5.2.3. Capa de datos.....	52
5.3. Herramientas	53
5.3.1. Adobe Dreamweaver 8.....	53
5.3.2. WAMP Server	54
5.3.3. Apache	56
5.3.4. GIMP 2	57
5.3.5. Notepad ++.....	58
5.4. Detalles implementación.....	59
5.4.1. Usuario visitante	60



5.4.2. Usuario cliente	68
5.5. Evaluación y Pruebas.....	69
5.5.1. Introducción	69
5.5.2. Pruebas	70
6. Conclusiones	76
7. Bibliografía	78
Anexos	79
Anexo A.....	79
Anexo B.....	82
Usuarios anónimos.....	82
Usuario registrado.....	84
Anexo C.....	87
1. Búsqueda de un hotel	87
2. Búsqueda de un vuelo	87
3. Búsqueda de un coche	88
4. Búsqueda de crucero.....	88
5. Registro de usuario	89
6. Inicio sesión.....	89
7. Reserva de hotel.....	90
8. Reserva de vuelos.....	90
9. Reserva coches.....	91
10. Reserva crucero.....	91



11. Modificar datos reserva	92
12. Eliminar reserva	93
Anexo D.....	94
Buscar_hoteles.tpl	94
BuscadorHoteles.php	101
Detalles_hotel.tpl.....	105
Ajax_hoteles.php	117
Anexo E	120
Detalle de implementación	120



Índice de ilustraciones

Ilustración 1. Web de muchoviaje.com	10
Ilustración 2. Web de booking.com	11
Ilustración 3. Esquema de la página principal.....	20
Ilustración 4. Actores del sistema.....	25
Ilustración 5. Casos de uso del actor usuario.....	26
Ilustración 6. Caso de uso del actor visitante	27
Ilustración 7. Diagrama de casos de uso del actor cliente	28
Ilustración 8. Diagrama de clases	29
Ilustración 9. Diagrama de secuencia 1	31
Ilustración 10. Diagrama de secuencia 2	32
Ilustración 11. Diagrama de secuencia 3	32
Ilustración 12. Arquitectura de tres capas	35
Ilustración 13. Ejemplo de la web usada como referencia.....	37
Ilustración 14. Barra de navegación	37
Ilustración 15. Banner publicitario	38
Ilustración 16. Panel de información de las ofertas.....	38
Ilustración 17. Temáticas del fórum	39
Ilustración 18. Pie de página.....	39
Ilustración 19. Capa de negocio.....	40
Ilustración 20. Diagrama entidad-relación.....	42



Ilustración 21. Adobe Dreamweaver 8	54
Ilustración 22. Ventana principal de Wamp Server.....	55
Ilustración 23. Gimp 2	57
Ilustración 24. Notepad ++	59
Ilustración 25. Vista de la página de búsqueda de hoteles	60
Ilustración 26. Consulta necesaria para el acceso a la página de búsqueda de hoteles	61
Ilustración 27. Vista de la página de detalles del hotel	62
Ilustración 28. Consulta para las búsquedas de hoteles	65
Ilustración 29. Página de acceso/inicio de sesión	66
Ilustración 30. Función de inserción de nuevo usuario.....	67
Ilustración 31. Funciones necesarias para la validación del usuario existente	67
Ilustración 32. Imagen de la página principal del fórum	68
Ilustración 33. Resolución de 640x480	70
Ilustración 34. Resolución de 800x600	71
Ilustración 35. Resolución de 1024x768	72
Ilustración 37. Vista del portal con Internet Explorer	73
Ilustración 36. Vista del portal con Mozilla Firefox.....	73
Ilustración 38. Vista del portal con Google Chrome.....	74
Ilustración 39. Vista del portal con Safari	74
Ilustración 40. Página de validación con el estándar w3c	75



1. Introducción

Este documento trata sobre la memoria del Proyecto Final de Carrera realizada por Iván García García, perteneciente a la titulación de Ingeniería Técnica Superior en Informática de Gestión, impartida en la Escuela Técnica Superior de Informática de la Universidad Politécnica de Valencia.

1.1. Objetivos

El propósito general del proyecto será desarrollar un sitio Web para una agencia de viajes con el fin de conseguir un producto que permita:

- Ofrecer información sobre hoteles, vuelos, cruceros y otros productos característicos de una agencia de viajes.
- Realizar búsquedas y reservas de los productos ofrecidos.
- Acceder a otros servicios como la consulta de ofertas o realización de comentarios en el foro, etc.

Para llevar a cabo dichos objetivos ha sido necesario el estudio en profundidad de las diversas tecnologías utilizadas para el desarrollo de aplicaciones Web así como el estudio de algunas empresas que ya ofrecen servicios de agencias de viajes en la red y que ha servido para comprender el tipo de contenido y funcionalidad que deberá tener la aplicación a desarrollar.

1.2. Contexto

Una idea del funcionamiento de dicha aplicación Web puede verse reflejada en las distintas páginas de agencias de viajes que utilizamos como referencia para abordar el proyecto, como la página Web de Viajes El Corte Inglés, Bancaja Viajes, Atrápalo, Halcón Viajes etc. sirviéndonos como ejemplo para elaborar el trabajo, ya que ayudarán a plasmar mejor la funcionalidad que tendrá la aplicación.

En la Ilustración 1 podemos ver en ejemplo la de la web de muchoviaje.com, mientras que en la Ilustración 2, otro de la web de Booking.com

The screenshot shows the muchoviaje.com website interface. At the top, there's a navigation bar with categories like Hoteles, Vuelos, and Vacaciones. A search box is visible with 'Madrid' as the destination and '16/4/2012' as the start date. Below the search box, there are several promotional banners and a grid of offers. The offers are categorized into 'Especial Primavera', 'Ofertas Puente Mayo', and 'Chollos'. Each offer includes the destination, the type of package (e.g., Hotel + Spa, Vuelo + Hotel), and the price starting from a certain amount.

Categoría	Destino	Detalle	Precio
Especial Primavera	Hoteles en Familia	Híjico Gratis	DESDE 15 €
	Denia	Hotel 5* Spa	DESDE 48 €
	Fin de semana	Ofertas	DESDE 20 €
Ofertas Puente Mayo	Lloret de Mar	desde	16 €
	Torremolinos	desde	14 €
	Roquetas	desde	12 €
Chollos	Crucero 2x1		781 €
	Santillana del Mar		19 €
	Crucero 8 días		129 €

Ilustración 1. Web de muchoviaje.com



Ilustración 2. Web de booking.com

1.3. Estructura de la memoria del proyecto

Para poder realizar la implementación del sitio Web con todas las características y funcionalidades que pueda tener una agencia de viajes online se ha seguido un proceso de desarrollo dividido por fases y que se resume en los siguientes puntos:

- **Capítulo 1:** Introducción. Objetivos y estructura del proyecto.
- **Capítulo 2:** Especificación de requisitos. Comprende la especificación de todos los requisitos que debe cumplir el producto a desarrollar.



- **Capítulo 3:** Análisis. Representación mediante modelos de los aspectos generales de la aplicación a desarrollar.
- **Capítulo 4:** Diseño. Arquitectura y composición de la aplicación.
- **Capítulo 5:** Implementación. Se describen todos los servicios y funciones implementados, atendiendo a la tecnología utilizada.
- **Capítulo 6:** Conclusión. Valoración y comentario final sobre el proyecto realizado.
- **Capítulo 7:** Bibliografía. Textos utilizados para la realización del proyecto.

Además se incluyen diversos anexos para complementar el detalle de los apartados anteriores.



2. Especificación de requisitos

2.1. Introducción

El propósito de esta ERS es servir de base para enunciar las características de diseño de interfaz, contenidos y funcionalidades del proyecto que se va a desarrollar y especificar los objetivos que quieren conseguirse en éste.

2.1.1. Ámbito

Este proyecto está orientado a la elaboración de un portal Web dedicado a viajes. La idea de llevar a cabo este proyecto es debido a que el turismo hoy en día se ha convertido en una necesidad básica de las personas. Hace unos cuantos años el turismo estaba destinado a aquellas personas con un alto nivel adquisitivo pero, poco a poco, ha ido introduciéndose en nuestra vida hasta llegar al punto de convertirse en una necesidad. Además, es una de las principales fuentes de ingreso de nuestra economía. Por todo ello, hemos decidido llevar a cabo este proyecto cuya intención es facilitar, ayudar y ofrecer todos los servicios necesarios para que nuestros usuarios disfruten de unas vacaciones de alta calidad.

El producto permitirá reservar billetes de avión, de autobús o de barco así como consultar sus tarifas, horarios, etc. También reservar estancias en hoteles y consultar información acerca de éstos y contratar otros servicios.

2.1.2. Definiciones, siglas y asignatura

Definiciones:

- **Portal Web:** Página de inicio que permite el acceso a las distintas secciones de un Sitio Web. Puede tener servicios como: *Chat, e-mail, foros, blogs, etc.*
- **Usuario:** Cualquier persona que hace uso de la aplicación.
- **Navegador:** Aplicación software que permite visualizar páginas de hipertexto.
- **Sitio Web:** Conjunto de páginas web bajo un mismo tema que son compartidas por un mismo dominio en Internet.
- **Dominio:** Denominación que identifica a un sitio en la red y expresa el tipo de actividad.
- **Proceso:** Conjunto de acciones que ejecutadas realizan y producen una serie de eventos en la aplicación.
- **Aplicación:** Programa preparado para una utilización específica, en este caso un Sitio Web que gestiona una agencia de viajes.
- **Servicio:** Conjunto de beneficios que se pueden adquirir en base a una remuneración.
- **Interfaz:** Mecanismo software que permite el trasvase de información entre varias aplicaciones o entre el propio programa y el usuario.
- **Estrella:** Categoría según la calidad de los servicios ofertados([1...5]).

Siglas:

- **PC:** Pensión completa (desayuno, almuerzo y cena)
- **MP:** Media pensión (desayuno y a elegir entre almuerzo y cena)
- **AD:** Alojamiento y desayuno
- **SA:** Sólo alojamiento



- **UML:** Lenguaje Unificado de Modelado. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

2.1.3. Referencias

- IEEE Std. 830-1998 Guía del IEEE para la Especificación de Requisitos.
- "Una Guía para la Realización y Supervisión de Proyectos Final de Carrera (PFC) en el ámbito de la Web" de la Editorial UPV (REF. 247).
ISBN 978-84-8363-325-0

2.1.4. Visión Global

Esta sección consta de tres partes. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda parte del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste realiza, los datos asociados y los variables además de las restricciones, sin entrar en excesivos detalles.

Por último, la tercera parte del documento es aquella en la que se definen detalladamente los requisitos que satisface el sistema.



2.2. Descripción General

2.2.1. Perspectiva del producto

El portal Web Touralia es un producto totalmente independiente que debe gestionar las incidencias habituales que se tratan en una Agencia de Viajes en un día cualquiera con la intención de satisfacer lo máximo posible al cliente.

El sistema estará incluido en un servidor Web. Escogeremos Mysql como base de datos por ser esta una base de datos robusta, de gran almacenamiento y de soporte permanente que permitirá almacenar los perfiles de clientes, hoteles, vuelos, coches de alquiler, etc. además de los datos transmitidos a la hora de realizar reservas. Esta plataforma podrá ser accedida por los clientes, ya sean anónimos o registrados (requisito necesario para realizar la reserva) y accederán a esta plataforma desde cualquier equipo con un navegador de Internet.

2.2.2. Funciones del Producto

En esta web se van a ofertar productos relacionados con los viajes, tales como hoteles, vuelos o cruceros. El usuario podrá escoger de forma sencilla entre las ofertas destacadas de la página web o elegir un servicio de forma personalizada, escogiendo detalles como el destino, la duración de la estancia en hoteles, las fechas de ida y regreso de un vuelo, entre otras.

Cuando el usuario haya escogido el producto, deberá efectuar un registro en el sistema para poder formalizar la reserva de dicho servicio. Para ello, deberá rellenar un formulario proporcionando sus datos personales, además de un nombre de usuario y una contraseña.



El registro en el sistema es indispensable para la contratación de productos y servicios en la web. Además servirá para que los clientes, que ya hayan utilizado los servicios del portal web, puedan dejar su opinión sobre el hotel, el vuelo, etc. en la sección de comentarios que el sitio web pondrá a disposición. También será de mucha utilidad a aquellos posibles futuros clientes, de momento usuarios, que con un simple registro podrá acceder al foro para hacerse una idea de lo que le espera si adquiere nuestros servicios.

A continuación, expondremos por separado las diferentes funciones que pueden realizar los distintos usuarios:

▪ **Usuarios anónimos:**

- **Consulta de ofertas:** la web propone hoteles a precios especiales.
- **Búsqueda de hoteles:** localizar hoteles acorde a un destino y unas fechas.
- **Búsquedas de vuelos:** localizar vuelos acorde a un destino y unas fechas.
- **Búsquedas de cruceros:** localizar cruceros acorde a un destino y unas fechas.
- **Búsquedas de coches en alquiler:** localizar vehículos para alquilar acorde a un destino y unas fechas.
- **Registrar en el sistema:** el usuario rellena un formulario con sus datos personales, proporcionando un nombre de usuario y una contraseña.

- **Usuarios registrados**, que además de poder realizar las acciones de los anónimos pueden realizar las siguientes:
 - **Reservar hotel:** el usuario registrado reserva un hotel para un destino y fechas seleccionadas.
 - **Reservar vuelo:** el usuario registrado reserva un vuelo para un destino y fechas seleccionadas.
 - **Reservar cruceros:** el usuario registrado reserva un crucero para un destino y fechas seleccionadas.
 - **Alquilar coches:** el usuario registrado alquila un coche para un destino y fechas seleccionadas.
 - **Dejar comentarios en el foro:** el usuario registrado puede dejar comentarios acerca de los servicios utilizados, manifestando su opinión.
 - **Autenticarse:** el usuario puede iniciar sesión identificándose previamente.
 - **Anular reserva:** el cliente puede solicitar a la empresa la anulación de su reserva.
 - **Modificar reserva:** el cliente puede modificar los datos correspondientes a su reserva.
 - **Consulta reserva:** el cliente puede comprobar los datos de la misma.

2.2.3. Características del usuario

Los usuarios que accedan a la página web deben ser personas, lógicamente que dispongan de un ordenador con conexión a internet y que se manejen bien en la red. En este sentido se identifican hasta tres usuarios distintos, que son los que siguen:



- **Usuarios anónimos:** son los usuarios que no están registrados en el sistema, y que por tanto, menos privilegios tienen. En su caso, tan sólo pueden consultar ofertas o noticias, sin tener posibilidad de interactuar con el sistema.
- **Usuarios registrados:** son los usuarios que están registrados en el sistema y en consecuencia, pueden interactuar con el sistema pudiendo efectuar reservas, comentando noticias, entre otros.

2.2.4. Restricciones

La finalidad de este proyecto es puramente académica. Es por ello, que en esta web los servicios que se oferten serán más limitados que en cualquier página web que esté operativa en la red, y por tanto, las bases de datos sean menores. Es posible, además, que algunos de los datos aparecidos en ella no sean reales.

Por otra parte, la contratación de los servicios ofertados, será lógicamente virtual, con lo cual, las transferencias bancarias no serán llevadas a cabo pese a que en el sistema se soliciten al usuario datos bancarios.

2.3. Requisitos específicos

2.3.1. Requisitos de interfaz

La interfaz de la aplicación utilizará la interfaz del navegador Web, ajustándose así al comportamiento de éste y pudiéndose utilizar gracias a esto en cualquier sistema operativo.

Hemos distribuido el espacio en la aplicación de forma clara, con tres partes bien diferenciadas. Cabecera, cuerpo y pie de página, como descubrimos en la Ilustración 3.

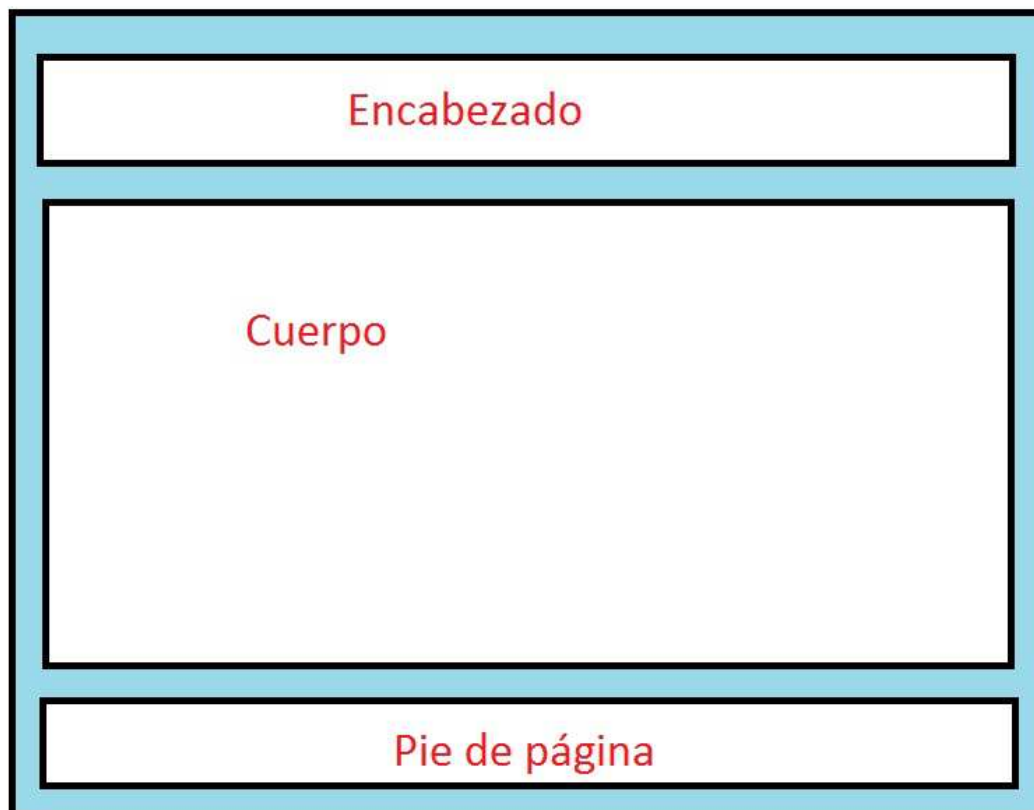


Ilustración 3. Esquema de la página principal

La parte superior contendrá un encabezado en la cual estará alojado el nombre que identifica a la empresa. Además, contendrá una barra de menú horizontal que permitirá el acceso a las distintas secciones de la web. En concreto, éstas serán las siguientes: **Acceder**, que enlazará a la página de inicio de sesión y/o registro de usuario; **Buscar**, que enlaza con las páginas de búsquedas de productos; **Comunidad**, mediante el cual se accede al fórum y **Acerca**, que contiene un formulario de contacto.

Por su parte, la zona centro contiene tres secciones, respetando así el diseño original de la web *Travel*, que veremos más adelante. En la primera de ellas, aparecerá información relacionada con hoteles tales como una fotografía, el nombre, la categoría (en número de estrellas), la ubicación y el precio de pernoctación de ellos.

La segunda sección contiene una tabla de ofertas de hoteles, indicando, para cada uno de ellos, el precio rebajado sobre el actual, y el período de rebajas, entre otros.

La tercera contiene enlaces a cada una de las secciones del fórum.

El pie de página contendrá, en su parte izquierda un submenú a distintas secciones de la web, y el membrete de la empresa en su parte derecha.

Todas las características anteriormente citadas se pueden apreciar de una forma más clara en el Anexo A, incluido en este documento, donde se describen con más detalle cada uno de los componentes de la interfaz. Además, en la Ilustración 3, aparece representado un boceto o un primer diseño de lo que podría ser la interfaz de la aplicación a desarrollar.

2.3.2. Requisitos funcionales

A continuación se describen las diversas funciones y servicios que ofrecerá la página web de la agencia de viajes, algunos de éstos ya citados en apartados anteriores.

Para una mejor comprensión se clasifican dependiendo del tipo de usuario el cual accede al servicio, dependiendo del nivel de accesibilidad que tenga éste, como puede contemplarse en el anexo B.

2.3.3. Requisitos de contenido

La zona principal de la página Web, como se ha mencionado en el apartado anterior, estará destinada a mostrar la información basada en el tipo de contenido seleccionado por el usuario, ya sea información sobre hoteles, vuelos, etc. que se irá generando a partir de los datos introducidos en la base de datos de la aplicación, por lo que todo el contenido de dicha aplicación se generará de forma dinámica. Se ofrecerá información sobre hoteles de diversos lugares, con alguna fotografía representativa de éste, y una descripción, que incluirá datos sobre su situación, servicios, precios, etc. Información sobre vuelos y cruceros, con datos sobre origen, destino, fechas de salida, precios, etc.

Todos los contenidos que aparecerán al realizar las búsquedas tendrán la opción de reserva/alquiler pero solo para usuarios que previamente se hayan registrado en la Web.

Para completar este apartado se puede consultar las tablas que siguen, con características detalladas del tipo de contenido que se ofrecerá en la aplicación de la agencia de viajes.



Apartado	Descripción
Título	Coches
Propósito	Mostrar información acerca de los coches disponibles, así como posibilidad de alquilar y consultar ofertas
Tipo	La información irá cambiando según vayan surgiendo las ofertas. También será susceptible de cambio el precio y la disponibilidad de los mismos
Longitud	Descripción de las características de cada uno de los coches: descripción del coche y precio
Formato	Texto, imágenes
Estilo	Formal

Apartado	Descripción
Título	Cruceros
Propósito	Mostrar información acerca de los cruceros disponibles, así como posibilidad de hacer una reserva y consultar ofertas
Tipo	La información irá cambiando según vayan surgiendo las ofertas. También será susceptible de cambio el precio y la disponibilidad de los mismos
Longitud	Descripción de las características de cada uno de los cruceros: fecha de salida, fecha de llegada, origen, precio.etc
Formato	Texto, imágenes
Estilo	Formal

Apartado	Descripción
Título	Hoteles
Propósito	Mostrar información acerca de los hoteles disponibles, así como posibilidad de hacer una reserva y consultar ofertas
Tipo	La información irá cambiando según vayan surgiendo las ofertas. También será susceptible de cambio el precio y la disponibilidad de los mismos
Longitud	Descripción de las características de cada uno de los hoteles: situación, precios, número de habitaciones.etc
Formato	Texto, imágenes
Estilo	Formal

Apartado	Descripción
Título	Vuelos
Propósito	Mostrar información acerca de los vuelos disponibles, así como posibilidad de hacer una reserva y consultar ofertas
Tipo	La información irá cambiando según vayan surgiendo las ofertas. También será susceptible de cambio el precio y la disponibilidad de los mismos
Longitud	Descripción de las características de cada uno de los vuelos: fecha de salida, fecha de llegada, origen, destino, precio.etc
Formato	Texto, imágenes
Estilo	Formal

3. Análisis

En este capítulo trataremos de establecer una conexión entre el punto anterior, es decir, los requisitos que debe tener la aplicación, y la futura implementación. Se pretende traducir las necesidades en modelos que puedan describir la aplicación a desarrollar, para hacerlo más entendible al futuro implementador del sistema.

Estos modelos nos permitirán comprender mejor la aplicación, especialmente en grandes aplicaciones, lo que nos facilitará la tarea de implementación, así como la ampliación del sistema si fuera necesario. Para ello, nos ayudamos de UML, lenguaje unificado de modelo.

Aunque UML define una gran cantidad de diagramas para representar los distintos aspectos del desarrollo de la aplicación, en nuestro caso nos centraremos en tres tipos de diagrama:

- Diagrama de clases
- Diagrama de casos de usos
- Diagrama de secuencia

Cada uno de estos diagramas pertenece una clase de diagramas (de estructura, de comportamiento y de interacción respectivamente), lo que nos permitirá cubrir los distintos aspectos del desarrollo.

3.1. Diagramas de caso de uso

Los diagramas de casos de uso describen el comportamiento de un sistema desde el punto de vista del usuario. De esta forma nos hacemos una idea de las funciones que un sistema puede ejecutar.

- Actores:

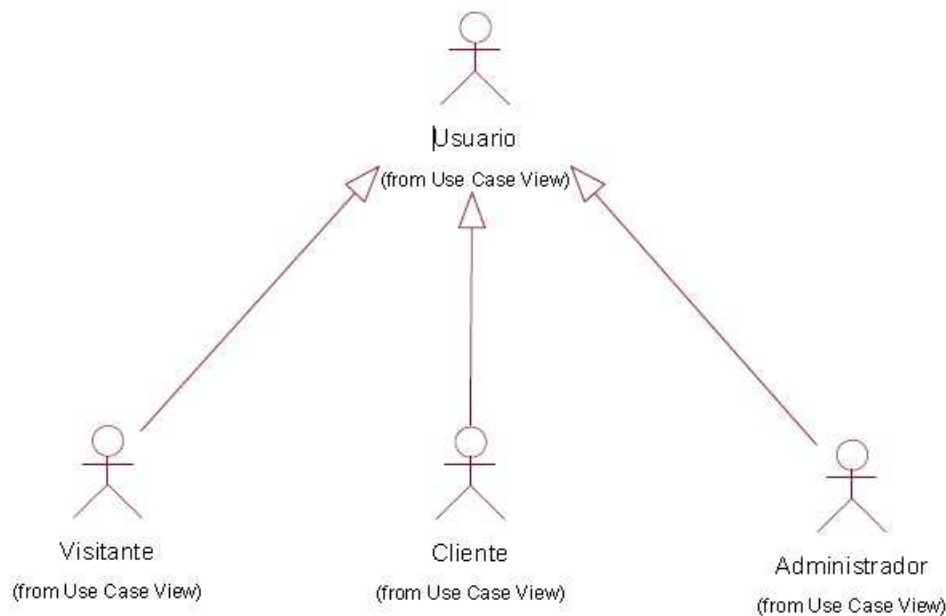


Ilustración 4. Actores del sistema

La Ilustración 4 nos muestra los distintos actores que podrán acceder a la web.

A continuación se describirán las funcionalidades que pueden desarrollar los actores descritos anteriormente.

- Usuario:

La entidad usuario puede realizar funciones de búsqueda dentro de la aplicación, característica común y generalizada de todos los actores en el sistema. Después, cada uno de ellos tendrá unas características que lo diferenciarán del resto en cuanto a funcionalidad se refiere, tal y como se muestra en la Ilustración 5:

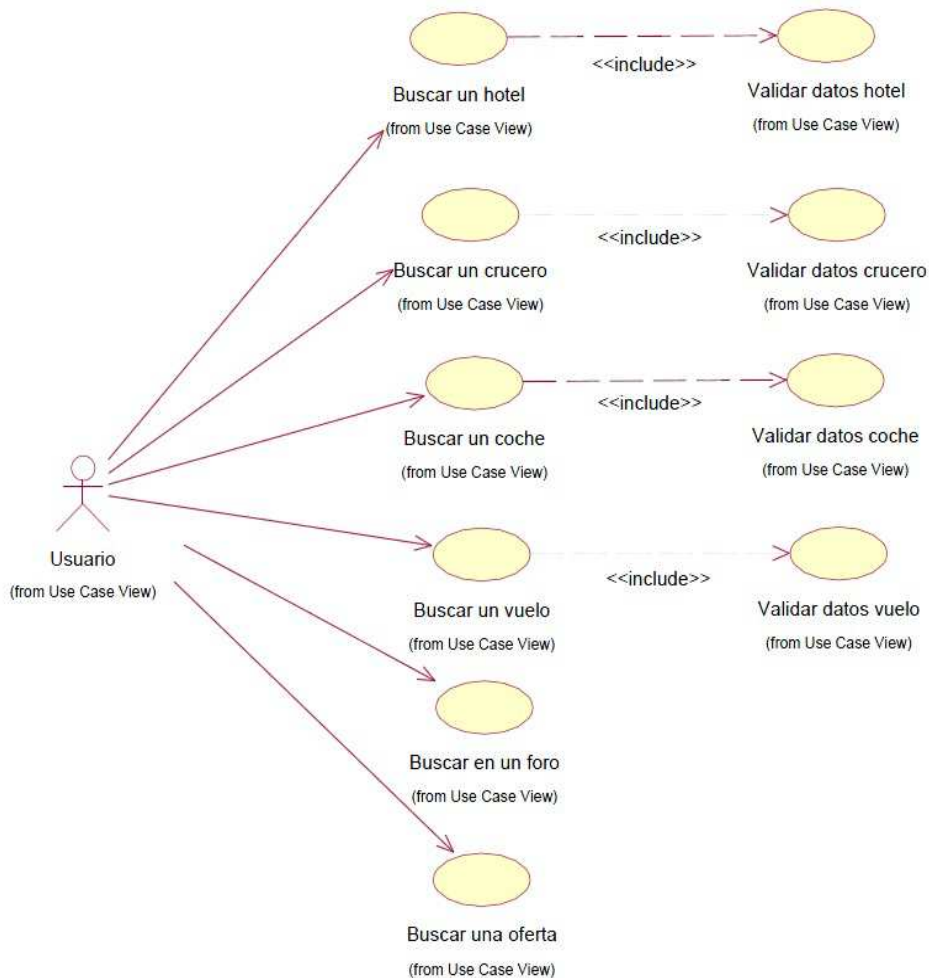


Ilustración 5. Casos de uso del actor usuario

- Visitante:

El actor visitante tiene las mismas funcionalidades que el actor usuario, pero además, debe poder tener la funcionalidad de registrarse en el sistema tal y como se muestra en la Ilustración 6:



Ilustración 6. Caso de uso del actor visitante

- Cliente:

El actor visitante, una vez se ha registrado en el sistema puede ejecutar un inicio de sistema, mediante el cual, tras haberse autenticado con éxito, debe permitirle realizar las siguientes acciones:

- a) Realizar reservas de los distintos productos, tras la búsqueda pertinente.
- b) Dejar comentarios en el fórum.
- c) Gestionar sus propias reservas, cancelando, modificando o simplemente consultas las mismas.

La Ilustración 7 muestra el diagrama correspondiente a los casos de uso del actor usuario registrado:

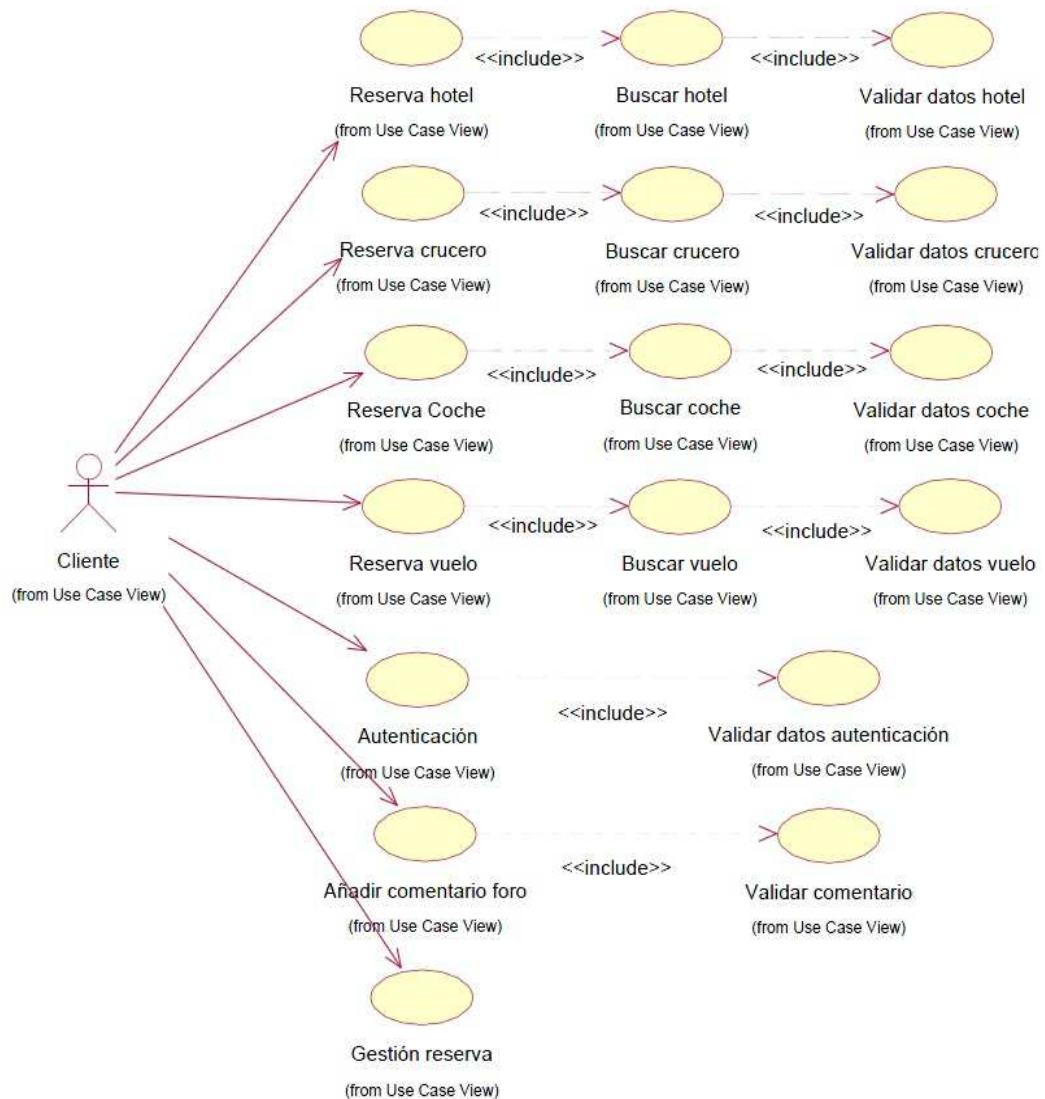


Ilustración 7. Diagrama de casos de uso del actor cliente

3.2. Diagrama de clases

Este tipo de diagrama nos describe la estructura de un sistema mostrando sus entidades, atributos y las relaciones existentes entre ellos. Estos diagramas se utilizan durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la

información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre ellos. Este diagrama nos permitirá conocer a grandes rasgos las clases u objetos de nuestro sistema.

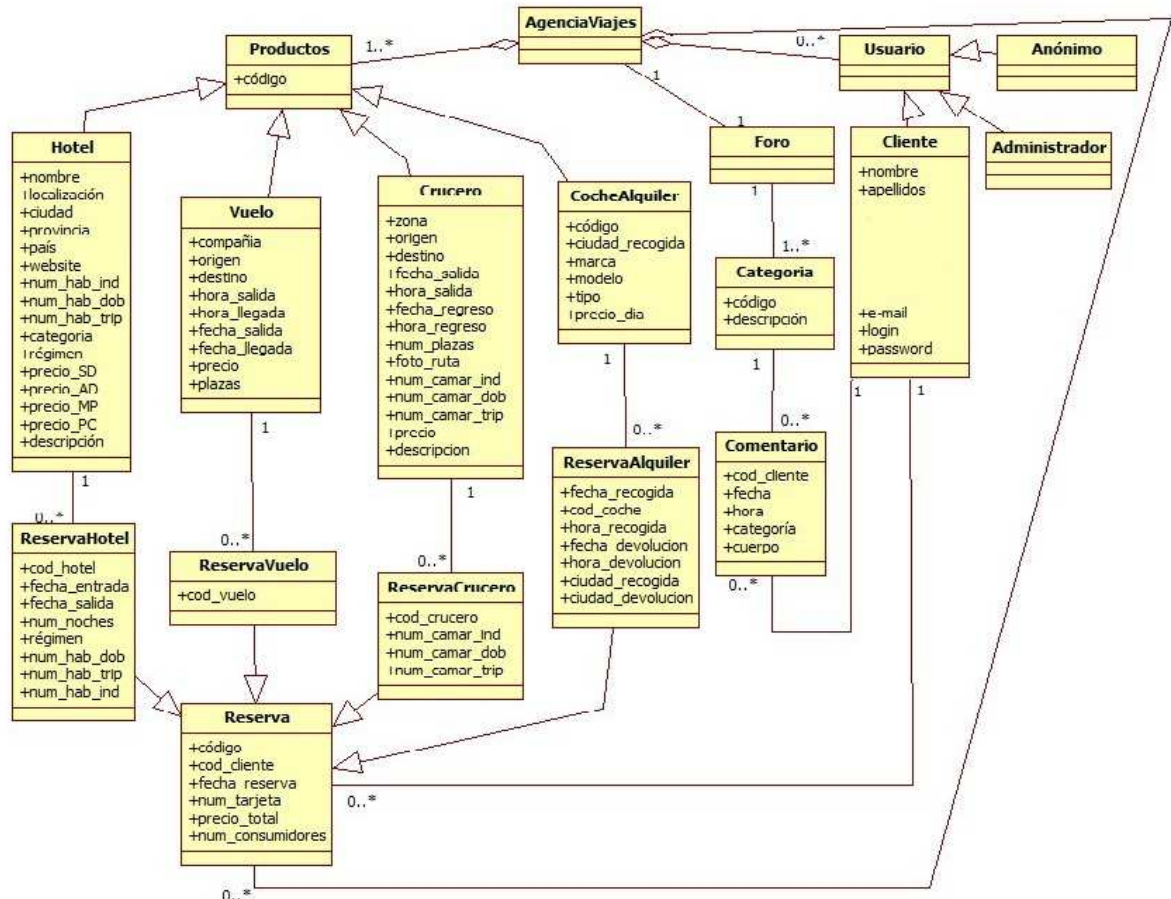


Ilustración 8. Diagrama de clases

- **Hotel:** Esta entidad forma parte de la entidad productos, y nos muestra los distintos destinos a los cuales podemos viajar.
- **Reserva:** Esta clase relacionas a los cliente con los agencia de viajes, estaría constituida por una fechas de entrada y salida, un destino y un usuario registrado.
- **AgenciaViajes:** Sería una de las principales clases de nuestra aplicación ya que sin su existencia no tendría ningún sentido. Sobre

esta clase los usuarios registrados realizan las reservas, u los usuarios no registrados, pueden ver nuestro catálogo para decidir si quieren registrarse para realizar alguna reserva. Sobre esta clase los usuarios registrados pueden realizar comentarios.

3.3. Diagramas de secuencia

A continuación se describen escenarios y diagramas de secuencia para los casos de uso indicados en el apartado anterior, mostrando la interacción que tendrá cada usuario con el sistema desarrollado.

Los escenarios dan información sobre los pasos a seguir para llevar a cabo las acciones descritas en los casos de uso. Se describirán los escenarios y los diagramas de secuencia asociados a los casos de uso las búsquedas de productos.

Escenarios productos

- **Búsquedas de productos:**
 1. Introducir datos producto
 2. Crear producto
 - 2.1. Buscar los datos para el producto
 - 2.2. Devolver producto
 - 2.3. Añadir el producto a la lista
 3. Devolver lista productos

- **Reservas de productos:**

1. Introducir datos producto
2. Crear producto
 - 2.1. Buscar los datos para el producto
 - 2.2. Devolver producto
 - 2.3. Añadir el producto a la lista
3. Devolver lista productos
4. Seleccionar hotel
5. Introducir datos reserva hotel
6. Insertar registro de la reserva de hotel en la base de datos
 - 6.1 Confirmar reserva

- **Identificación de cliente:**

1. Introducir datos usuario registrado
2. Comprobar datos cliente
3. Devolver datos correctos
4. Devolver login correcto

- **Búsqueda de productos:**

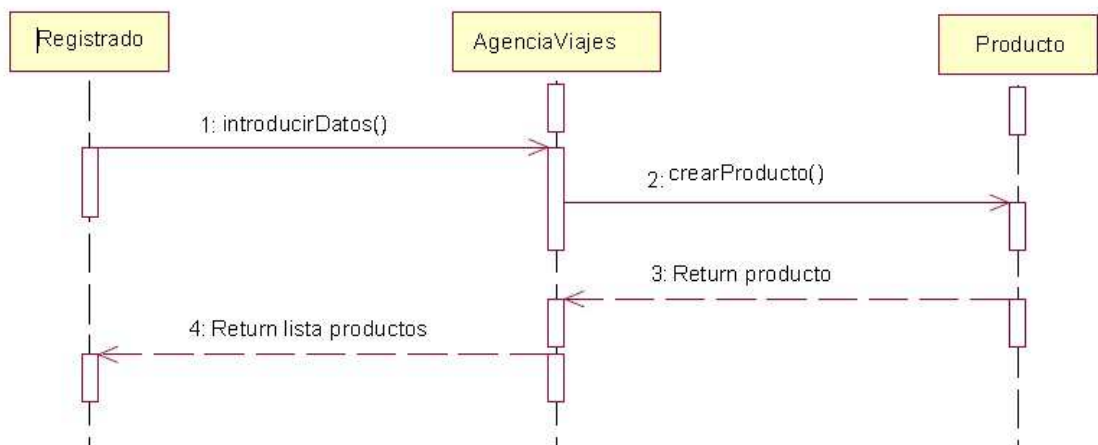


Ilustración 9. Diagrama de secuencia 1

- **Reserva de productos:**

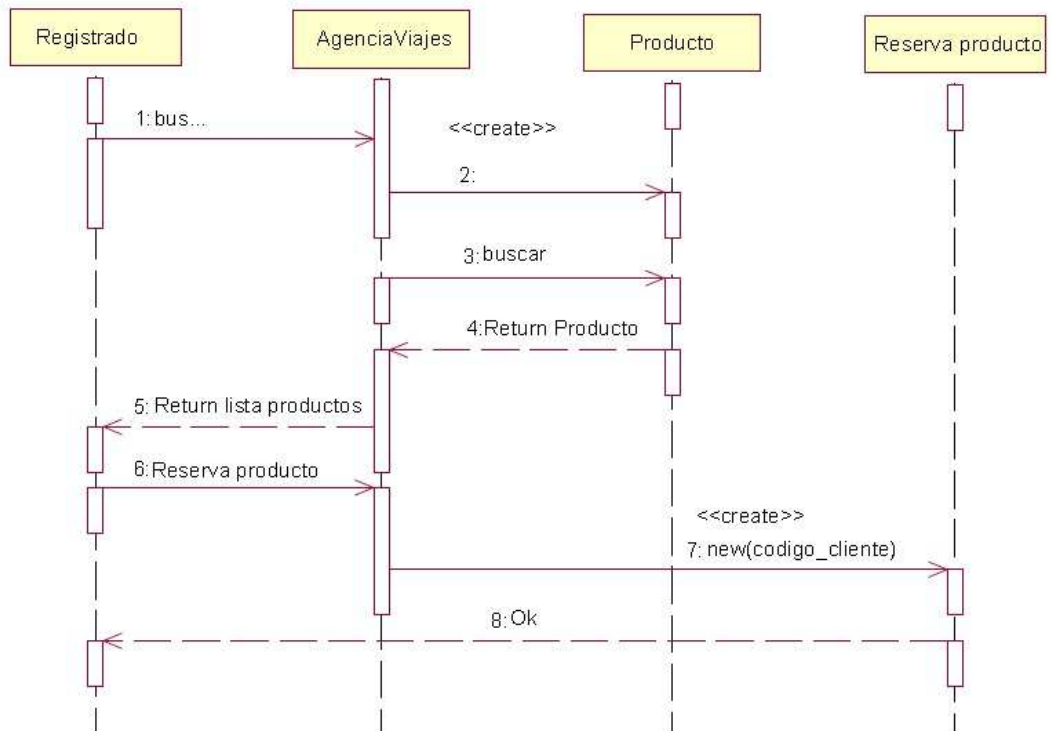


Ilustración 10. Diagrama de secuencia 2

- **Login usuario:**

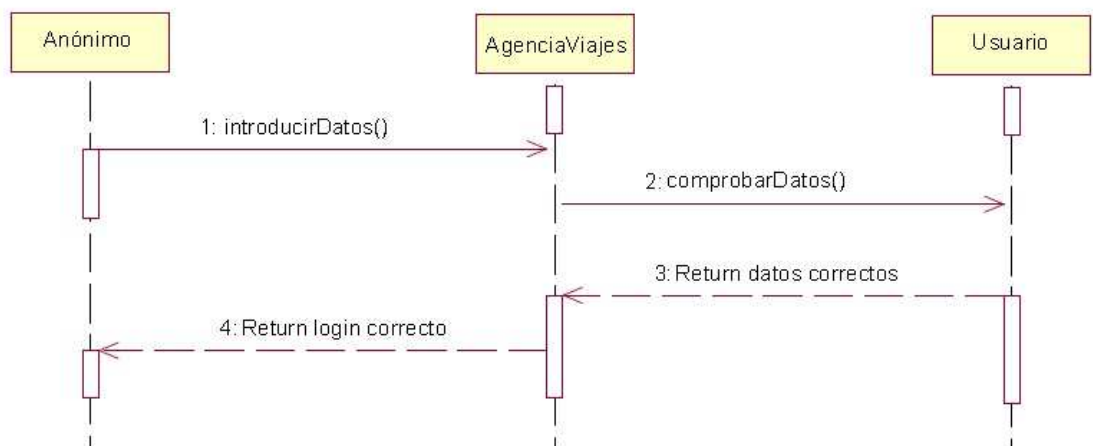


Ilustración 11. Diagrama de secuencia 3



En el anexo C, se encuentran los diagramas de secuencia para cada una de las acciones que se pueden llevar a cabo en el sistema de forma más detallada.

4. Diseño

4.1. Introducción

Tradicionalmente, se han creado aplicaciones compactas, sin ninguna capa separada. Por esta razón, surgían múltiples problemas de integración en sistemas complejos. Además, brotaban inconvenientes en cuanto a la escalabilidad, la disponibilidad y la seguridad del sistema.

Para el desarrollo del sitio web Tournalia hemos optado por un sistema de programación por capas. Este modelo propone la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

4.2. Arquitectura del sistema

La arquitectura tres capas propone separar el sistema en varios niveles. Por una parte, existe la capa que se utilizará para guardar la información, es decir, la base de datos. Seguidamente, la lógica de negocio y, por último, la tercera capa, la interfaz gráfica que ayude al usuario final a usar el sistema. En la Ilustración 12, se puede visualizar gráficamente un ejemplo.

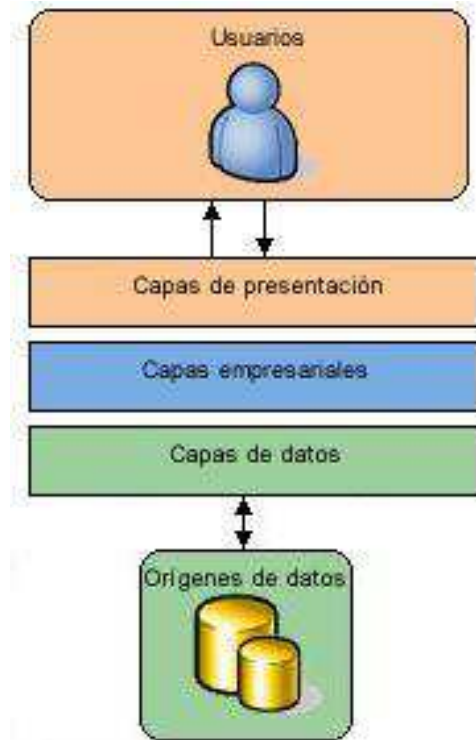


Ilustración 12. Arquitectura de tres capas

Como ventajas de esta arquitectura destacaremos lo siguiente:

- Las llamadas de la interfaz del usuario en la estación de trabajo, al servidor de capa intermedia, son más flexibles que en el diseño de dos capas, ya que la estación solo necesita transferir parámetros a la capa intermedia.
- Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en el PC.
- El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular.
- La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.



4.2.1. Capa de presentación

Esta capa se encarga de proveer una interfaz entre el sistema y el usuario. Básicamente, se responsabiliza de que se le comunique información al usuario por parte del sistema y viceversa, manteniendo una comunicación exclusiva con la capa de negocio que veremos a continuación. Además dentro de esta capa entraría aquello que el usuario “ve” cuando se conecta a la aplicación.

El apartado en cuestión va a describir la capa de presentación de la aplicación web de la agencia de viajes Touralia. Veremos como la aplicación dispone de una interfaz para el usuario, funcional y distinguida, que va a permitir a los clientes interactuar con el sistema de forma fácil y precisa.

El diseño elegido, está basado en el diseño de la web Travel de ThemeForests Website Templates, consta de cuatro partes diferenciadas, que serán, un banner publicitario, un encabezado, un pie de página y la zona central. Con ello, conseguimos mostrar de forma clara y organizada cada una de las secciones de las que se compone la aplicación Web y ayuda a poder estructurarla de una forma más sencilla.

En el anexo se puede observar cierto detalle de su implementación.

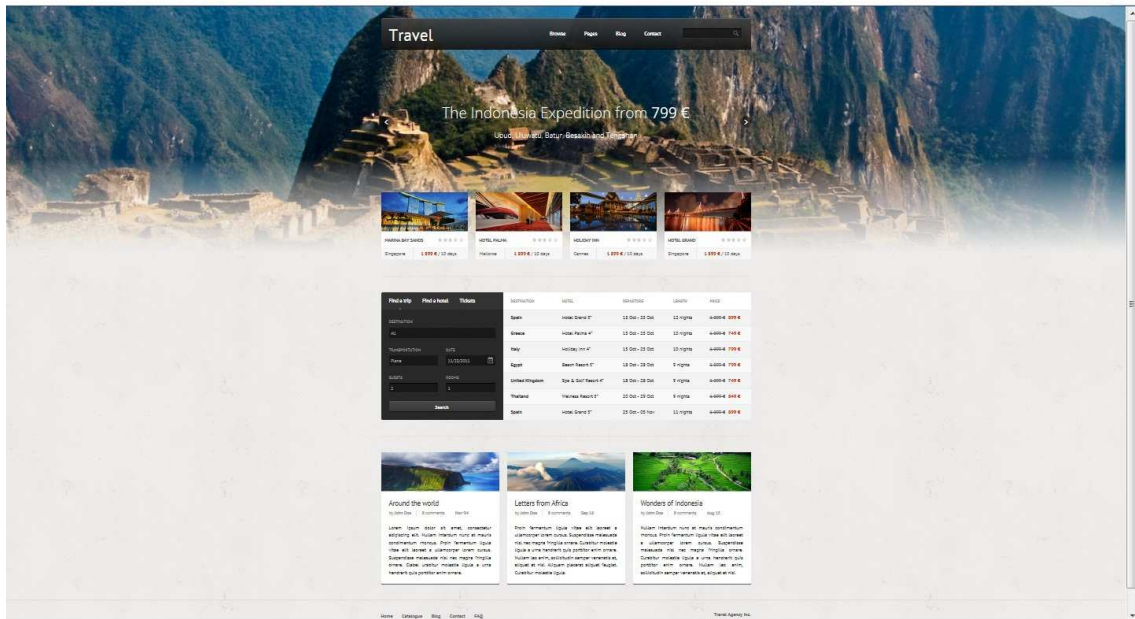


Ilustración 13. Ejemplo de la web usada como referencia

A continuación, hemos descrito cada una de las partes que componen esta página

Detalladamente la web contiene los siguientes elementos:

1. La barra de menú de navegación que contiene el nombre la empresa con el contenido explicitado en la sección de requisitos funcionales.



Ilustración 14. Barra de navegación

2. Dentro del cuerpo hay cuatro zonas:

2.1. Banner publicitario



Ilustración 15. Banner publicitario

2.2. Información ofertas

DESTINO	HOTEL	FECHAS	DÍAS	PRECIO
Amsterdam	Amstel InterContinental Amsterdam 5*	14 Mar - 21 Mar	1	100 80 €
Las Vegas	Flamingo 4*	14 Mar - 21 Mar	1	49 39.2 €
Mallorca	Puro Hotel Palma 4*	14 Mar - 21 Mar	1	130 104 €
Estocolmo	Wallin Hotel 4*	14 Mar - 21 Mar	1	274 219.2 €
Florianopolis	Don Zepe Hotel 3*	14 Mar - 21 Mar	1	85 68 €
Las Vegas	Stratosphere Casino & Hotel 3*	14 Mar - 21 Mar	1	41 32.8 €
Las Vegas	Riviera Hotel and Casino 3*	14 Mar - 21 Mar	1	45 36 €

Ilustración 16. Panel de información de las ofertas

2.3. Temáticas del fórum



Ilustración 17. Temáticas del fórum

2.4. Pie de página



Ilustración 18. Pie de página

4.2.2. Capa de negocio

Es la capa que contiene los procesos a realizar con la información recibida desde la capa de presentación, las peticiones que el usuario ha realizado, y responsabilizándose de que se le envíen las respuestas adecuadas a la capa de presentación.

Podríamos verla como una capa intermedia, a medio camino entre la capa de presentación y la capa de datos, puesto que se relaciona con ambas y por supuesto, procesa también la información devuelta por la capa de datos.

La capa de negocio contendrá un componente con funciones de acceso a la base de datos que permitirá:

- Realizar la conexión con la base de datos.
- Ejecutar sentencias en SQL, como consultas, inserciones, actualizaciones y borrados.
- Obtener el número de registros de una tabla.
- Otras funcionalidades relacionadas con la base de datos.

En la Ilustración 20, se observa en la capa de negocio tres grandes componentes. Por un lado, el analizador del login, que comprueba en la base de datos, si el usuario/administrador es quien dice ser. En caso afirmativo, se redirige a las funciones que tiene disponibles; en caso contrario, no se podrá acceder al sistema.

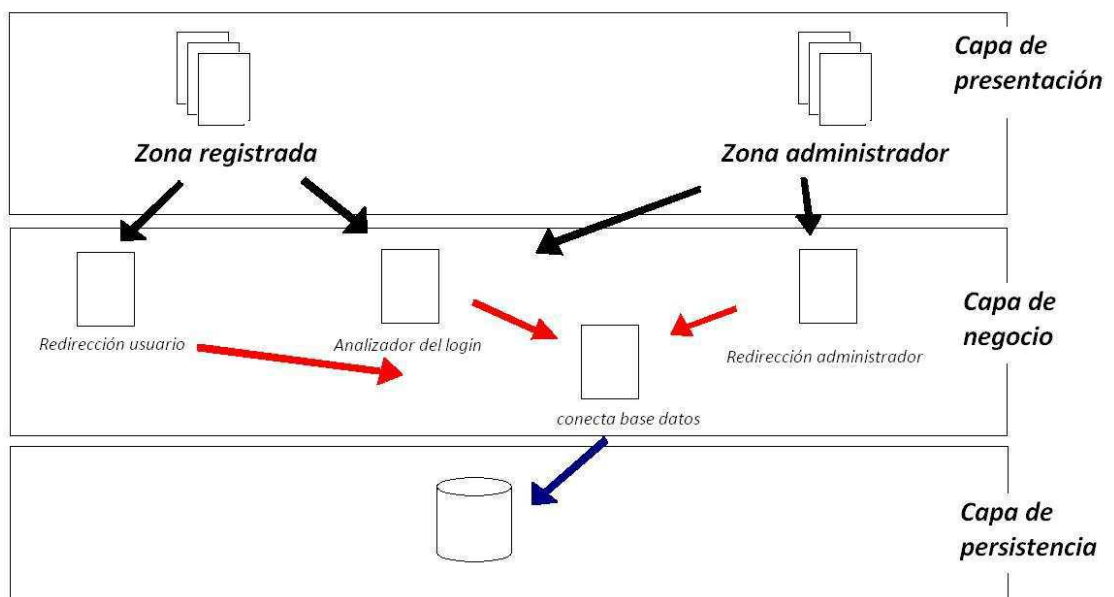


Ilustración 19. Capa de negocio

4.2.3. Capa de bases de datos

Por último, la capa donde se almacenan los datos. Mediante la capa de negocio, se puede encargar de ofrecer, modificar, almacenar, borrar y recuperar datos, mediante el gestor (o los gestores) de bases de datos que la aplicación requiera.

El modelo de datos entidad-relación describe de una manera visual y gráfica la capa de datos de nuestra aplicación. Se puede observar la estructura de nuestra base de datos, el lugar donde se almacena cualquier dato del sistema. Se pueden diversificar varios componentes, como son las entidades, las relaciones y los atributos.

Las entidades son cualquier tipo de objeto sobre el que se puede recoger información importante y útil para el sistema. Gráficamente, se representan con rectángulos y el nombre es mostrado en el interior.

Las relaciones son asociaciones entre entidades. Cada una de las relaciones, tienen un nombre que describe su función y se representan mediante un rombo.

Finalmente, los atributos son una característica que contiene información sobre una entidad o una relación. Representa la parte más básica del modelo.

La Ilustración 20, muestra una parte del modelo entidad-relación. En ella, destacaríamos las entidades, por una parte, **hotel**, **crucero**, **vuelo**, **coche** que representan los productos que ofrece el portal web. Dichas entidades se relacionan, por un lado, con las entidades **reservahotel**, **reservacrucero**, **reservavuelo**, **reservacoche** las cuales, representan cada una de las peculiaridades de cada reserva de cada producto y que se relacionan con la entidad **reserva**. Por otra parte, las entidades que representan los productos se relacionan con las entidades **hotelimagen**, **cruceroimagen**, **vueloimagen**, **cocheimagen** las cuales enlazan la entidad **imagen** con su producto asociado.

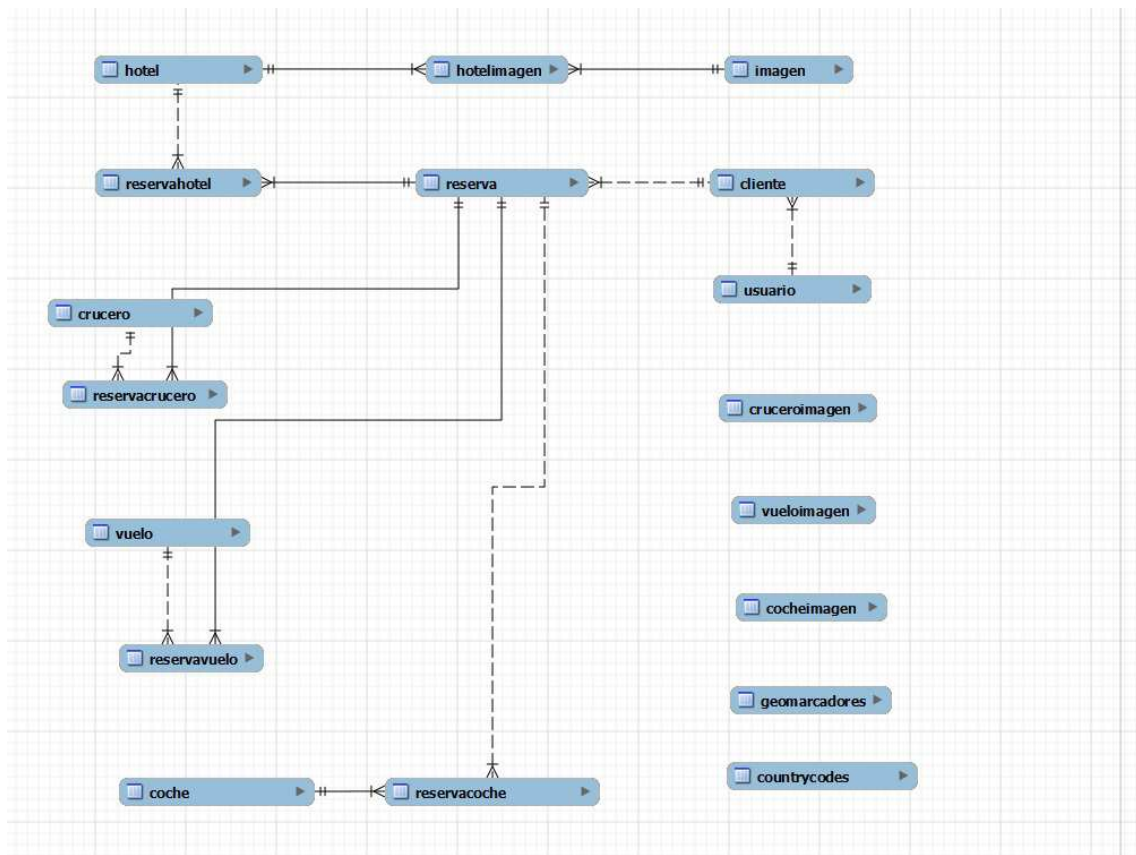


Ilustración 20. Diagrama entidad-relación

A continuación, se muestra el esquema lógico, detallado en datos:

hotel(idHotel:dom_num, nombre:dom_nom, localización:dom_nom, ciudad:dom_nom, provincia:dom_nom, pais:dom_nom, website:dom_nom, num_hab_ind:dom_num, num_hab_dob:dom_num, num_hab_trip:dom_num, categoria:dom_num, regimen:dom_nom, precio_SD:dom_decimal, precio_AD:dom_decimal, precio_MP:dom_decimal, precio_PC:dom_decimal, descripcion:dom_desc)

CP: {idHotel}

VNN: {nombre, localización, ciudad, provincia, pais, num_hab_ind, num_hab_dob, num_hab_trip, categoria, regimen, precio_SD, precio_AD, precio_MP, precio_PC}

reservahotel(fechaEntrada:dom_fec, fechaSalida:dom_fec, regimen:dom_cadena, num_hab_ind:dom_num, num_hab_dob:dom_num, num_hab_trip:dom_num, codigo:dom_num, idHotel:dom_num)

CP: {idHotel, codigo}

CAj: {idHotel} → **hotel**

CAj: {codigo} → **reserva**

VNN: {fechaEntrada, fechaSalida, régimen, num_hab_ind, num_hab_dob, num_hab_trip, código, idHotel}

reserva(codigo:dom_num, fechaReserva:dom_fec, num_tarjeta:dom_num, precioTotal:dom_decimal, num_consumidores:dom_num, idCliente:dom_num)

CP: {codigo}

CAj: {idCliente} → **cliente**

VNN: {fechaReserva, num_tarjeta, precioTotal, num_consumidores, idCliente}

crucero(idCrucero:dom_num, compañía:dom_nom, barco:dom_nom, nombre:dom_nom, zona:dom_nom, origen:dom_nom, destino:dom_nom, fecha_salida:dom_fec, fecha_llegada:dom_fec, website:dom_nom, num_hab_dob:dom_num, num_hab_trip:dom_num, num_hab_cuat:dom_num, categoria:dom_num, regimen:dom_nom, precio_HD:dom_decimal, precio_HT:dom_decimal, precio_HC:dom_decimal, descripcion:dom_desc)

CP: {idCrucero}

VNN: {compañía, barco, nombre, zona, origen, destino, fecha_salida, fecha_llegada, num_hab_dob, num_hab_trip, num_hab_cuat, categoria, regimen, precio_HD, precio_HT, precio_HC}

reservacrucero(fechaEntrada:dom_fec, fechaSalida:dom_fec, regimen:dom_cadena, num_hab_dob:dom_num, num_hab_trip:dom_num, num_hab_cuat:dom_num, codigo:dom_num, idCrucero:dom_num)

CP: {idCrucero, codigo}

CAj: {idCrucero} → **crucero**

CAj: {codigo} → **reserva**

VNN:

{fechaEntrada, fechaSalida, régimen, num_hab_dob, num_hab_trip, num_hab_cuat}

coche(idCoche:dom_num, ciudad_recogida:dom_nom, marca:dom_nom, modelo:dom_nom, tipo:dom_nom, precio:dom_decimal, descripcion:dom_desc)

CP: {idCoche}

VNN: {idCoche, ciudad_recogida, marca, modelo, tipo, precio}



reservacoche(codigo:dom_num,idCoche:dom_num,ciudad_recogida:dom_nom,ciudad_devolucion:dom_nom,fecha_hora_recogida:dom_fec,fecha_hora_devolucion:dom_fec)

CP: {idCoche,codigo}

CAj: {idCoche} → **coche**

CAj: {codigo} → **reserva**

VNN:

{ciudad_recogida,ciudad_devolucion,fecha_hora_recogida,fecha_hora_devolucion}

vuelo(idVuelo:dom_num,companyía:dom_nom,origen:dom_nom, destino:dom_nom, fecha_salida:dom_fec,fecha_llegada:dom_fec,hora_salida:dom_hor,hora_llegada:dom_fec,num_asientos:dom_num,website:dom_nom,tipo:dom_nom, precio:dom_decimal, descripcion:dom_desc)

CP: {idVuelo}

VNN: {companyía, origen,destino,fecha_salida,fecha_llegada,hora_salida,hora_llegada, num_asientos, tipo,precio}

oferta(idOferta:dom_num,rebaja:dom_decimal, fecha_inicio:dom_fec, fecha_fin:dom_fec, descripcion:dom_desc)

CP: {idOferta}

VNN: {rebaja, fecha_inicio,fecha_fin}

ofertahotel(idOferta:dom_num,idHotel:dom_num, noches:dom_num, descripcion:dom_desc)

CP: {idOferta,idHotel}

CAj: {idOferta} → **oferta**

CAj: {idHotel} → **hotel**

VNN: {noches}

imagen(idImagen:dom_num, src:dom_num,width:dom_num,height:dom_num, alt:dom_num)

CP: {idImagen}

VNN: {src,width,height,alt}

imagenhotel(idHotel:dom_num,idImagen:dom_num)

CP: {idHotel,idImagen}

CAj: {idImagen} → **imagen**

CAj: {idHotel} → **hotel**



imagen crucero(idCrucero:dom_num,idImagen:dom_num)

CP: {idCrucero,idImagen}

CAj: {idImagen} → **imagen**

CAj: {idCrucero} → **crucero**

imagen coche(idCoche:dom_num,idImagen:dom_num)

CP: {idCoche,idImagen}

CAj: {idImagen} → **imagen**

CAj: {idCoche} → **coche**

imagen vuelo(idVuelo:dom_num,idImagen:dom_num)

CP: {idVuelo,idImagen}

CAj: {idImagen} → **imagen**

CAj: {idVuelo} → **vuelo**

usuario(idUsuario:dom_num,login:dom_nom,
password:dom_nom,email:dom_nom,nombre:dom_nom,apellidos:dom_nom)

CP: {idUsuario}

VNN: {login,password,email,nombre,apellidos}

geomarcadores(idGeoMarcadores:dom_num,
nombre:dom_nom,direccion:dom_nom,lat:dom_float,lng:dom_float,tipo:dom_nom)

CP: {idGeoMarcadores}

VNN: {nombre,dirección,lat,lng,tipo }



Definición de los dominios:

Nombre	Dominio
Dom_nom	Varchar(45)
Dom_num	Int(11)
Dom_fec	date
Dom_hor	time
Dom_dec	Decimal(10,0)
Dom_float	float



5. Implementación

5.1. Introducción

A continuación, vamos a repasar cuáles han sido las distintas tecnologías y aplicaciones utilizadas para la confección de la página web Touralia, así como su estructura y contenido.

5.2. Tecnologías

La apariencia de nuestro sitio web es sencilla. Las herramientas utilizadas para la parte estática son el lenguaje HTML, así como los CSS, necesarios para mantener la misma apariencia en todas las páginas. En esta sección también se han utilizado las tecnologías de JavaScript y jQuery.

Para la programación hemos usado otro lenguaje de programación PHP acompañado de la base de datos MySQL. Son tecnologías de libre distribución, por lo tanto, encontrar documentación sobre ellas, es fácil.

Seguidamente vamos a describir las herramientas utilizadas en cada una de las capas para la programación web:

5.2.1. Capa de presentación

HTML

El HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). Fue creado en 1986 por el físico nuclear Tim Berners-Lee; el cual tomo dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) el cual permite conectar dos elementos entre si y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizara en la forma como éste lo entienda.

El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html

Estos documentos pueden ser mostrados por los visores o "browsers" de páginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer.

También existe el HTML Dinámico (DHTML), que es una mejora de Microsoft de la versión 4.0 de HTML que le permite crear efectos especiales como, por ejemplo, texto que vuela desde la página palabra por palabra o efectos de transición al estilo de anuncio publicitario giratorio entre página y página.

CSS

El nombre hojas de estilo en cascada viene del inglés *Cascading Style Sheets*, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y *por extensión en XHTML*). El W3C (*World Wide Web Consortium*) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la *estructura* de un documento de su *presentación*.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "<style>".

JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipificado y dinámico.

Se utiliza principalmente en su forma del *lado del cliente (client-side)*, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador...⁴ aunque existe una forma de JavaScript del lado del servidor (*Server-side JavaScript o SSJS*). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (*mayoritariamente widgets*) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Jquery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, 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 páginas web. jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, 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.

Como características presenta:

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un *plugin* básico de XPath.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.

- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones como trim() (elimina los espacios en blanco del principio y final de una cadena de caracteres), etc.
- Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.

5.2.2. Capa Lógica o de negocio

PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. Podemos saber algo más sobre la programación del servidor y del cliente en el artículo qué es DHTML.

5.2.3. Capa de datos

MySql

MySQL es la base de datos *open source* más popular y, posiblemente, mejor del mundo. Su continuo desarrollo y su creciente popularidad está haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle.

MySQL es un sistema de administración de bases de datos (*Database Management System, DBMS*) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información.

MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

También es muy destacable, la condición de *open source* de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.



5.3. Herramientas

En esta sección se recogerán las distintas herramientas usadas para confeccionar la página web.

5.3.1. Adobe Dreamweaver 8

Es una aplicación de pago destinada a la construcción y edición de sitios y aplicaciones Web. Compatible con los sistemas operativos Windows y Mac. Entre sus características destacan la creación de páginas Web sin escribir código HTML(WYSIWYG), la administración de sitios, agrupando los archivos según el proyecto al que pertenezcan, la inclusión de un cliente FTP, que permite subir los archivos editados inmediatamente al sitio en Internet y la funcionalidad de autocompletado y resaltado de la sintaxis de código HTML, PHP, ASP, etc.

```
1 <?php
2
3 require_once 'forms/Plantilla.php';
4 require_once 'forms/UIUtils.php';
5 require_once 'sistema/AgenciaViajes.php';
6 require_once 'sistema/Comunes.php';
7
8 class BuscadorHoteles extends Plantilla {
9
10     const per_page = 3;
11
12     function __construct($page, $sec_session) {
13         parent::__construct($page, $sec_session);
14     }
15
16     function Load() {
17         parent::Load();
18         $agencia = new AgenciaViajes();
19         if (isset($_GET['Buscar'])) { // Formulario de búsqueda de hoteles
20             $_SESSION['buscar_hotel_destino'] =
21                 (!isset($_GET['destino']) || empty($_GET['destino']))
22                 || strtoupper($_GET['destino']) == Comunes::TODOS
23                 ? Comunes::TODOS : $_GET['destino'];
24             $_SESSION['buscar_hotel_entrada'] = (isset($_GET['entrada']))
25                 ? UIUtils::parseDateToISO8601($_GET['entrada']) : UIUtils::nowDateEs();
26             $_SESSION['buscar_hotel_salida'] = (isset($_GET['salida']))
27                 ? UIUtils::parseDateToISO8601($_GET['salida']) : UIUtils::nowDateEs();
28             if (strtotime($_SESSION['buscar_hotel_entrada']) > strtotime($_SESSION['buscar_hotel_salida'])) {
29                 $_SESSION['buscar_hotel_salida'] = $_SESSION['buscar_hotel_entrada'];
30             }
31             $_SESSION['buscar_hotel_huespedes'] =
32                 (isset($_GET['huespedes']) && is_numeric($_GET['huespedes']) && intval($_GET['huespedes']) > 0)
33                 ? intval($_GET['huespedes']) : 2;
34             $_SESSION['buscar_hotel_habitaciones'] =
35                 (isset($_GET['habitaciones']) && is_numeric($_GET['habitaciones']) && intval($_GET['habitaciones']) > 0)
36                 ? intval($_GET['habitaciones']) : 1;
```

Ilustración 21. Adobe Dreamweaver 8

5.3.2. WAMP Server

WampServer de Windows es un entorno de desarrollo Web. Permite crear aplicaciones web y ejecutarlas con Apache, PHP y la base de datos MySQL. También se puede utilizar SQLite Manager, PHPMYAdmin para administrar más fácilmente las bases de datos.

WampServer se instala automáticamente (mediante un instalador), y su uso es muy intuitivo. Permite afinar un servidor sin tan siquiera tocar los archivos de configuración.

WampServer es la única solución que permitirá tener servidor de propio producción. Una vez WampServer está instalado, permite la

posibilidad de añadir el mayor número de Apache, MySQL y PHP como las emisiones que se deseen.

Funcionalidades de WampServer:

WampServer tiene funcionalidades que lo hacen muy completo y fácil de usar. Con un click con el botón izquierdo del ratón sobre el icono de WampServer, se puede:

- Gestionar sus servicios de Apache y MySQL,
- Cambiar de línea / fuera de línea (dar acceso a todos o sólo local)
- Instalar y cambiar de Apache, MySQL y PHP emisiones
- Gestión de la configuración de sus servidores
- Acceder a sus registros
- Acceder a sus archivos de configuración
- Crear alias

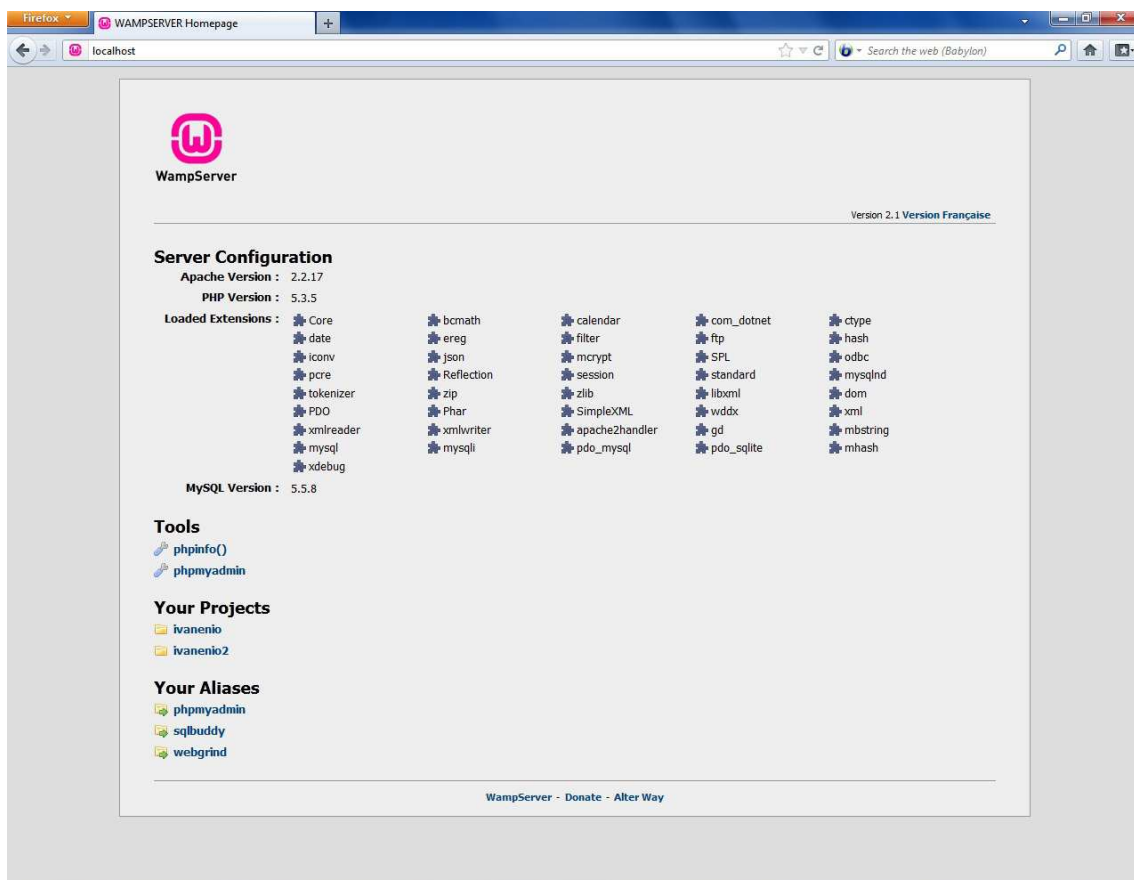


Ilustración 22. Ventana principal de Wamp Server



5.3.3. Apache

Es un servidor HTTP de código abierto diseñado para plataformas Microsoft Windows, Macintosh y Unix. Su nombre es debido a que su creador Brian Behelendorf quiso que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU.

El proyecto Apache es desarrollado dentro del proyecto HTTP Server de la Apache Software Foundation. Presenta características configurables, bases de datos de autenticación y negociado por contenido. Sin embargo, fue muy criticado por la falta de una interfaz gráfica que ayudase a su configuración. Es el servidor web más utilizado en todo el mundo.

Apache se usa mayoritariamente para el envío de páginas web estáticas y dinámicas de Internet. La mayoría de aplicaciones web están diseñadas asumiendo Apache como servidor, o al menos, un servidor que tenga las características propias de este servidor. El funcionamiento es sencillo, cuando el servidor recibe una petición de página con extensión *.php*, llama al intérprete de PHP. Este se encarga del procesamiento y la interpretación del código almacenado en la página solicitada y una vez procesado le devuelve el resultado al servidor Apache. Con toda la información ya procesada el servidor Apache solo tiene que servir el código HTML facilitado por el intérprete a los usuarios finales.

5.3.4. GIMP 2

El The Gimp 2.0 es un programa edición de imágenes con código abierto y gratuito, muchos de los cambios que ha experimentado este software los han desarrollado los propios usuarios.

Consta de una interfaz recientemente revisada con una serie de herramientas básicas, actualmente ya aparece la herramienta de texto manejando caracteres que combina con total sencillez la anterior herramienta de texto con el *plugin* de texto dinámico. La herramienta de trazado está provista de una interfaz nueva pudiéndose crear trazados abiertos.

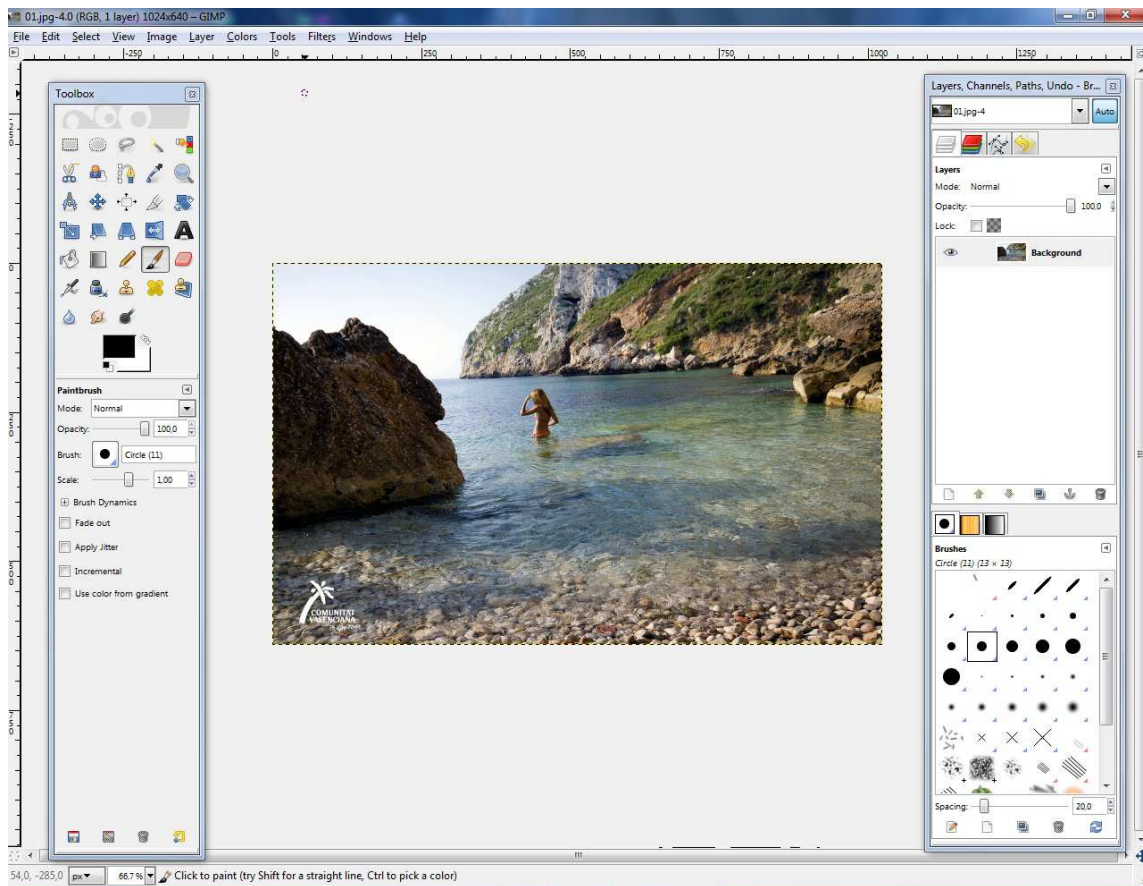


Ilustración 23. Gimp 2



5.3.5. Notepad ++

Notepad++ es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Solo funciona en Microsoft Windows.

Se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple. No obstante, incluye opciones más avanzadas que pueden ser útiles para usuarios avanzados como desarrolladores y programadores.

Se distribuye bajo los términos de la Licencia Pública General de GNU.

```
1 <!doctype html>
2 <!--[if lt IE 7]> <html class="no-js ie6 oldie" lang="es"> <![endif]-->
3 <!--[if IE 7]> <html class="no-js ie7 oldie" lang="es"> <![endif]-->
4 <!--[if IE 8]> <html class="no-js ie8 oldie" lang="es"> <![endif]-->
5 <!--[if gt IE 8]><!-- <html class="no-js" lang="es"> <!--<![endif]-->
6
7 <head>
8     <meta charset="utf-8">
9
10    <title>{&title}</title>
11    <meta name="description" content="{&description}">
12    <meta name="author" content="{&author}">
13
14    <!-- Load CSS -->
15    <link rel="stylesheet" href="css/style.css">
16    <link rel="stylesheet" href="fancybox/jquery.fancybox-1.3.4.css">
17    <link rel="stylesheet" href="css/smoothness/jquery-ui-({&jquery_ui_version}).custom.css">
18    <link rel="stylesheet" href="css/search.css">
19
20    <!-- Page icon -->
21    <link rel="shortcut icon" href="favicon.png">
22
23    <!-- Load Modernizr -->
24    <script src="js/libs/modernizr-2.0.min.js"></script>
25
26    <!-- Load JavaScript -->
27    <script src="http://ajax.googleapis.com/ajax/libs/jquery/{&jquery_version}/jquery.min.js"></script>
28    <script>window.jQuery || document.write('<script src="js/libs/jquery-{&jquery_version}.min.js"></script>')</script>
29    <script src="js/libs/jquery-ui-({&jquery_ui_version}).custom.min.js"></script>
30    <script src="js/libs/jquery-ui.datepicker-es.js"></script>
31    <script src="js/script.js"></script>
32    <script src="fancybox/jquery.fancybox-1.3.4.pack.js"></script>
33    <script src="http://maps.googleapis.com/maps/api/js?sensor=false"></script>
34    <script type="text/javascript">
35        current_page = "1";
36        $(function() {
37            var availableCiudades = [{loop:"available_ciudades_list"}
38                {$value.ciudad} {$value.separator} {else}
39                {/loop}
40            ];
41            $('#destino').autocomplete({ source: availableCiudades });
42        });
43        $(document).ready(function() {
44            function showLoader() {
45                $('&search-background').fadeIn(200);
46            }
47            function hideLoader() {
48                $('&search-background').fadeOut(200);
49            }
50        });
```

Ilustración 24. Notepad ++

5.4. Detalles implementación

En este apartado vamos a describir los pasos seguidos para llevar a cabo la implementación completa de nuestro sitio web Touralia. Seguidamente se detallarán el modo en que han sido implementadas las principales funcionalidades del sitio.

5.4.1. Usuario visitante

Para acceder a la página de inicio de la agencia de viajes se accederá directamente al dominio o dirección IP del servidor donde esté alojada. Se ha modificado la idea de diseño inicial, ya que la página Web quedará ahora centrada en la pantalla del usuario, con un menor ancho, lo que permitirá ser compatible con resoluciones de hasta 800x600, siendo compatible con un mayor número de dispositivos móviles, ultraportátiles, etc. muy utilizados hoy en día.

Como visitante, cualquier usuario debe tener la posibilidad de buscar entre los diferentes productos y consultar información referente a cada uno de los detalles de los mismos. Así por ejemplo, debe poder acceder a la página de búsquedas de hoteles, tal y como se muestra en la siguiente ilustración:

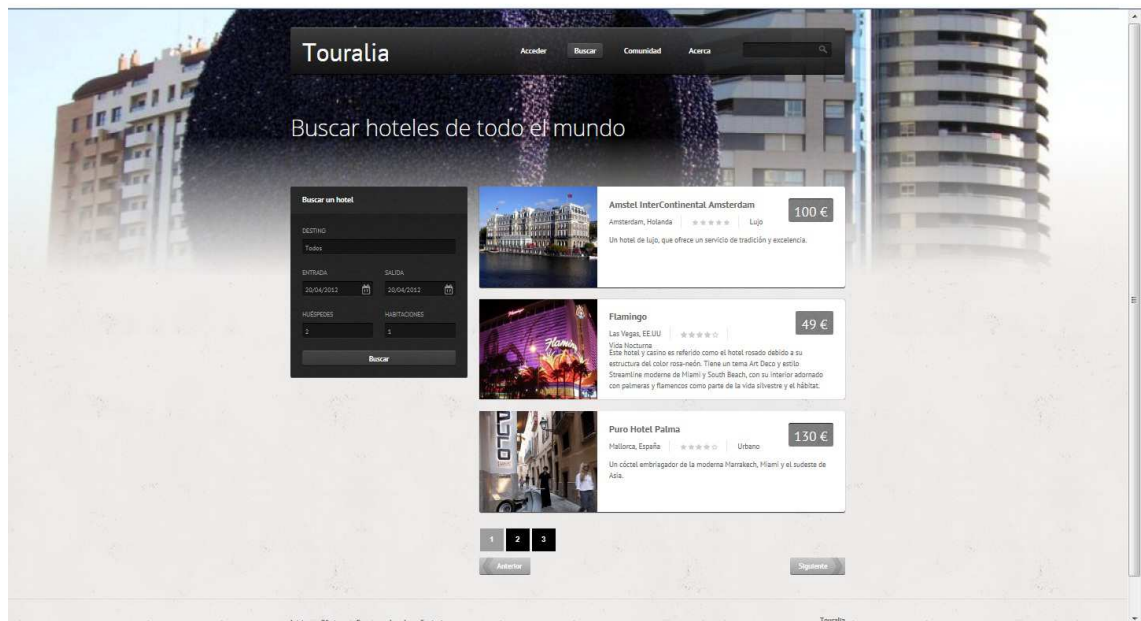


Ilustración 25. Vista de la página de búsqueda de hoteles

A continuación, en la siguiente Ilustración, mostramos el código necesario para la carga de los hoteles.

```
function getHoteles($offset = 0, $max_num_rows = 18446744073709551615) {  
  
    $hoteles = array();  
  
    $query = 'SELECT idHotel, nombre, localizacion, ciudad, provincia,  
        pais, website, num_hab_ind, num_hab_dob, num_hab_trip, categoria,  
        regimen, precio_SD, precio_AD, precio_MP, precio_PC, descripcion  
        FROM hotel LIMIT ' . $offset . ', ' . $max_num_rows;  
  
    $result = $this->mysql->query($query, MYSQLI_STORE_RESULT);  
  
    if ($result) {  
        while ($row = $result->fetch_object()) {  
            $hoteles[] = new Hotel($row->idHotel, $row->nombre,  
                $row->localizacion, $row->ciudad, $row->provincia,  
                $row->pais, $row->website, $row->num_hab_ind,  
                $row->num_hab_dob, $row->num_hab_trip,  
                Categoria::parseCategory($row->categoria),  
                $row->regimen, $row->precio_SD, $row->precio_AD,  
                $row->precio_MP, $row->precio_PC, $row->descripcion);  
        }  
        $result->free();  
    }  
  
    $this->fillImágenesHoteles($hoteles);  
  
    return $hoteles;  
}
```

Ilustración 26. Consulta necesaria para el acceso a la página de búsqueda de hoteles

Llegado a este punto, el usuario puede efectuar una búsqueda de un hotel mediante el formulario, en el que deberá introducir un destino, dos fechas, uno de llegada y otra de salida, y especificar el número de huéspedes y de habitaciones. Alternativamente, puede consultar información relativa al hotel pinchando sobre su cuadro de información directamente. Con ello, se accede a la página de detalles de los hoteles, que se muestra a continuación:

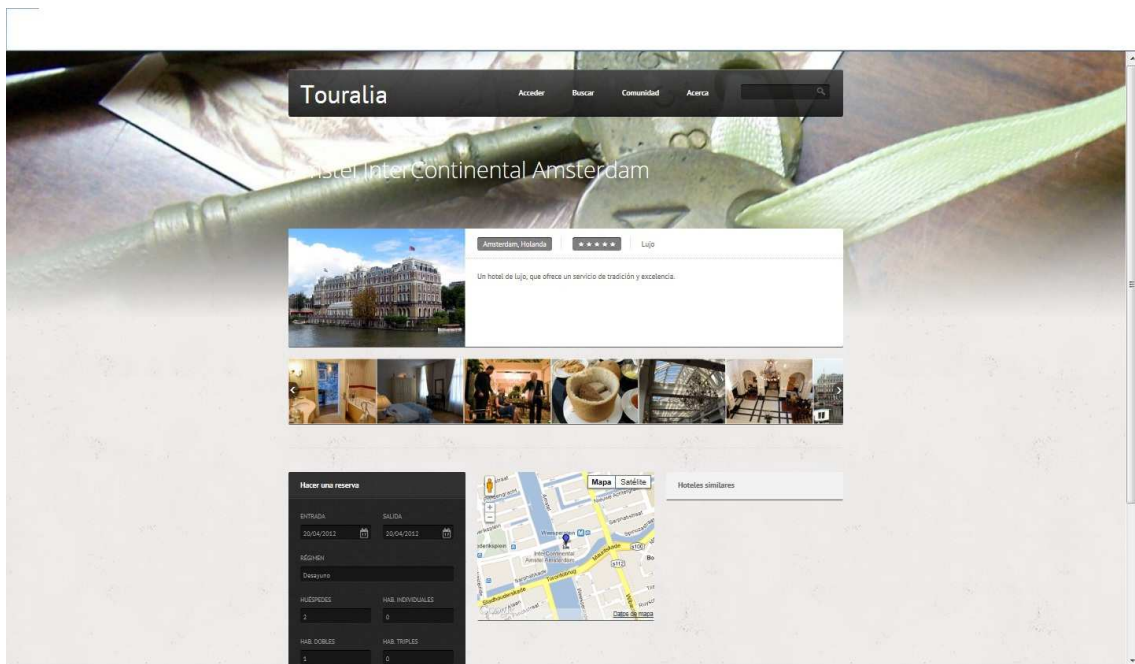


Ilustración 27. Vista de la página de detalles del hotel

Seguidamente, como veremos, aparece el código necesario para poder efectuar dichas búsquedas:

```
function searchHoteles($location, $fecha_entrada, $fecha_salida, $num_hab,
    $num_huespedes, $offset = 0, $max_num_rows = 18446744073709551615) {
    $hoteles = array();
    if ($location != Comunes::TODOS) {
        $query = 'SELECT h.idHotel, h.nombre, h.localizacion, h.ciudad,
            h.provincia, h.pais, h.website, h.num_hab_ind, h.num_hab_dob,
            h.num_hab_trip, h.categoria, h.regimen, h.precio_SD,
            h.precio_AD, h.precio_MP, h.precio_PC, h.descripcion,
            COALESCE(SUM(r.num_hab_ind), 0) AS res_ind,
            COALESCE(SUM(r.num_hab_dob), 0) AS res_dob,
            COALESCE(SUM(r.num_hab_trip), 0) AS res_trip
            FROM hotel h LEFT JOIN reservahotel r ON h.idHotel = r.idHotel
            WHERE h.ciudad = ?
            AND ((r.fecha_entrada >= ? AND r.fecha_salida <= ?)
            OR (r.fecha_entrada IS NULL AND r.fecha_salida IS NULL))
            GROUP BY r.idHotel, h.idHotel
            HAVING (h.num_hab_ind + h.num_hab_dob + h.num_hab_trip)
            - (res_ind + res_dob + res_trip) >= ?
            AND (h.num_hab_ind + h.num_hab_dob * 2 + h.num_hab_trip * 3)
            - (res_ind + res_dob * 2 + res_trip * 3) >= ?
            LIMIT ' . $offset . ', ' . $max_num_rows;
        $stmt = $this->mysql->prepare($query);
        $stmt->bind_param('sssii', $location, $fecha_entrada, $fecha_salida,
            $num_hab, $num_huespedes);
    } else {
        $query = 'SELECT h.idHotel, h.nombre, h.localizacion, h.ciudad,
```



```
h.provincia, h.pais, h.website, h.num_hab_ind, h.num_hab_dob,
h.num_hab_trip, h.categoria, h.regimen, h.precio_SD,
h.precio_AD, h.precio_MP, h.precio_PC, h.descripcion,
COALESCE(SUM(r.num_hab_ind), 0) AS res_ind,
COALESCE(SUM(r.num_hab_dob), 0) AS res_dob,
COALESCE(SUM(r.num_hab_trip), 0) AS res_trip
FROM hotel h LEFT JOIN reservahotel r ON h.idHotel = r.idHotel
WHERE ((r.fecha_entrada >= ? AND r.fecha_salida <= ?)
OR (r.fecha_entrada IS NULL AND r.fecha_salida IS NULL))
GROUP BY r.idHotel, h.idHotel
HAVING (h.num_hab_ind + h.num_hab_dob + h.num_hab_trip)
- (res_ind + res_dob + res_trip) >= ?
AND (h.num_hab_ind + h.num_hab_dob * 2 + h.num_hab_trip * 3)
- (res_ind + res_dob * 2 + res_trip * 3) >= ?
LIMIT ' . $offset . ' . $max_num_rows;
$stmt = $this->mysqli->prepare($query);
$stmt->bind_param('ssii', $fecha_entrada, $fecha_salida, $num_hab,
$num_huespedes);
}
$stmt->execute();
$stmt->bind_result($idHotel, $nombre, $localizacion, $ciudad,
$provincia, $pais, $website, $num_hab_ind, $num_hab_dob,
$num_hab_trip, $categoria, $regimen, $precio_SD, $precio_AD,
$precio_MP, $precio_PC, $descripcion, $res_ind, $res_dob,
$res_trip);
while ($stmt->fetch()) {
$hoteles[] = new Hotel($idHotel, $nombre, $localizacion, $ciudad,
```



```
$provincia, $pais, $website, $num_hab_ind, $num_hab_dob,  
$num_hab_trip, Categoria::parseCategory($categoria), $regimen,  
$precio_SD, $precio_AD, $precio_MP, $precio_PC,  
$descripcion);  
}  
$stmt->close();  
$this->fillImágenesHoteles($hoteles);  
return $hoteles;  
}
```

Ilustración 28. Consulta para las búsquedas de hoteles

Llegados a este punto, si el visitante desea efectuar una reserva debe registrarse en el sistema, si no lo ha hecho nunca, o en caso contrario efectuar un inicio de sesión. Para ambas acciones deberá ir al menú, pinchar sobre la opción Acceder y entonces se redireccionará a la página mostrada a continuación:

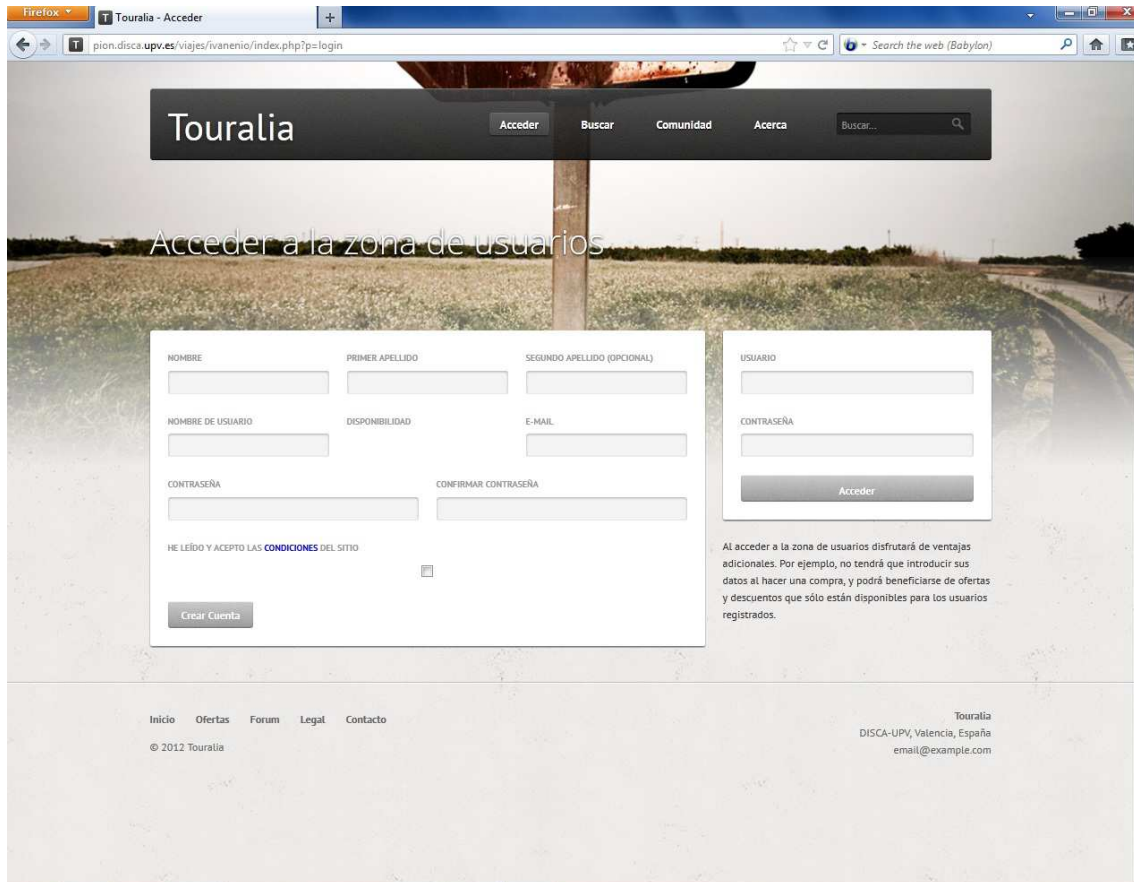


Ilustración 29. Página de acceso/inicio de sesión

Seguidamente podremos ver a continuación el código necesario para la inserción del nuevo usuario o la validación del usuario registrado:

```
function subscribe($usuario) {
    $query = "INSERT INTO usuario (login, password, email, nombre,
apellidos) SELECT ?, MD5(?), ?, ?, ? FROM DUAL WHERE NOT EXISTS
(SELECT login FROM usuario WHERE login = ?)";
    $stmt = $this->mysqli->stmt_init();
    $stmt->prepare($query);
    $stmt->bind_param('sssss', $login1, $password, $email, $nombre,
    $apellidos, $login2);
    $login1 = $usuario->login;
    $password = $usuario->password;
    $email = $usuario->email;
    $nombre = $usuario->nombre;
    $apellidos = $usuario->apellidos;
    $login2 = $usuario->login;
    $stmt->execute();
    $is_inserted = ($stmt->affected_rows > 0) ? TRUE : FALSE;
    $stmt->close();
    return $is_inserted;
}
```

Ilustración 30. Función de inserción de nuevo usuario

```
function userLogin($username, $password) {
    $query = "SELECT nombre, apellidos FROM usuario WHERE login = ? AND
password = MD5(?) LIMIT 1";
    $stmt = $this->mysqli->prepare($query);
    $stmt->bind_param('ss', $username, $password);
    $stmt->execute();
    $stmt->store_result();
    $res = null;
    if ($stmt->num_rows == 1) {
        $nombre = null;
        $apellidos = null;
        $stmt->bind_result($nombre, $apellidos);
        $stmt->fetch();
        $res = trim($nombre) . ' ' . trim($apellidos);
    }
    return $res;
}

function userExists($username) {
    $query = 'SELECT nombre FROM usuario WHERE login = ? LIMIT 1';
    $stmt = $this->mysqli->prepare($query);
    $stmt->bind_param('s', $username);
    $stmt->execute();
    $stmt->store_result();
    return ($stmt->num_rows == 1) ? TRUE : FALSE;
}
```

Ilustración 31. Funciones necesarias para la validación del usuario existente

Una vez que el usuario visitante se haya convertido en un usuario cliente se modificará ligeramente el menú, no apareciendo la opción Acceder en él, además de la consiguiente posibilidad de efectuar reservas y dejar comentarios en el fórum, el cual podrá consultar aunque no esté registrado, como se muestra seguidamente:

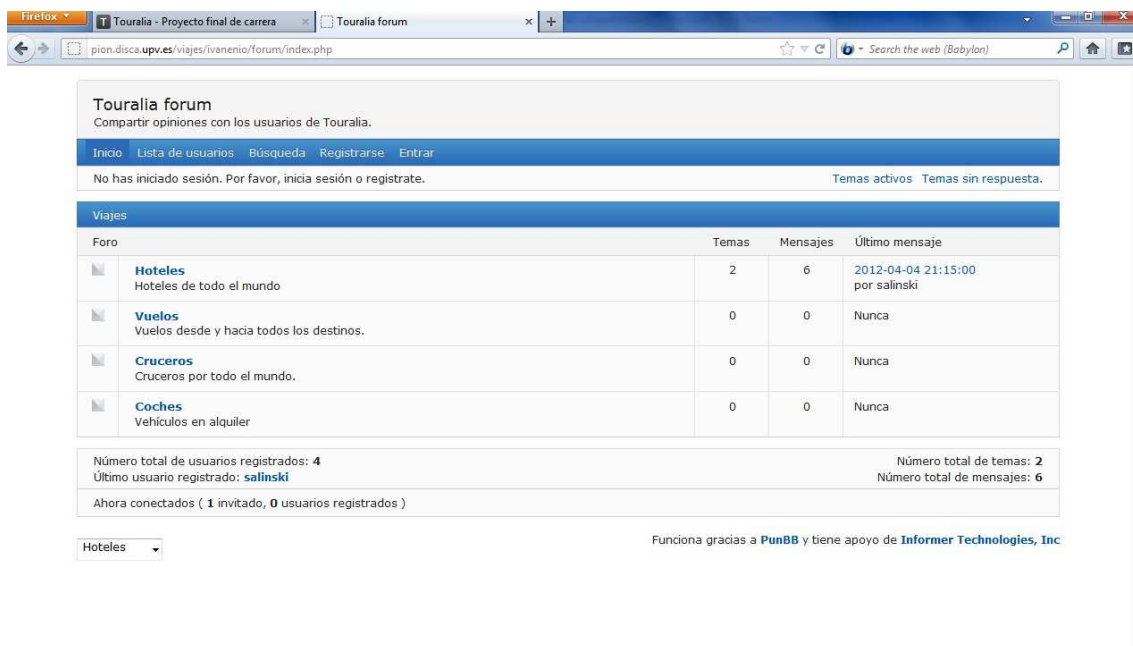


Ilustración 32. Imagen de la página principal del fórum

5.4.2. Usuario cliente

El usuario Cliente puede acceder a los lugares que precisan autenticación del usuario, como son el foro y la posibilidad de realización de reservas. Además cada cliente tiene acceso a sus datos para poder modificarlos cuando quiera y puede consultar también las reservas realizadas. Para ello, debe seguir los pasos explicados en el anterior apartado al final del mismo.



A partir de este punto el cliente ya puede realizar reservas y ver el historial de reservas realizadas desde la zona del cliente. El proceso de pago, como ya se indicó en el apartado de requisitos, no se ha tenido en cuenta, solo se introduce de forma simbólica un número de tarjeta que es con la que se debería realizar dicho pago.

Por otra parte, además, el usuario cliente también puede dejar comentarios en el fórum, tras registrarse e iniciar sesión. Nótese, que en proyecto, fórum y web son dos aplicaciones distintas con mecanismos distintos de identificación de usuarios. En cualquier caso, en ambos casos el funcionamiento de registro e inicio de sesión son sencillos de utilizar.

5.5. Evaluación y Pruebas

5.5.1. Introducción

La fase de evaluación y pruebas tiene como fin comprobar la respuesta que nuestra aplicación va a tener frente a distintas situaciones. Son pruebas básicas que todo sitio web debe superar con éxito. Mostramos a continuación los resultados obtenidos en las siguientes pruebas.

5.5.2. Pruebas

5.5.2.1. Pruebas de resolución

Las pruebas de resolución son necesarias para que los usuarios puedan ver correctamente la página web. Hay que cuidar mucho este aspecto, por eso hemos probado distintas resoluciones en la web <http://www.anybrowser.com>. Dicha web, se encarga de ajustar una página web, proporcionada mediante URL, a los distintos ajustes

El diseño de la página Web, como se ha indicado anteriormente, se ha realizado teniendo en cuenta la compatibilidad de la misma con distintas resoluciones de pantalla, debido a que hoy en día la utilización de dispositivos portátiles es cada vez mayor y se utilizan un mayor rango de tamaños de pantalla. Las pruebas realizadas muestran la compatibilidad del sitio de la agencia de viajes con hasta una resolución mínima de 640x480 píxeles sin utilizar *scroll* horizontal, como muestra la Ilustración 33.

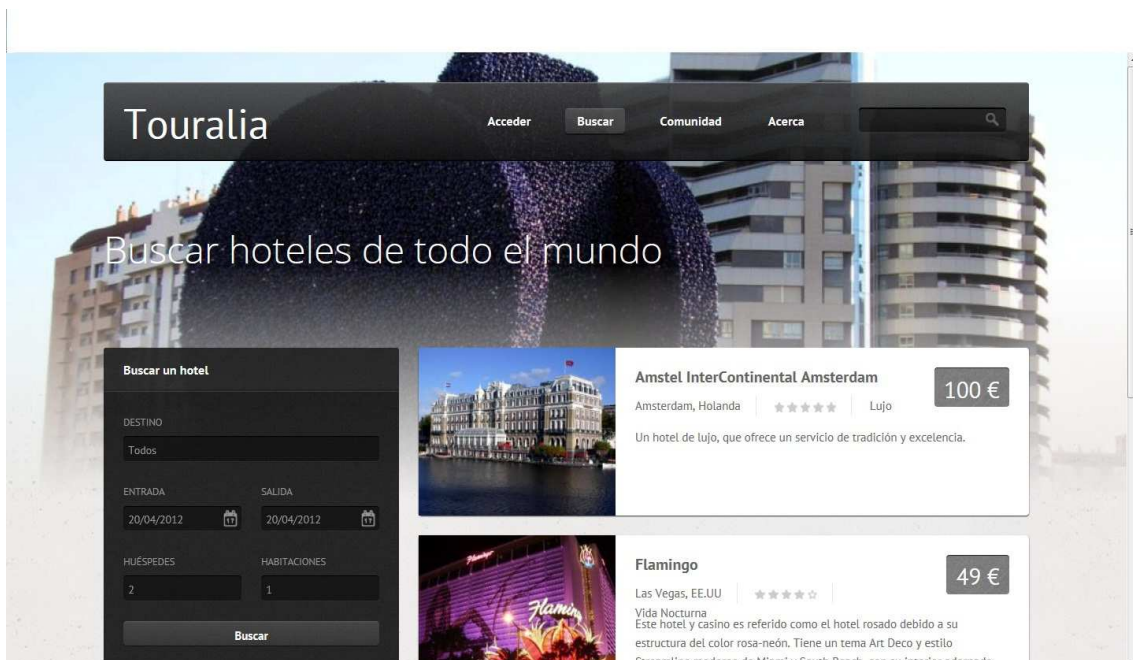


Ilustración 33. Resolución de 640x480

Seguidamente, hemos probado con una resolución de 800x600. Como se puede comprobar la visión no es todavía la más óptima pero sí que mejora.



Ilustración 34. Resolución de 800x600

Por último, le hemos aplicado la conllustración de 1024x768:

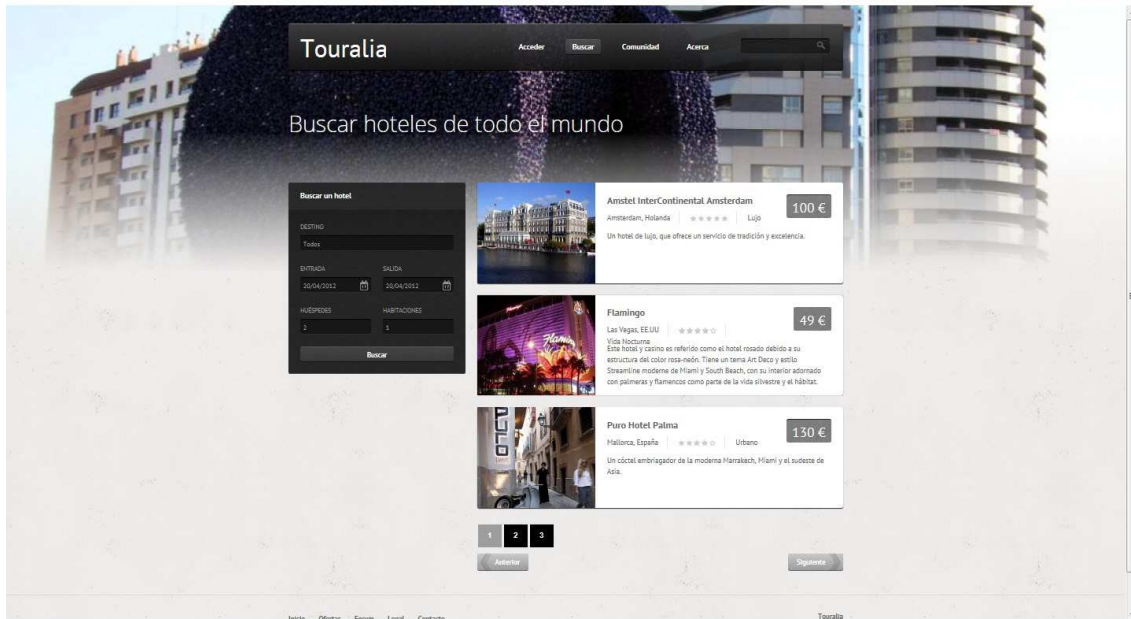


Ilustración 35. Resolución de 1024x768

5.5.2.2. Pruebas con navegadores

Esta prueba consiste en comprobar el correcto funcionamiento en los distintos navegadores. Hemos escogido los más representativos, Mozilla Firefox, Internet Explore, Google Chrome y Safari.

Mozilla Firefox

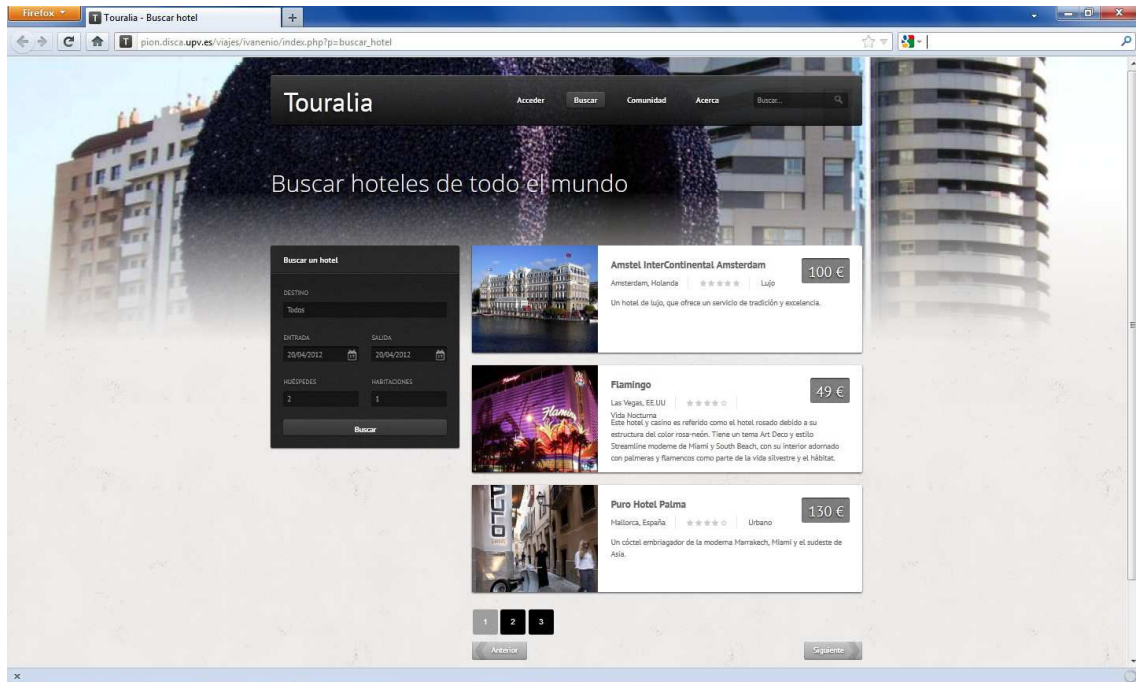


Ilustración 36. Vista del portal con Mozilla Firefox

Internet Explorer

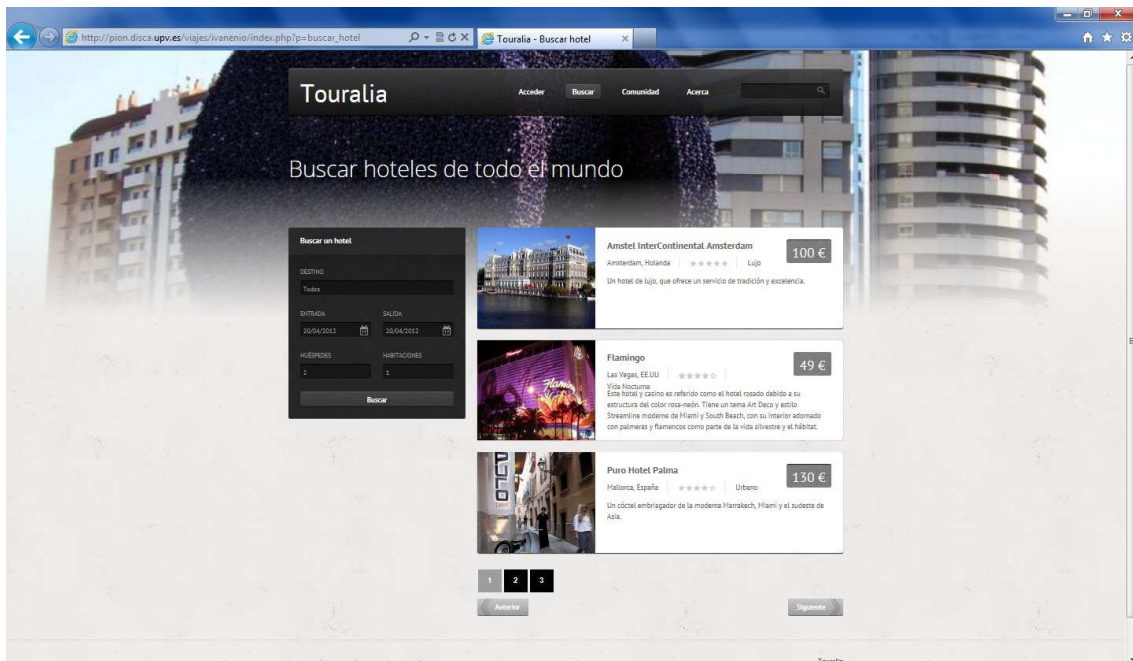


Ilustración 37. Vista del portal con Internet Explorer

Google Chrome

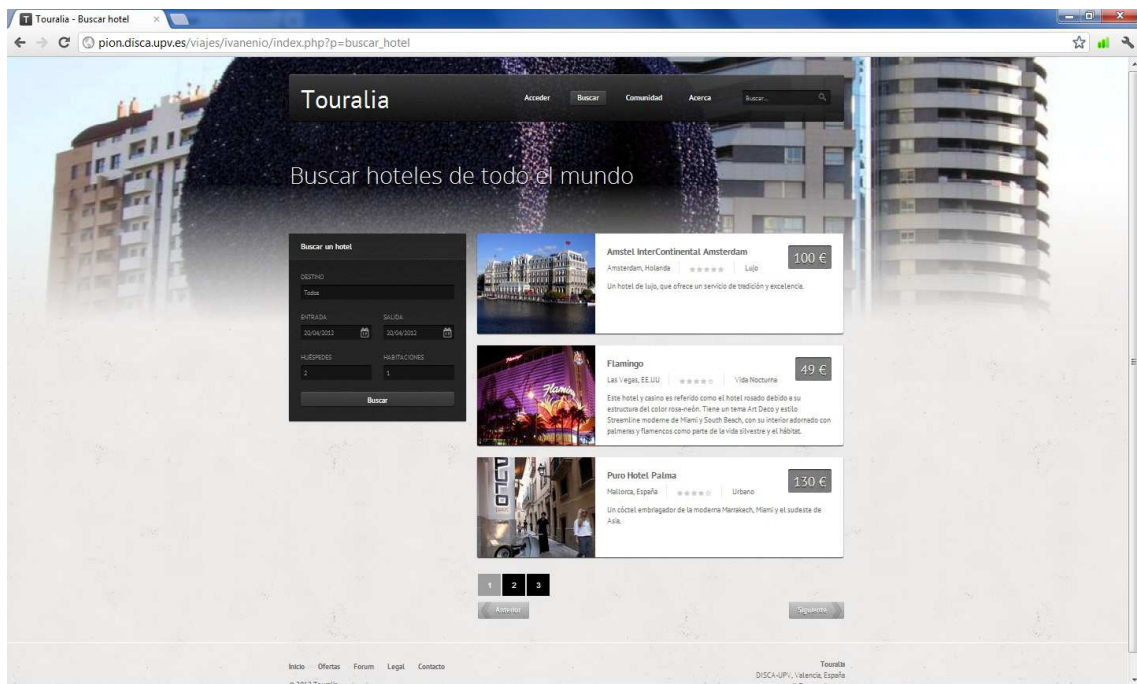


Ilustración 38. Vista del portal con Google Chrome

Safari

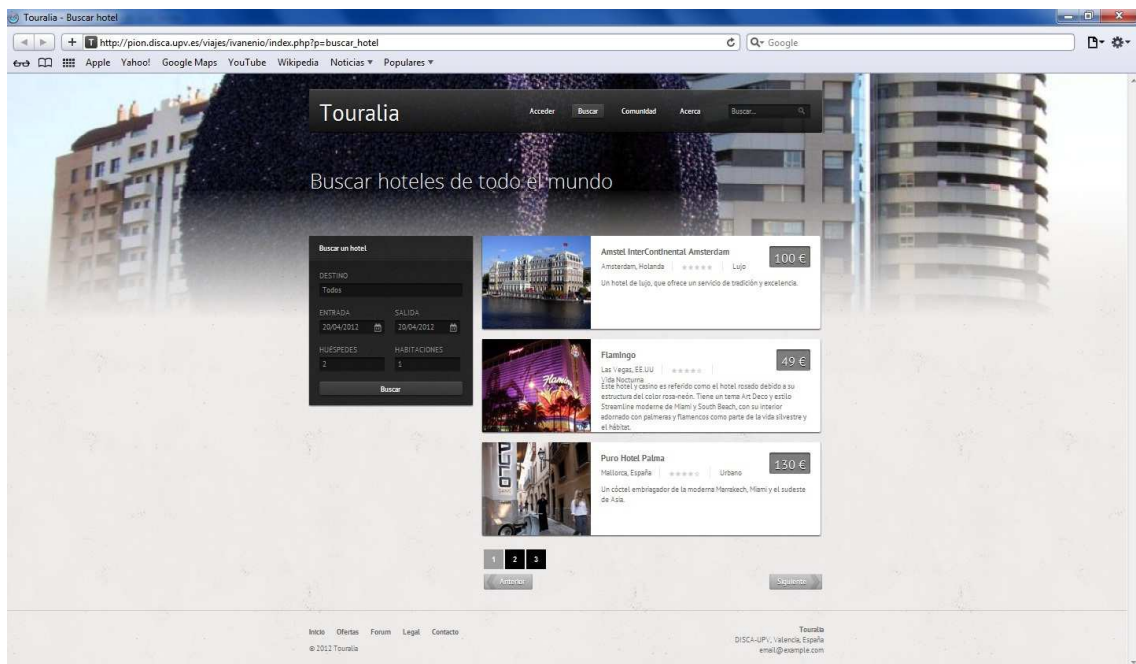


Ilustración 39. Vista del portal con Safari

5.5.2.3. Pruebas de validación

Los estilos css nos van a permitir cambiar el modo de las distintas etiquetas que se encuentran en las páginas HTML, siempre que estas páginas añadan dichos estilos. En este caso, al igual que en el estándar XHTML, se ha utilizado una aplicación online del w3c. La dirección para acceder a esta aplicación es la siguiente: *http://jigsaw.w3.org/css-validator/*. El resultado obtenido al pasarle el css de la Web se puede ver en la Ilustración.

The screenshot shows the W3C CSS Validator interface. At the top, it says 'El Servicio de Validación de CSS del W3C' and 'Resultados del Validador CSS del W3C para style.css (CSS versión 2.1)'. Below this, a green banner reads '¡Enhorabuena! No error encontrado.' followed by '¡Este documento es CSS versión 2.1 válido!'. A paragraph explains that users can show this icon on their pages to indicate interoperability. Two code snippets are provided: one for a yellow 'W3C CSS' icon and another for a blue 'W3C CSS' icon. At the bottom, there is a note about HTML 4.01 compatibility, a logo for HP, a donation link, and a '3119' counter.

Ilustración 40. Página de validación con el estándar w3c

6. Conclusiones

Finalmente, comentaremos las conclusiones que he sacado realizando este proyecto final de carrera.

Lo primero de todo fue decidir el tema acerca del cual iba a hacer una página web. Escogí una web para una agencia de viajes porque me parecía una posibilidad de crear una web con multitud de productos y variedad de servicios. Con ello, lo más importante en un primer momento era definir los requisitos que iba a tener la web. Inicialmente preví algunos servicios y productos que finalmente han sido descartados, por ser innecesarios, por ser costosos, así como por otras causas, como fue la imposibilidad de seguir adelante con mi compañero de proyecto o la falta de tiempo, pues me encuentro buscando trabajo en este momento y realizando unos cursos de inglés con vistas a realizar un máster. Respecto al diseño, ante lo pobre de mi primer prototipo decidí utilizar un diseño existente de una web de profesionales del diseño gráfico para mejorar el aspecto y su calidad visual. Para todo ello, ha sido fundamental la realización de un cursillo para un hipotético trabajo o la asistencia a clases de PHP pues debo confesar que mis nociones en este campo eran ínfimas.

La realización de este proyecto me ha permitido ostentar una visión global de un proyecto informático como es la programación de una web. Desde tener una idea tras leer un archivo pdf en la biblioteca de un proyecto anterior hasta la confección de esta memoria, último paso antes de finalizar el proyecto final de carrera. A lo largo de la realización del mismo, ha habido que tomar decisiones, efectuar cambios, mejorar aspectos relativos a la presentación, muestra de datos.etc., orientados principalmente por el tutor del proyecto, la visualización de webs similares o incluso la consulta a amigos y compañeros de clase.

Lo que me queda por añadir, al margen de las ampliaciones, que me parecen muchas, es que espero que este proyecto me sirva como experiencia y punto de partida a una vida laboral dedicada, a la ingeniería informática, y en especial al campo de la programación web.



Como mejoras propongo lo siguiente a modo de ejemplo:

- Adición de otros productos, tales como paquetes vacacionales.
- Adición de otros servicios, como las noticias o un sistema de alertas.
- Unificación de web y foro en un mismo entorno.
- Posibilidad de dotar a la web de un medio de pago por internet real
- Añadir información adicional a cada uno de los productos existentes.



7. Bibliografía

- “Una guía para la realización y supervisión de proyectos Web (PFC)” Félix Buendía, Editorial UPV 2007.
- “Php y MySQL” Davis, Michele E. y Phillips, Jon A., Ed. Anaya Multimedia.
- Enciclopedia Online Wikipedia <http://www.wikipedia.es>.
- Desarrollo Web <http://www.desarrolloweb.com>.
- W3 Schools <http://www.w3schools.com>.
- Web php.net.
- “PHP 6” Charle Ojeda, Francisco.
- “PHP 6: Un curso de programación” Andrea, Edgar d’.
- “PHP, MySQL y Apache” Meloni, Julie C.
- “Programación con PHP 6 y MySQL” Harris, Andy.
- <http://docs.jquery.com/Plugins/Validation>.
- <http://www.99points.info/2011/11/ajax-pagination-using-jquery-and-php-with-animation>.
- http://dev.mysql.com/doc/refman/5.5/en/encryption-functions.html#function_sha2.
- <http://youhack.me/2010/05/04/username-availability-check-in-registration-form-using-jqueryphp/>.

Anexos

Anexo A

Apartado	Descripción
Título	Área principal
Propósito	Mostrar el contenido de cada una de las opciones del menú de la barra horizontal superior
Posición	Zona central de la página principal
Formato	Imagen jpg, texto
Texto	Varios
Interacción	Contendrá diversos tipos de controles dependiendo del contenido a mostrar
Tamaño	Fijo
Medidas	

Apartado	Descripción
Título	Botón de búsqueda de coches
Propósito	Mostrar el botón de coches, para el acceso a la sección
Posición	Barra de navegación de menús horizontal
Formato	Imagen jpg
Texto	No
Interacción	Abre en el área principal de la página información sobre coches, con posibilidad de realizar un alquiler
Tamaño	Fijo
Medidas	

Apartado	Descripción
Título	Botón de búsqueda de cruceros
Propósito	Mostrar el botón de cruceros, para el acceso a la sección
Posición	Barra de navegación de menús horizontal
Formato	Imagen jpg
Texto	No
Interacción	Abre en el área principal de la página información sobre cruceros, con posibilidad de realizar una reserva
Tamaño	Fijo
Medidas	

Apartado	Descripción
Título	Botón de búsqueda de hoteles
Propósito	Mostrar el botón de hoteles, para el acceso a la sección
Posición	Barra de navegación de menús horizontal
Formato	Imagen jpg
Texto	No
Interacción	Abre en el área principal de la página información sobre hoteles, con posibilidad de realizar una reserva
Tamaño	Fijo
Medidas	

Apartado	Descripción
Título	Botón de búsqueda de vuelos
Propósito	Mostrar el botón de vuelos, para el acceso a la sección
Posición	Barra de navegación de menús horizontal
Formato	Imagen jpg
Texto	No
Interacción	Abre en el área principal de la página información sobre vuelos, con posibilidad de realizar una reserva
Tamaño	Fijo
Medidas	

Apartado	Descripción
Título	Foro
Propósito	Mostrar el botón de acceso al foro de comentarios
Posición	Zona derecha de la página principal
Formato	Imagen jpg
Texto	No
Interacción	Aparece una ventana donde se insertan comentarios
Tamaño	Fijo
Medidas	

Apartado	Descripción
Título	Nombre
Propósito	Aparece el nombre de la agencia de viajes
Posición	Barra superior
Formato	Imagen jpg
Texto	No
Interacción	No
Tamaño	Fijo
Medidas	
Apartado	Descripción
Título	Oferta



Propósito	Mostrar servicios cuyos precios son extraordinarios
Posición	Zona central de la página principal
Formato	Texto
Texto	Texto con la descripción de las ofertas
Interacción	Aparece una ventana con información sobre la oferta seleccionada
Tamaño	Fijo
Medidas	

Apartado	Descripción
Título	Pie de página
Propósito	Decorativo, acceso a inicio, fórum
Posición	Zona inferior de la página principal
Formato	Texto
Texto	Nombre de sección de acceso
Interacción	Enlaza a la página principal, la zona de fórum
Tamaño	Fijo
Medidas	

Anexo B

Usuarios anónimos

Especificación de requisitos funcionales			
Nombre	Descripción		
Buscar coches en alquiler	Se accede desde el menú horizontal de la página principal.		
Entradas	Ciudad de recogida del coche, ciudad de devolución del coche, fecha de recogida y fecha de devolución del vehículo.		
Fuente	Formulario de consulta de coches en alquiler.	Salida	Si el registro se encuentra en la base de datos, se mostrará un enlace con algunas características del registro.
Destino	Bases de datos.	Restricciones	Todos los campos han de estar completos.
Proceso	Con los datos introducidos se realizará una búsqueda en la base de datos. Si se encuentra el registro buscado se listará en el área principal de la página.		

Especificación de requisitos funcionales			
Nombre	Descripción		
Buscar cruceros	Se accede desde el menú horizontal de la página principal.		
Entradas	Fecha de salida, fecha de llegada, zona, número de huéspedes, número de habitaciones.		
Fuente	Formulario de consulta de cruceros.	Salida	Si el registro se encuentra en la base de datos, se mostrará un enlace con algunas características del registro.
Destino	Bases de datos.	Restricciones	Todos los campos han de estar completos.
Proceso	Con los datos introducidos se realizará una búsqueda en la base de datos. Si se encuentra el registro buscado se listará en el área principal de la página.		

Especificación de requisitos funcionales			
Nombre	Descripción		
Buscar hoteles	Se accede desde el menú horizontal de la página principal. También desde la ventana de búsqueda de la misma.		
Entradas	Lugar de destino, fechas de entrada y de salida, número de noches, número de habitaciones.		
Fuente	Formulario de consulta de hoteles.	Salida	Si el registro se encuentra en la base de datos, se mostrará un enlace con algunas características del registro.
Destino	Bases de datos.	Restricciones	Todos los campos han de estar completos.
Proceso	Con los datos introducidos se realizará una búsqueda en la base de datos. Si se encuentra el registro buscado se listará en el área principal de la página.		

Especificación de requisitos funcionales			
Nombre	Descripción		
Buscar vuelos	Se accede desde el menú horizontal de la página principal.		
Entradas	Lugar de origen y lugar de destino, fecha de salida y de regreso, con opción de elegir sólo ida, número de pasajeros.		
Fuente	Formulario de consulta de vuelos.	Salida	Si el registro se encuentra en la base de datos, se mostrará un enlace con algunas características del registro.
Destino	Bases de datos.	Restricciones	Todos los campos han de estar completos.
Proceso	Con los datos introducidos se realizará una búsqueda en la base de datos. Si se encuentra el registro buscado se listará en el área principal de la página.		

Especificación de requisitos funcionales			
Nombre	Descripción		
Consulta ofertas.	Se accede desde la columna central de la página principal.		
Entradas	Ninguna.		
Fuente	Página de ofertas.	Salida	Página de ofertas.
Destino	Ninguno.	Restricciones	Ninguna.
Proceso	Al visitar la web, en la zona donde aparecen los destinos, existe una sección con ofertas asociadas a los viajes.		

Especificación de requisitos funcionales			
Nombre	Descripción		
Registro usuario	Se accede desde la columna derecha de la página principal.		
Entradas	Nombre, Apellidos, correo electrónico, login y Password.		
Fuente	Formulario introducción de datos.	Salida	Confirmación de registro del usuario.
Destino	Bases de datos	Restricciones	Todos los campos han de estar completos.
Proceso	El usuario anónimo tendrá la posibilidad de registrarse en la página Web, para ello tendrá que proporcionar, el nombre, los apellidos, correo electrónico, login y el Password. Se tendrá que ofrecer un formulario de registro para proceder.		

Usuario registrado

Especificación de requisitos funcionales			
Nombre	Descripción		
Alquilar coche	Se accede desde la página de detalles de cada vehículo en alquiler.		
Entradas	Ninguna.		
Fuente	Formulario alquiler coches.	Salida	Confirmación del alquiler.
Destino	Bases de datos.	Restricciones	Ninguna.
Proceso	Se comprobará la posibilidad (si hay disponibilidad) de realizar dicha reserva. Si es posible, se procederá al pago de dicha reserva.		

Especificación de requisitos funcionales			
Nombre	Descripción		
Autenticación	Se accede desde la sección Acceder el menú de la página principal.		
Entradas	Nombre de usuario y contraseña.		
Fuente	Formulario de registro.	Salida	Ninguna. El usuario puede acceder a los servicios.
Destino	Base de datos.	Restricciones	Todos los campos han de estar completos.
Proceso	Se comprueba en la base de datos la existencia del usuario a partir del nombre de usuario y la contraseña introducidos. Si existe, el usuario estará autenticado (cliente) y podrá acceder a todos los servicios, si no, continuará como visitante.		

Especificación de requisitos funcionales			
Nombre	Descripción		
Introducción comentario en foro	Se accede desde la sección centro de la página principal.		
Entradas	Nombre de usuario, fecha y hora de publicación del mensaje, tema y cuerpo del mensaje, categoría del mensaje.		
Fuente	Formulario de comentarios.	Salida	Se guardarán los datos introducidos en la base de datos. Al acceder al foro se listarán los registros guardados en la base de datos según la sección del foro a la que se accede.
Destino	Base de datos.	Restricciones	Se debe indicar el nombre de usuario.
Proceso	Se guardarán los datos introducidos en la base de datos. Al acceder al foro se listarán los registros guardados en la base de datos según la sección del foro a la que se accede.		

Especificación de requisitos funcionales			
Nombre	Descripción		
Reserva crucero	Se accede desde la página de detalles de cada crucero.		
Entradas	Ninguna.		
Fuente	Formulario reserva cruceros.	Salida	Confirmación de la reserva.
Destino	Bases de datos.	Restricciones	Ninguna.
Proceso	Se comprobará la posibilidad (si hay disponibilidad) de realizar dicha reserva. Si es posible, se procederá al pago de dicha reserva.		

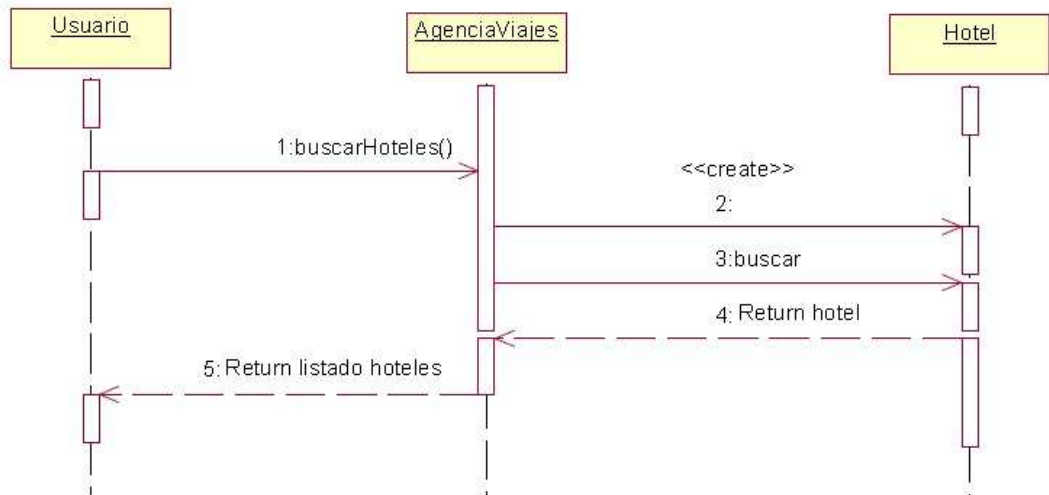
Especificación de requisitos funcionales			
Nombre	Descripción		
Reservar hotel	Se accede desde la página de detalles de cada hotel.		
Entradas	Ninguna.		
Fuente	Formulario reserva destino.	Salida	Confirmación de la reserva
Destino	Bases de datos.	Restricciones	Ninguna.
Proceso	Se comprobará la posibilidad (si hay habitaciones disponibles o no) de realizar dicha reserva. Si es posible, se procederá al pago de dicha reserva.		



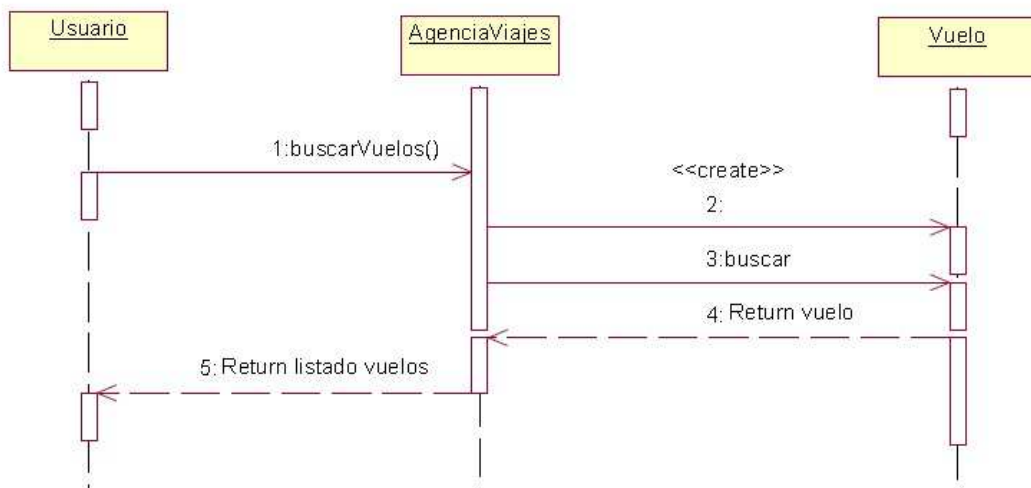
Especificación de requisitos funcionales			
Nombre	Descripción		
Reserva vuelos	Se accede desde la página de detalles de cada vuelo.		
Entradas	Ninguna.		
Fuente	Formulario reserva vuelos.	Salida	Confirmación de la reserva.
Destino	Bases de datos.	Restricciones	Ninguna.
Proceso	Se comprobará la posibilidad (si hay disponibilidad) de realizar dicha reserva. Si es posible, se procederá al pago de dicha reserva.		

Anexo C

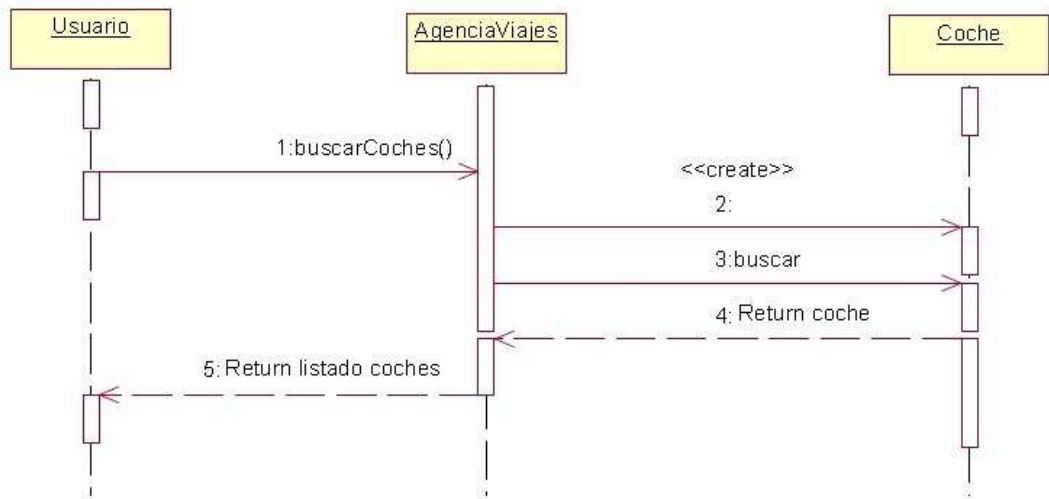
1. Búsqueda de un hotel



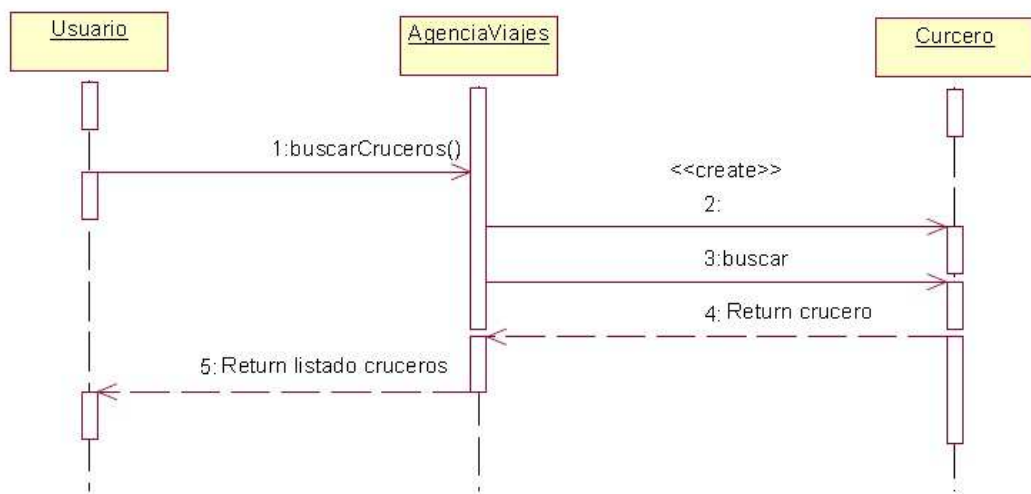
2. Búsqueda de un vuelo



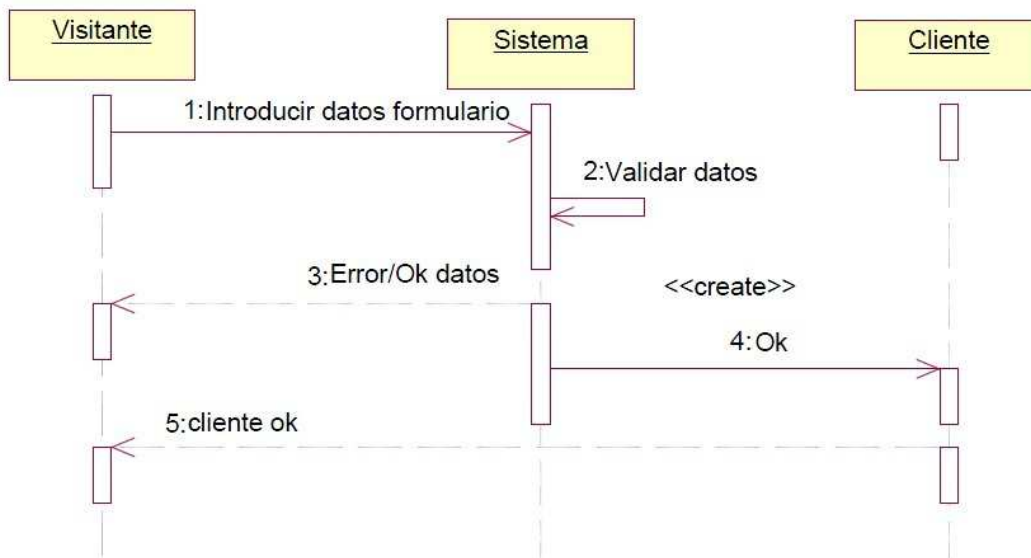
3. Búsqueda de un coche



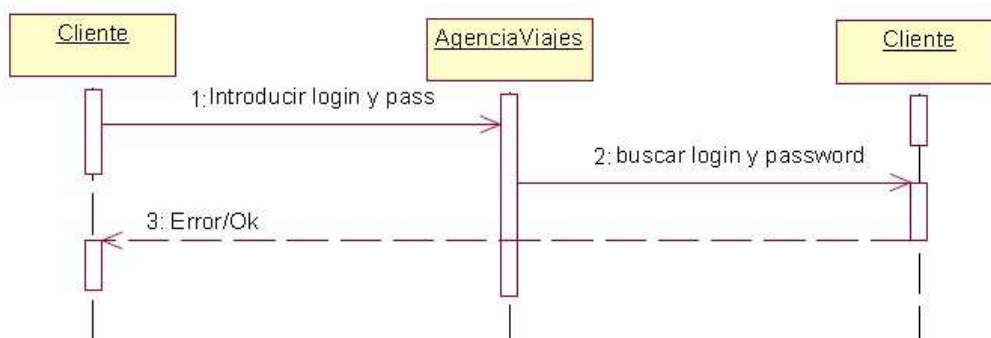
4. Búsqueda de crucero



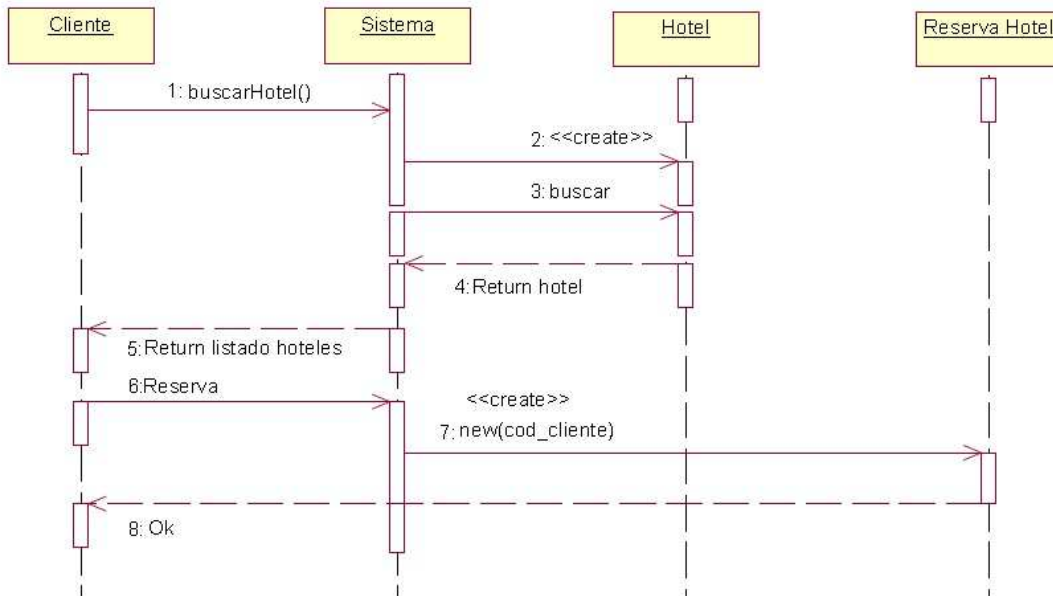
5. Registro de usuario



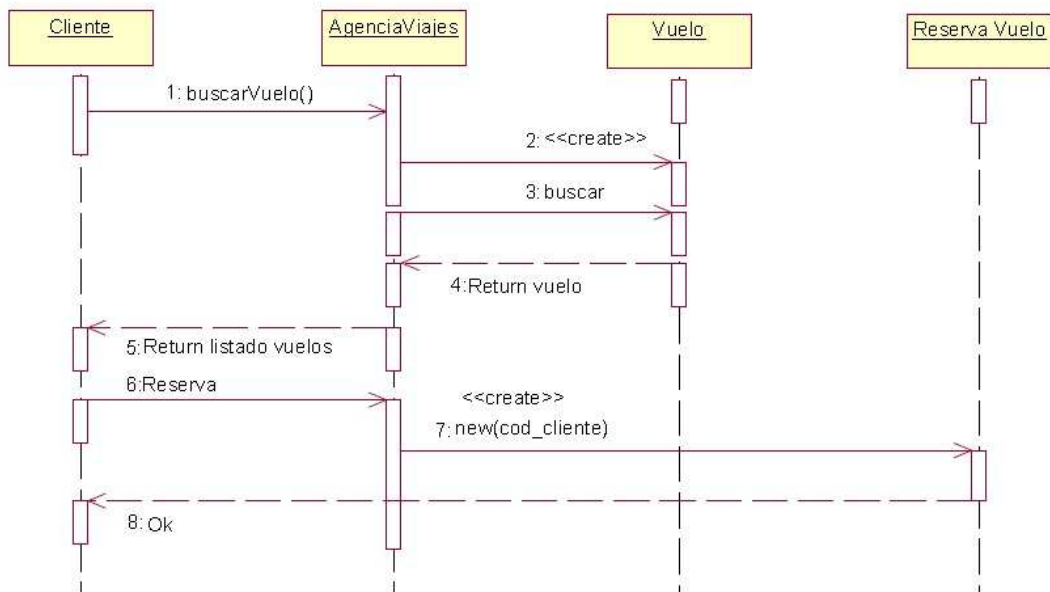
6. Inicio sesión



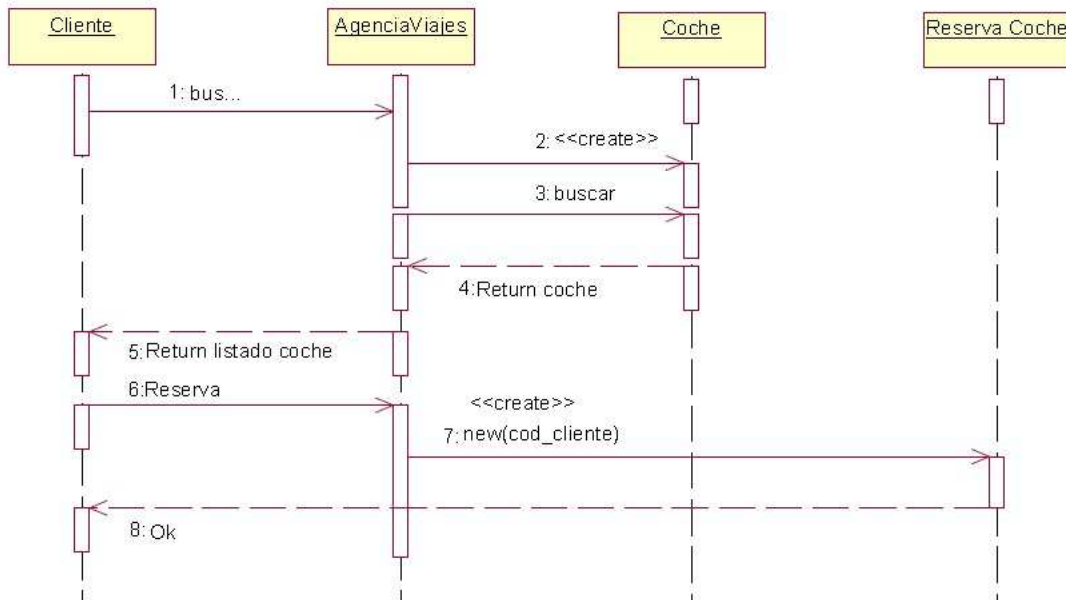
7. Reserva de hotel



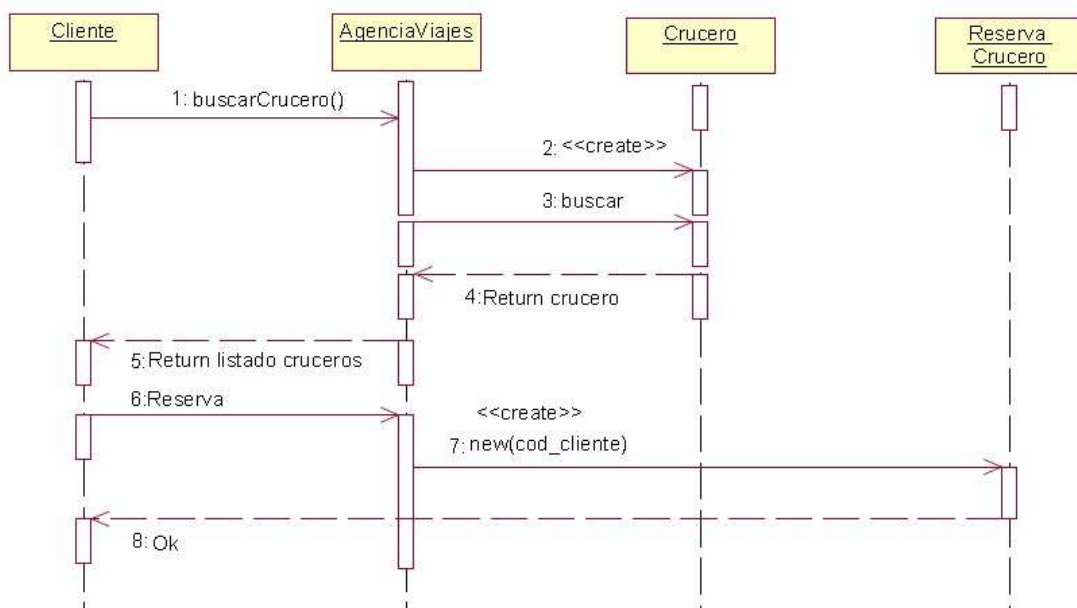
8. Reserva de vuelos



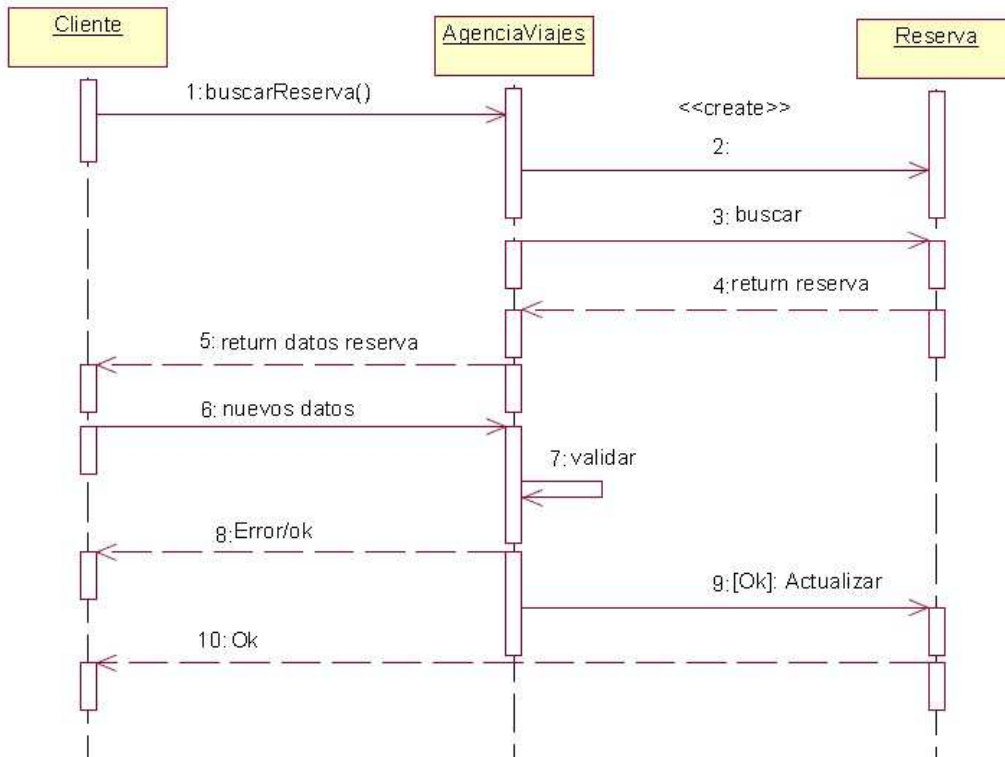
9. Reserva coches



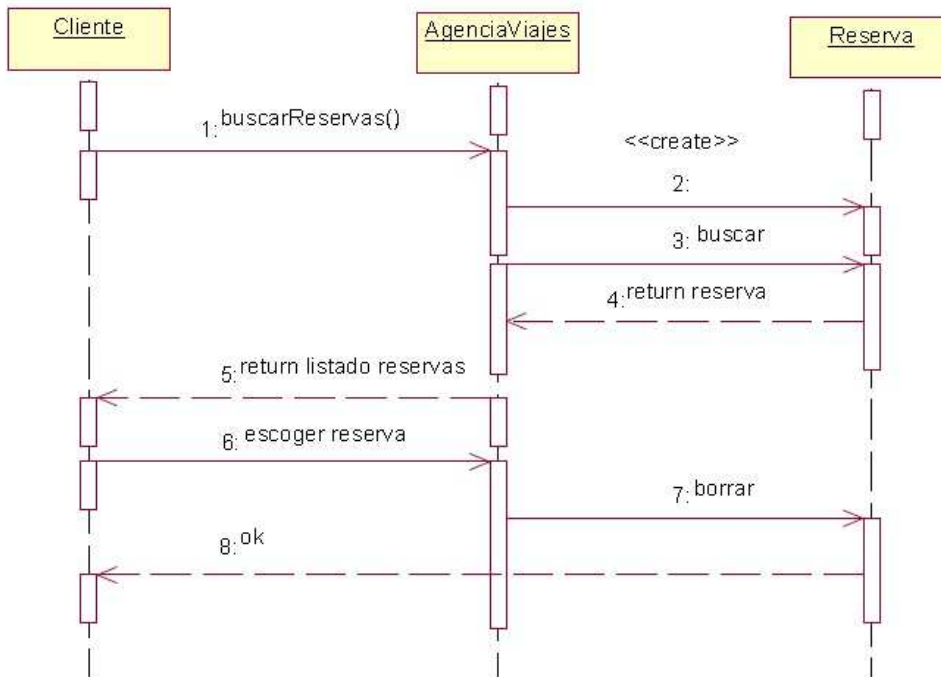
10. Reserva crucero



11. Modificar datos reserva



12. Eliminar reserva



Anexo D

Buscar_hoteles.tpl

```
<!doctype html>

<!--[if lt IE 7]> <html class="no-js ie6 oldie" lang="es"> <![endif]-->

<!--[if IE 7]> <html class="no-js ie7 oldie" lang="es"> <![endif]-->

<!--[if IE 8]> <html class="no-js ie8 oldie" lang="es"> <![endif]-->

<!--[if gt IE 8]><!--> <html class="no-js" lang="es"> <!--<![endif]-->

<head>

  <meta charset="utf-8">

  <title>{$title}</title>

  <meta name="description" content="{$description}">

  <meta name="author" content="{$author}">

  <!-- Load CSS -->

  <link rel="stylesheet" href="css/style.css">

  <link rel="stylesheet" href="fancybox/jquery.fancybox-1.3.4.css">

  <link rel="stylesheet" href="css/smoothness/jquery-ui-{$jquery_ui_version}.custom.css">

  <link rel="stylesheet" href="css/search.css">

  <!-- Page icon -->

  <link rel="shortcut icon" href="favicon.png">

  <!-- Load Modernizr -->

  <script src="js/libs/modernizr-2.0.min.js"></script>
```

```
<!-- Load JavaScript -->
```

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/{jQuery_version}/jquery.min.js"></script>
```

```
<script>window.jQuery || document.write('<script src="js/libs/jquery-  
{jQuery_version}.min.js"></script>')</script>
```

```
<script src="js/libs/jquery-ui-{jQuery_ui_version}.custom.min.js"></script>
```

```
<script src="js/libs/jquery.ui.datepicker-es.js"></script>
```

```
<script src="js/script.js"></script>
```

```
<script src="fancybox/jquery.fancybox-1.3.4.pack.js"></script>
```

```
<script src="http://maps.googleapis.com/maps/api/js?sensor=false"></script>
```

```
<script type="text/javascript">
```

```
current_page = "1";
```

```
$(function() {
```

```
var availableCiudades = [{loop="available_ciudades_list"
```

```
"{$value.ciudad}"{$value.separator}{else}
```

```
{/loop}
```

```
];
```

```
$("#destino").autocomplete({ source: availableCiudades });
```

```
});
```

```
$(document).ready(function(){
```

```
function showLoader(){
```

```
$('.search-background').fadeIn(200);
```

```
}
```

```
function hideLoader(){
```

```
$('.search-background').fadeOut(200);
```

```
};
```

```
$("#paging_button li").click(function(){
```

```
current_page = this.id;
```

```
showLoader();
```

```
$("#paging_button li").css({'background-color' : ''});

$(this).css({'background-color' : '#9d9d9d'});

$("#content").load("index.php?p=ajax_hotel"

+ "&per_page=" + $('input[name="per_page"]').val()

+ "&start_page=" + this.id,

hideLoader);

return false;

});

$("#nav_arrows a").click(function(){

var goto_page = "#-123";

if (this.id == "previous_arrow") {

//alert('Handler for .click() called from PREVIOUS.');
```

```
goto_page = "#" + (parseInt(current_page) - 1).toString();

} else if (this.id == "next_arrow") {

//alert('Handler for .click() called from NEXT.');
```

```
goto_page = "#" + (parseInt(current_page) + 1).toString();

}

if ($goto_page){

var e = jQuery.Event("click");

jQuery(goto_page).trigger(e);

}

return false;

});

$.datepicker.setDefaults( $.datepicker.regional[ "es" ] );

$("#1").css({'background-color' : '#9d9d9d'});

showLoader();

$("#content").load("index.php?p=ajax_hotel"
```



```
+ "&per_page=" + $('input[name="per_page"]').val()

+ "&start_page=1",

hideLoader);

});

</script>

</head>

<body>

<!-- Header -->

<header style="background-image:url(tpl/img/placeholders/1280x1024/{$bg_image});">

<div class="container_12">

<!-- Title and navigation panel -->

<div id="panel" class="grid_12">

<!-- Title -->

<h1><a href="index.php">{$h1_title}</a></h1>

<!-- Navigation -->

<nav>

{include="navigation"}

<!-- Search -->

{include="search" cache="3600"}

</nav>
```

```
</div>
```

```
</div>
```

```
<!-- Heading -->
```

```
<h2>{$h2_title}</h2>
```

```
</header>
```

```
<!-- Main content -->
```

```
<div class="container_12">
```

```
<!-- Search form -->
```

```
<section class="search_box sidebar grid_4">
```

```
<h2>Buscar un hotel</h2>
```

```
<!-- Find a hotel form -->
```

```
<form action="index.php" class="black">
```

```
<label>Destino</label>
```

```
<input type="text" name="destino" id="destino" value="Todos" />
```

```
<div class="half">
```

```
<label>Entrada</label>
```

```
<input type="text" name="entrada" class="date" value="{$date}" />
```

```
</div>
```

```
<div class="half last">

  <label>Salida</label>

  <input type="text" name="salida" class="date" value="{date}" />

</div>

<div class="half">

  <label>Hu&eacute;spedes</label>

  <input type="text" name="huespedes" value="2" />

</div>

<div class="half last">

  <label>Habitaciones</label>

  <input type="text" name="habitaciones" value="1" />

</div>

<input type="hidden" name="p" value="buscar_hotel" />

<input type="hidden" name="per_page" value="{per_page}" />

<input type="submit" value="Buscar" name="Buscar">

</form>

</section>

<!-- Results -->

<div id="container">
```

```
<div class="search-background">

  <label></label>

</div>

<div id="content">

  &nbsp;

</div>

</div>

<div id="paging_button" align="center">

  {include="paging_buttons"}

</div>

<div class="clearfix"></div>

<!-- Pagination -->

<nav id="nav_arrows" class="grid_8 prefix_4">

  <a href="#" id="previous_arrow" class="previous">Anterior</a>

  <a href="#" id="next_arrow" class="next">Siguiete</a>

</nav>

</div>

<!-- Footer -->

<footer><div class="container_12">

  {include="footer" cache="3600"}

</div></footer>
```



```
<!-- Google Analytics -->

<script type="text/javascript">

    var _gaq = _gaq || [];

    _gaq.push(['_setAccount', '{$google_analytics_id}']); // Set your Google Analytics ID here

    _gaq.push(['_trackPageview']);

    (function() {

        var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;

        ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-
analytics.com/ga.js';

        var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);

    })();

</script>

</body>
</html>
```

BuscadorHoteles.php

```
<?php

require_once 'forms/Plantilla.php';

require_once 'forms/UIUtils.php';

require_once 'sistema/AgenciaViajes.php';

require_once 'sistema/Comunes.php';
```

```
class BuscadorHoteles extends Plantilla {

    const per_page = 3;

    function __construct($page, $sec_session) {
        parent::__construct($page, $sec_session);
    }

    function Load() {
        parent::Load();

        $agencia = new AgenciaViajes();

        if (isset($_GET['Buscar'])) { // Formulario de búsqueda de hoteles

            $_SESSION['buscar_hotel_destino'] =

                (!isset($_GET['destino']) || empty($_GET['destino'])

                || strtoupper($_GET['destino']) == Comunes::TODOS)

                ? Comunes::TODOS : $_GET['destino'];

            $_SESSION['buscar_hotel_entrada'] = ( isset($_GET['entrada']) )

                ? UIUtils::parseDateToISO8601( $_GET['entrada'] ) : UIUtils::nowDateEs();

            $_SESSION['buscar_hotel_salida'] = ( isset($_GET['salida']) )

                ? UIUtils::parseDateToISO8601( $_GET['salida'] ) : UIUtils::nowDateEs();

            if( strtotime($_SESSION['buscar_hotel_entrada']) >
                strtotime($_SESSION['buscar_hotel_salida']) ) {

                $_SESSION['buscar_hotel_salida'] = $_SESSION['buscar_hotel_entrada'];

            }

            $_SESSION['buscar_hotel_huespedes'] =

                ( isset($_GET['huespedes']) && is_numeric( $_GET['huespedes'] ) &&
                intval($_GET['huespedes']) > 0 )
```

```
? intval($_GET['huespedes']) : 2;

$_SESSION['buscar_hotel_habitaciones'] =

( isset($_GET['habitaciones']) && is_numeric( $_GET['habitaciones']) &&
intval($_GET['habitaciones']) > 0 )

? intval($_GET['habitaciones']) : 1;

$destino = ($_SESSION['buscar_hotel_destino'] != Comunes::TODOS)

? $_SESSION['buscar_hotel_destino'] : 'cualquier destino';

$habitaciones = $_SESSION['buscar_hotel_habitaciones'];

.(($_SESSION['buscar_hotel_habitaciones'] == 1) ? 'habitación' : 'habitaciones');

$huespedes = $_SESSION['buscar_hotel_huespedes'];

.(($_SESSION['buscar_hotel_huespedes'] == 1) ? 'huésped' : 'huéspedes');

$h2_title = $habitaciones.' en '.$destino.' para '.$huespedes.', del '

.UIUtils::esDateFromISO8601($_SESSION['buscar_hotel_entrada'])

.' al '.UIUtils::esDateFromISO8601($_SESSION['buscar_hotel_salida']);

$total_hoteles = $agencia->countHotelesResult(

    $_SESSION['buscar_hotel_destino'],

    $_SESSION['buscar_hotel_entrada'],

    $_SESSION['buscar_hotel_salida'],

    $_SESSION['buscar_hotel_habitaciones'],

    $_SESSION['buscar_hotel_huespedes'] );

} else {

    $this->DropThisFromSession();

    $h2_title = 'Buscar hoteles de todo el mundo';

    $total_hoteles = $agencia->countHoteles();

}

$max_num_rows = ( isset($_GET['per_page']) ) ? $_GET['per_page'] :
BuscadorHoteles::per_page;

// assign variables to template
```

```
$this->tpl->assign('title', 'Touralia - Buscar hotel');

$this->tpl->assign('h1_title', 'Touralia');

$this->tpl->assign('h2_title', $h2_title);

$this->tpl->assign('bg_image', '05.jpg');

$this->tpl->assign('available_ciudades_list', $agencia->getCiudadesFromHotel());

$this->tpl->assign('per_page', $max_num_rows);

// pagging buttons

$total_pages = ceil($total_hoteles / $max_num_rows);

$paging_array = array();

for ($i = 1; $i <= $total_pages; $i++)

    $paging_array[] = array('id' => $i);

$this->tpl->assign("paging_results", $paging_array);

// create page

$html = $this->tpl->draw(TPlantilla::BuscadorHoteles, $return_string = true);

return $html;

}

private function DropThisFromSession() {

    unset($_SESSION['buscar_hotel_destino']);

    unset($_SESSION['buscar_hotel_entrada']);

    unset($_SESSION['buscar_hotel_salida']);

    unset($_SESSION['buscar_hotel_huespedes']);

    unset($_SESSION['buscar_hotel_habitaciones']);

}

}
```



```
?>
```

Detalles_hotel.tpl

```
<!doctype html>

<!--[if lt IE 7]> <html class="no-js ie6 oldie" lang="es"> <![endif-->

<!--[if IE 7]> <html class="no-js ie7 oldie" lang="es"> <![endif-->

<!--[if IE 8]> <html class="no-js ie8 oldie" lang="es"> <![endif-->

<!--[if gt IE 8]><!--> <html class="no-js" lang="es"> <!--<![endif-->

<head>

  <meta charset="utf-8">

  <title>{$title}</title>

  <meta name="description" content="{ $description}">

  <meta name="author" content="{ $author}">

  <!-- Load CSS -->

  <link rel="stylesheet" href="css/style.css">

  <link rel="stylesheet" href="fancybox/jquery.fancybox-1.3.4.css">

  <link rel="stylesheet" href="css/smoothness/jquery-ui-{$jquery_ui_version}.custom.css">

  <!-- Page icon -->

  <link rel="shortcut icon" href="favicon.png">

  <!-- Load Modernizr -->

  <script src="js/libs/modernizr-2.0.min.js"></script>
```

```
<!-- Load JavaScript -->
```

```
<script
```

```
src="http://ajax.googleapis.com/ajax/libs/jquery/{jQuery_version}/jquery.min.js"></script>
```

```
<script>window.jQuery || document.write('<script src="js/libs/jquery-  
{jQuery_version}.min.js"></script>')</script>
```

```
<script src="js/libs/jquery-ui-{jQuery_ui_version}.custom.min.js"></script>
```

```
<script src="js/script.js"></script>
```

```
<script src="fancybox/jquery.fancybox-1.3.4.pack.js"></script>
```

```
<script src="http://maps.google.com/maps/api/js?sensor=false"
```

```
type="text/javascript"></script>
```

```
<script type="text/javascript">
```

```
//</pre></div><div data-bbox="183 470 326 484" data-label="Text"><pre>var customIcons = {</pre></div><div data-bbox="198 500 258 513" data-label="Text"><pre>Hotel: {</pre></div><div data-bbox="213 526 694 542" data-label="Text"><pre>icon: 'http://labs.google.com/ridefinder/images/mm_20_blue.png',</pre></div><div data-bbox="213 555 740 570" data-label="Text"><pre>shadow: 'http://labs.google.com/ridefinder/images/mm_20_shadow.png'</pre></div><div data-bbox="198 584 216 599" data-label="Text"><pre>},</pre></div><div data-bbox="198 612 269 627" data-label="Text"><pre>Ciudad: {</pre></div><div data-bbox="213 641 687 656" data-label="Text"><pre>icon: 'http://labs.google.com/ridefinder/images/mm_20_red.png',</pre></div><div data-bbox="213 669 740 685" data-label="Text"><pre>shadow: 'http://labs.google.com/ridefinder/images/mm_20_shadow.png'</pre></div><div data-bbox="198 698 213 713" data-label="Text"><pre>}</pre></div><div data-bbox="183 727 202 741" data-label="Text"><pre>};</pre></div><div data-bbox="183 782 343 798" data-label="Text"><pre>function load_map() {</pre></div><div data-bbox="198 810 756 826" data-label="Text"><pre>var map = new google.maps.Map(document.getElementById("map_canvas"), {</pre></div><div data-bbox="213 839 289 854" data-label="Text"><pre>zoom: 15,</pre></div><div data-bbox="213 867 377 883" data-label="Text"><pre>mapTypeId: 'roadmap'</pre></div><div data-bbox="138 924 222 940" data-label="Page-Footer">TOURALIA</div><div data-bbox="922 913 960 933" data-label="Page-Footer">10</div><div data-bbox="938 944 960 963" data-label="Page-Footer">6</div>
```

```
});  
  
var infoWindow = new google.maps.InfoWindow;  
  
// Change this depending on the name of your PHP file  
downloadUrl("index.php?p=geo_markers&id={$hotel_id}", function(data) {  
    var xml = data.responseXML;  
    var markers = xml.documentElement.getElementsByTagName("marker");  
  
    if (markers.length > 0) {  
        var name = markers[0].getAttribute("name");  
        var address = markers[0].getAttribute("address");  
        var type = markers[0].getAttribute("type");  
        var latitude = markers[0].getAttribute("lat");  
        var longitude = markers[0].getAttribute("lng");  
  
        var point = new google.maps.LatLng(  
            parseFloat(latitude),  
            parseFloat(longitude));  
  
        var html = "<b><a href=\"http://maps.google.es/maps?q=" +  
            escape(name + '@' + parseFloat(latitude) + "," + parseFloat(longitude))  
            + "\" target=\"_blank\">" + name + "</a></b> <br/>" + address;  
  
        var icon = customIcons[type] || {};  
  
        var marker = new google.maps.Marker({  
            map: map,  
            position: point,  
            icon: icon.icon,  
            shadow: icon.shadow  
        });  
  
        bindInfoWindow(marker, map, infoWindow, html);  
    }  
});
```



```
        map.setCenter(new google.maps.LatLng(
            parseFloat(latitude),
            parseFloat(longitude)));
    }
});
}

function bindInfoWindow(marker, map, infoWindow, html) {
    google.maps.event.addListener(marker, 'click', function() {
        infoWindow.setContent(html);
        infoWindow.open(map, marker);
    });
}

function downloadUrl(url, callback) {
    var request = window.ActiveXObject ?
        new ActiveXObject('Microsoft.XMLHTTP') :
        new XMLHttpRequest;

    request.onreadystatechange = function() {
        if (request.readyState == 4) {
            request.onreadystatechange = doNothing;
            callback(request, request.status);
        }
    };

    request.open('GET', url, true);
```



```
        request.send(null);
    }

    function doNothing() {}

    //]]>
</script>
</head>
<body onload="load_map()">

<!-- Header -->
<header style="background-image:url(tpl/img/placeholders/1280x1024/{"$bg_image"});">

<div class="container_12">

<!-- Title and navigation panel -->
<div id="panel" class="grid_12">

<!-- Title -->
<h1><a href="index.php">{"$h1_title}</a></h1>

<!-- Navigation -->
<nav>
    {include="navigation"}

<!-- Search -->
    {include="search" cache="3600"}
```

```
</nav>

</div>

</div>

<!-- Heading -->
<h2>{$h2_title}</h2>

</header>

<!-- Main content -->
<div class="container_12">

<!-- Description -->
<section class="description results_wide grid_12">

    <a href="tpl/{$hotel_img_large_src}" class="main_image fancybox"></a>

    <div>
        <span>
            <span class="boxed"><a
href="index.php?p=buscar_hotel&location={$hotel_localizacion}">{$hotel_localizacion}</a></span>
        </span>
        <span>
            <span class="stars boxed">
                {$hotel_categoria}
            </span>
        </span>
    </div>
</div>
</div>
</div>
```

```
</span>

</span>

<span>{$hotel_regimen}</span>

</div>

<p>{$hotel_descripcion}</p>

</section>

<!-- Image gallery -->

<section class="gallery grid_12">

  <!-- Slider navigation -->

  <nav class="slider_nav">

    <a href="index.php#" class="left">&nbsp;</a>

    <a href="index.php#" class="right">&nbsp;</a>

  </nav>

  <!-- Slider -->

  <div class="slider_wrapper">

    <!-- Slider content -->

    <div class="slider_content">

      {loop="hotel_imagenes_array"}

      <a href="tpl/{$value.img_large_src}">

        <img src "{$value.img_small_src}" height="150" width="110" alt="" />

      </a>

    </div>

  </div>

</section>
```

```
        {/loop}

    </div>

</div>

</section>

<div class="clearfix"></div>
<hr class="dashed grid_12" />

<!-- Reservation form -->
<section class="search_box sidebar grid_4">

    <h2>
        <span data-form="find_hotel">Hacer una reserva</span>
    </h2>

    <!-- Find a hotel form -->
    <form action="index.php" data-form="find_hotel" class="black">

        <div class="half">
            <label>Entrada</label>

            <input type="text" name="entrada" class="date" value="{ $fecha_entrada }" />
        </div>

        <div class="half last">
            <label>Salida</label>
```



```
<input type="text" name="salida" class="date" value="{ $fecha_salida}" />

</div>

<div class="full">

  <label>R&eacute;gimen</label>

  <input type="text" name="regimen" value="Desayuno" />

</div>

<div class="half">

  <label>Hu&eacute;spedes</label>

  <input type="text" name="huespedes" value="{ $huespedes}" />

</div>

<div class="half last">

  <label>Hab. individuales</label>

  <input type="text" name="hab_ind" value="0" />

</div>

<div class="half">

  <label>Hab. dobles</label>

  <input type="text" name="hab_dob" value="{ $habitaciones}" />

</div>

<div class="half last">

  <label>Hab. triples</label>

  <input type="text" name="hab_trip" value="0" />

</div>
```

```
<div class="full">

  <label>Número de tarjeta</label>

  <input type="text" name="num_tarjeta" value="" />

</div>

<div class="half">

  <label>Importe (EUR)</label>

  <input type="text" name="importe" value="" />

</div>

<div class="half last">

  <label>Importe con IVA (EUR)</label>

  <input type="text" name="importe_iva" value="" />

</div>

<input type="hidden" name="p" value="reserva" />

<input type="hidden" name="tipo_reserva" value="1" />

<input type="hidden" name="id_producto" value="{\$id_producto}" />

<input type="submit" value="Reservar" name="Reservar" />

</form>

</section>
```

```
<!-- Map -->
```

```
<section class="map grid_4">
```

```
  <div id="map_canvas" style="width:300px; height:250px"></div>
```

```
</section>
```

```
<!-- Similar hotels -->
```

```
<section class="similar_hotels grid_4">
```

```
  <h2 class="section_heading">Hoteles similares</h2>
```

```
  <ul>
```

```
    {loop="similar_hotels_array"}
```

```
    <li>
```

```
      <a href="index.php?p=hotel&id={$value.id}" class="thumb"></a>
```

```
      <h3><a href="index.php?p=hotel&id={$value.id}">{$value.nombre}</a></h3>
```

```
      <span class="stars">
```

```
        {$value.category}
```

```
      </span>
```

```
      <div>
```

```
        <span><a  
href="index.php?p=buscar_hotel&location={$value.location}">{$value.location}</a></span>
```

```
        <span><strong>{$value.precio_total} &euro;</strong> / {$value.noches}  
noches</span>
```

```
      </div>
```

```
    </li>
```

```
  {/loop}
```

```
</ul>
```



```
</section>

</div>

<!-- Footer -->

<footer><div class="container_12">

    {include="footer" cache="3600"}

</div></footer>

<!-- Google Analytics -->

<script type="text/javascript">

    var _gaq = _gaq || [];

    _gaq.push(['_setAccount', '{$google_analytics_id}']); // Set your Google Analytics ID here

    _gaq.push(['_trackPageview']);

    (function() {

        var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;

        ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-

analytics.com/ga.js';

        var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);

    })();

</script>

</body>

</html>
```

Ajax_hoteles.php

```
<?php

require_once 'forms/Plantilla.php';

require_once 'forms/UIUtils.php';

require_once 'sistema/AgenciaViajes.php';

require_once 'sistema/Imagen.php';

class AjaxHoteles extends Plantilla {

    function __construct($page, $sec_session) {

        parent::__construct($page, $sec_session);

    }

    function Load() {

        sleep( 2 );

        $max_num_rows = ( isset($_GET['per_page']) ) ? $_GET['per_page'] : 18446744073709551615;

        $offset = ( isset($_GET['start_page']) ) ? ( $_GET['start_page'] - 1 ) * $max_num_rows : 0;

        // results

        $agencia = new AgenciaViajes();

        if ( isset( $_SESSION['buscar_hotel_destino'] )

            && isset( $_SESSION['buscar_hotel_entrada'] )

            && isset( $_SESSION['buscar_hotel_salida'] )

            && isset( $_SESSION['buscar_hotel_habitaciones'] )

            && isset( $_SESSION['buscar_hotel_huespedes'] ) ) {

            $hoteles = $agencia->searchHoteles(

                $_SESSION['buscar_hotel_destino'],
```

```
    $_SESSION['buscar_hotel_entrada'],
    $_SESSION['buscar_hotel_salida'],
    $_SESSION['buscar_hotel_habitaciones'],
    $_SESSION['buscar_hotel_huespedes'],
    $offset, $max_num_rows);
} else {
    $hoteles = $agencia->getHoteles( $offset, $max_num_rows );
}
$res_count = 0;
$res_array = array();
foreach ($hoteles as $hotel) {
    //$imagen = array_shift( array_values( $hotel->imagenes ) );
    $imagen = Imagen::getBySize($hotel->imagenes, 200, 170);
    $res_array[] = array(
        'id' => $hotel->idHotel,
        'name' => UIUtils::uiEncode( $hotel->nombre ),
        'img_src' => $imagen->src,
        'img_height' => $imagen->height,
        'img_width' => $imagen->width,
        'price' => $hotel->precio_AD,
        'location' => UIUtils::uiEncode( $hotel->ciudad.', '.$hotel->pais ),
        'category' => Categoria::getCategory( $hotel->categoria ),
        'regimen' => UIUtils::uiEncode( $hotel->regimen ),
        'description' => UIUtils::uiEncode( $hotel->descripcion ),
    );
    $res_count++;
}
```



```
$this->tpl->assign("hotel_results", $res_array);

// create page

$html = $this->tpl->draw( TPlantilla::ResultadosHotel, $return_string = true );

return $html;

}

}

?>
```

Anexo E

Detalle de implementación

```
<!doctype html>

<!--[if lt IE 7]> <html class="no-js ie6 oldie" lang="en"> <![endif]-->

<!--[if IE 7]> <html class="no-js ie7 oldie" lang="en"> <![endif]-->

<!--[if IE 8]> <html class="no-js ie8 oldie" lang="en"> <![endif]-->

<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->

<head>

    <meta charset="utf-8">

    <title>Travel - Premium HTML Template</title>

    <meta name="description" content="">

    <meta name="author" content="">

    <!-- Load CSS -->

    <link rel="stylesheet" href="css/style.css">

    <link rel="stylesheet" href="fancybox/jquery.fancybox-1.3.4.css">

    <link rel="stylesheet" href="css/smoothness/jquery-ui-1.8.16.custom.css">

    <!-- Page icon -->

    <link rel="shortcut icon" href="favicon.png">

    <!-- Load Modernizr -->

    <script src="js/libs/modernizr-2.0.min.js"></script>

    <!-- Load JavaScript -->
```



```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js"></script>

<script>window.jQuery || document.write('<script src="js/libs/jquery-
1.6.2.min.js"></script>')</script>

<script src="js/libs/jquery-ui-1.8.16.custom.min.js"></script>

<script src="js/script.js"></script>

<script src="fancybox/jquery.fancybox-1.3.4.pack.js"></script>

<script src="http://travel.equiet.sk/js/datepicker.js"></script>

<script src="http://maps.googleapis.com/maps/api/js?sensor=false"></script>

</head>

<body>

<!-- Header -->

<header style="background-image:url(img/placeholders/1280x1024/12.jpg);">

    <div class="container_12">

        <!-- Title and navigation panel -->

        <div id="panel" class="grid_12">

            <!-- Title -->

            <h1><a href="index.html">Travel</a></h1>

            <!-- Navigation -->

            <nav>

                <ul>

                    <li>

                        <a href="browse.html">Browse</a>


```



```

        <ul>
            <li><a href="browse.html">Browse
all</a></li>
            <li><a
href="browse_hotels.html">Browse hotels</a></li>
            <li><a href="hotel.html">Hotel</a></li>
            <li><a href="trip.html">Trip</a></li>
        </ul>
    </li>
    <li>
        <a href="faq.html">Pages</a>
        <ul>
            <li><a href="faq.html">FAQ</a></li>
            <li><a href="forms.html">Forms</a></li>
        </ul>
    </li>
    <li>
        <a href="blog.html">Blog</a>
        <ul>
            <li><a href="blog.html">Blog</a></li>
            <li><a
href="blogpost.html">Blogpost</a></li>
        </ul>
    </li>
    <li>
        <a href="contact.html">Contact</a>
    </li>
</ul>

```

```
        <!-- Search -->
        <form action="index.html#" class="black">
            <input name="search" type="text"
placeholder="Search..." />
            <input type="submit" />
        </form>
    </nav>

</div>

</div>

<!-- Slider navigation -->
<nav class="slider_nav">
    <a href="index.html#" class="left">&nbsp;</a>
    <a href="index.html#" class="right">&nbsp;</a>
</nav>

<!-- Slider -->
<div class="slider_wrapper">

    <!-- Slider content -->
    <ul class="homepage_slider">

        <!-- First slide -->
        <li>

            <h2><a href="trip.html">The Indonesia Expedition from
```

```
<strong>799 €</strong></a></h2>

    <p>Ubud, Uluwatu, Batur, Besakih and Tenganan</p>

</li>

<!-- Second slide -->

<li>

    <h2><a href="hotel.html">A wonderful week in Singapore from
<strong>999 €</strong></a></h2>

    <p>With accomodation in Marina Bay Sands</p>

</li>

</ul>

<div class="clearfix"></div>

</div>

</header>

<!-- Main content -->

<div class="container_12">

    <!-- Recommended trips -->

    <ul class="results">

        <li class="short grid_3">

            <a href="hotel.html"></a>

            <h3><a href="hotel.html">Marina Bay Sands</a></h3>
```

```
<span class="stars">

</span>

<div>

    <span><a href="index.html#">Singapore</a></span>

    <span><strong>1 899 €</strong> / 10 days</span>

</div>

</li>

<li class="short grid_3">

    <a href="hotel.html"></a>

    <h3><a href="hotel.html">Hotel Palma</a></h3>

    <span class="stars">

    </span>

    <div>

        <span><a href="index.html#">Mallorca</a></span>

        <span><strong>1 899 €</strong> / 10 days</span>

    </div>

</li>
```

```
</li>

<li class="short grid_3">
  <a href="hotel.html"></a>

  <h3><a href="hotel.html">Holiday Inn</a></h3>

  <span class="stars">
    
    
    
    
    
  </span>

  <div>
    <span><a href="index.html#">Cannes</a></span>
    <span><strong>1 899 €</strong> / 10 days</span>
  </div>
</li>

<li class="short grid_3">
  <a href="hotel.html"></a>

  <h3><a href="hotel.html">Hotel Grand</a></h3>

  <span class="stars">
    
    
    
    
  </span>
</li>
```

```


</span>

<div>

  <span><a href="index.html#">Singapore</a></span>

  <span><strong>1 899 €</strong> / 10 days</span>

</div>

</li>

</ul>

<div class="clearfix"></div>

<hr class="dashed grid_12" />

<!-- Search form and last minute -->

<section class="search_lm grid_12">

  <!-- Ticket search form -->

  <section class="search sidebar">

    <h2>

      <span data-form="find_trip" class="selected">Find a trip</span>

      <span data-form="find_hotel">Find a hotel</span>

      <span data-form="tickets">Tickets</span>

    </h2>

    <!-- Find a trip form -->
```



```
<form action="index.html#" data-form="find_trip" class="black">

    <label>Destination</label>

    <input type="text" name="destination" value="All" />

    <div class="half">

        <label>Transportation</label>

        <input type="text" name="transportation" value="Plane"

/>

    </div>

    <div class="half last">

        <label>Date</label>

        <input type="text" name="date" class="date"

value="11/23/2011" />

    </div>

    <div class="half">

        <label>Guests</label>

        <input type="text" name="guests" value="2" />

    </div>

    <div class="half last">

        <label>Rooms</label>

        <input type="text" name="rooms" value="1" />

    </div>
```




```
<input type="submit" value="Search">

</form>

<!-- Find a hotel form -->
<form action="index.html#" data-form="find_hotel" class="black"
style="display:none;">

    <label>Destination</label>

    <input type="text" name="destination" value="All" />

    <div class="half">

        <label>Check in</label>

        <input type="text" name="check_in" class="date"
value="11/23/2011" />

    </div>

    <div class="half last">

        <label>Check out</label>

        <input type="text" name="check_out" class="date"
value="11/23/2011" />

    </div>

    <div class="half">

        <label>Guests</label>

        <input type="text" name="guests" value="2" />

    </div>
```



```
<div class="half last">

    <label>Rooms</label>

    <input type="text" name="rooms" value="1" />

</div>

<input type="submit" value="Search">

</form>

<!-- Tickets form -->

<form action="index.html#" data-form="tickets" class="black"
style="display:none;">

    <div class="half">

        <label>From</label>

        <input type="text" name="from" value="JFK" />

    </div>

    <div class="half last">

        <label>To</label>

        <input type="text" name="to" value="SFO" />

    </div>

    <div class="half">

        <label>Leaving date</label>

        <input type="text" name="leaving-date" class="date"
value="11/23/2011" />
```



```
</div>

<div class="half last">
    <label>Return date</label>
    <input type="text" name="return-date" class="date"
value="11/23/2011" />
</div>

<div class="half">
    <label>Adults</label>
    <input type="text" name="adults" value="2" />
</div>

<div class="half last">
    <label>Children</label>
    <input type="text" name="children" value="0" />
</div>

<input type="submit" value="Search">

</form>
</section>

<!-- Last minute -->
<section class="last_minute">
    <table>
```



```
<tr class="header">
  <th>Destination</th>
  <th>Hotel</th>
  <th>Departure</th>
  <th>Length</th>
  <th>Price</th>
</tr>
<tr>
  <td><a href="browse.html">Spain</a></td>
  <td>Hotel Grand 5*</td>
  <td>13 Oct - 25 Oct</td>
  <td>12 nights</td>
  <td><del>1 099 €</del> 899 €</td>
</tr>
<tr>
  <td><a href="browse.html">Greece</a></td>
  <td>Hotel Palma 4*</td>
  <td>15 Oct - 25 Oct</td>
  <td>10 nights</td>
  <td><del>1 099 €</del> 749 €</td>
</tr>
<tr>
  <td><a href="browse.html">Italy</a></td>
  <td>Holiday Inn 4*</td>
  <td>15 Oct - 25 Oct</td>
  <td>10 nights</td>
  <td><del>1 099 €</del> 799 €</td>
```

```
</tr>
```

```
<tr>
```

```
<td><a href="browse.html">Egypt</a></td>
```

```
<td>Beach Resort 5*</td>
```

```
<td>18 Oct - 28 Oct</td>
```

```
<td>9 nights</td>
```

```
<td><del>1 099 €</del> 799 €</td>
```

```
</tr>
```

```
<tr>
```

```
<td><a href="browse.html">United Kingdom</a></td>
```

```
<td>Spa & Golf Resort 4*</td>
```

```
<td>18 Oct - 28 Oct</td>
```

```
<td>9 nights</td>
```

```
<td><del>1 099 €</del> 749 €</td>
```

```
</tr>
```

```
<tr>
```

```
<td><a href="browse.html">Thailand</a></td>
```

```
<td>Wellness Resort 5*</td>
```

```
<td>20 Oct - 29 Oct</td>
```

```
<td>9 nights</td>
```

```
<td><del>1 099 €</del> 849 €</td>
```

```
</tr>
```

```
<tr>
```

```
<td><a href="browse.html">Spain</a></td>
```

```
<td>Hotel Grand 5*</td>
```

```
<td>25 Oct - 05 Nov</td>
```

```
<td>11 nights</td>
```

```
<td><del>1 099 €</del> 899 €</td>

</tr>

</table>

</section>

</section>

<div class="clearfix"></div>

<hr class="dashed grid_12" />

<!-- Latest blog articles -->

<section class="latest_articles">

    <article class="grid_4">

        <a href="blogpost.html"></a>

        <h2><a href="blogpost.html">Around the world</a></h2>

        <div class="info">

            by <strong>John Doe</strong>

            <strong>8</strong> comments

            <strong>Nov 04</strong>

        </div>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam
interdum nunc at mauris condimentum rhoncus. Proin fermentum ligula vitae elit laoreet a
ullamcorper lorem cursus. Suspendisse malesuada nisl nec magna fringilla ornare. Clabel urabitur
molestie ligula a urna hendrerit quis porttitor enim ornare.</p>

    </article>
```

```
<article class="grid_4">

  <a href="blogpost.html"></a>

  <h2><a href="blogpost.html">Letters from Africa</a></h2>

  <div class="info">

    by <strong>John Doe</strong>

    <strong>8</strong> comments

    <strong>Sep 16</strong>

  </div>

  <p>Proin fermentum ligula vitae elit laoreet a ullamcorper lorem cursus.
Suspendisse malesuada nisl nec magna fringilla ornare. Curabitur molestie ligula a urna hendrerit
quis porttitor enim ornare. Nullam leo enim, sollicitudin semper venenatis at, aliquet et nisi. Aliquam
placemat aliquet feugiat. Curabitur molestie ligula.</p>

</article>

<article class="grid_4">

  <a href="blogpost.html"></a>

  <h2><a href="blogpost.html">Wonders of Indonesia</a></h2>

  <div class="info">

    by <strong>John Doe</strong>

    <strong>8</strong> comments

    <strong>Aug 15</strong>

  </div>

  <p>Nullam interdum nunc at mauris condimentum rhoncus. Proin
```



fermentum ligula vitae elit laoreet a ullamcorper lorem cursus. Suspendisse malesuada nisl nec magna fringilla ornare. Curabitur molestie ligula a urna hendrerit quis porttitor enim ornare. Nullam leo enim, sollicitudin semper venenatis at, aliquet et nisi.</p>

</article>

</section>

</div>

<!-- Footer -->

<footer><div class="container_12">

<nav class="grid_8">

Home

Catalogue

Blog

Contact

FAQ

</nav>

<p class="address grid_4">

Travel Agency Inc.

123 Wall Street , New York

contact@travelagency.com

</p>

<p class="copyright grid_8">



© 2011 Travel Agency

</p>

</div></footer>

<!-- Google Analytics -->

<script type="text/javascript">

```
var _gaq = _gaq || [];
```

```
_gaq.push(['_setAccount', 'UA-2529322-12']); // Set your Google Analytics ID here
```

```
_gaq.push(['_trackPageview']);
```

```
(function() {
```

```
var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
```

```
ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +  
'google-analytics.com/ga.js';
```

```
var s = document.getElementsByTagName('script')[0];  
s.parentNode.insertBefore(ga, s);
```

```
})();
```

</script>

</body>

</html>