



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

CREACIÓN DE UNA PÁGINA WEB PARA LA RECOMENDACIÓN DE RUTAS TURÍSTICAS PLANIFICADAS EN LA CIUDAD DE VALENCIA

Proyecto Final de Carrera
Ingeniería Técnica
Informática de Gestión

Autor: Javier Crespo Tapia
Directora: Laura Sebastia Tarín
Fecha: Abril de 2013

Agradecimientos

A mi directora de proyecto, Dra. Laura Sebastia Tarín, por proporcionarme todos los medios disponibles a su alcance para facilitarme la labor de creación de esta interfaz gráfica para la aplicación de recomendación de rutas por la ciudad de Valencia mediante tecnología de web, también por leerse detenidamente cada capítulo de dicha memoria y por su aporte de ideas y mejoras durante todo el proceso de diseño e implementación.

Gracias a mis padres Higinio y Teresa y a mi hermano Fernando por vuestra confianza y apoyo, y por otorgarme todos los medios materiales necesarios para desarrollar dicho proyecto.

Gracias a mi compañera sentimental Tania por ser un motivo diario de superación y mejora y por su ayuda en el diseño de la página web que aquí presento.

Y por último agradecimientos a todos aquellos profesores que se han esforzado por inculcarme todos los conocimientos que utilizo para la realización de dicho proyecto.

Javier Crespo Tapia

1.	Índice	6
2.	Introducción	7
	2.1 Presentación	7
	2.2 Objetivos	8
	2.3 Organización de la memoria	9
	2.3.1 Introducción	9
	2.3.2 Antecedentes	9
	2.3.3 Diseño	9
	2.3.4 Arquitectura	9
	2.3.5 Conclusiones	9
3.	Antecedentes	10
	3.1 Sistema de recomendación	10
	3.1.1 Definición del sistema de recomendación	10
	3.1.2 Descripción funcionamiento y arquitectura del Sistema de recomendación.	11
	3.2 BD : esquema	13
4.	Diseño de la web	14
	4.1 Interfaz / Manual usuario	14
	4.2 Registro y acceso	16
	4.3 Perfil de usuario	20
	4.4 Visita actual, recomendación y ruta	22
5.	Arquitectura/Implementación de la web	29
	5.1 Detalles técnicos de la implementación	30
	5.1.1 Plantilla HTML/CSS	31
	5.1.2 Estructura y codificación de las paginas JSP	35
	5.2 Utilización del api de Google Maps	46
	5.3 Programas utilizados	48
6.	Conclusiones	52
7.	Bibliografía	53

2.1 Presentación

Según la Organización Mundial del Turismo, el turismo comprende las actividades que realizan las personas durante sus viajes y estancias en lugares distintos al de su entorno habitual, por un período consecutivo inferior a un año y mayor a un día, con fines de ocio, por negocios o por otros motivos.

El turismo es uno de los principales motores económicos de la Comunidad Valenciana. Supone un 13% del PIB, genera el 12,8% del total del empleo y representa el 12,4% del total de la recaudación tributaria en la comunidad.

Somos el tercer destino de preferencia para el turismo nacional y el quinto destino para el turismo extranjero. En total nos visitan una media de 23 millones de turistas anuales, de los cuales 5,7 millones lo hacen desde el extranjero y los 17,3 millones restantes lo hacen desde dentro de nuestras fronteras.

El 87% de los visitantes que recibimos lo hacen sin recomendación de visita, y su estancia media es de 5,7 días en la ciudad de Valencia. Este es el perfil de gente para el que se desarrolla este proyecto.

Concretamente, este proyecto consiste en la creación de un entorno web como interfaz de un sistema de recomendación de rutas por la ciudad de Valencia. Este sistema de recomendación ha sido desarrollado por el Grupo de Tecnología Informática – Inteligencia Artificial del Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia. A grandes rasgos, este sistema de recomendación ofrece una ruta turística personalizada al perfil del usuario.

El proyecto utiliza HTML, CSS, Java Script y jQuery para darle una apariencia visual agradable, sencilla e intuitiva al visitante, y también tecnologías JSP, Servlets, Java y MySQL para registrar la información y ofrecer buenas recomendaciones.

2.2 Objetivos

Como se introduce en el apartado anterior, uno de los principales pilares económicos de la ciudad de Valencia se basa en el número de turistas que recibe. Por ello se realiza la siguiente página web, como ayuda a esta gente, para que puedan conocer los lugares que mejor se adapten a su perfil.

Este documento describe el trabajo realizado en el proyecto final de carrera de Ingeniería Técnica en Informática de Gestión. El proyecto, como se ha citado anteriormente, consiste en el desarrollo de un sitio web capaz de mediante la información obtenida de sus visitantes, crear una ruta por Valencia. El objetivo de la web es poder recomendar a sus usuarios, mediante la aplicación de recomendación, una ruta para visitar diferentes lugares y monumentos de la Ciudad del Turia.

Esta ruta se calcula utilizando la información del registro y del perfil del usuario para hacerla lo más personalizada posible, y se intenta plasmar en la pantalla del ordenador de una manera ordenada y clara.

El usuario puede moverse cómodamente por los menús de la web, modificando su información cuando sea necesario para hacer que la recomendación sea lo más ajustada posible, además podrá visualizar los lugares y su situación en el mapa, antes obtener la ruta.

Tanto la interfaz web que se desarrolla en este proyecto como el sistema de recomendación forman parte de una aplicación que hemos denominado Valencia Custom Tours (VTC)

Es importante recalcar que el presente trabajo se centra fundamentalmente en la tecnología de web y no en el recomendador. Concretamente, la realización del proyecto consiste en la creación de una página web que utilice una aplicación Java de recomendación y una base de datos para la visualización de rutas. Para ello se cuenta con una aplicación y una base de datos diseñada donde se irá insertando la información recogida por la web. Por tanto, otro de los objetivos principales de este proyecto es que todo el proceso de registro y recomendación se realice en un tiempo de carga mínimo, para que la navegación no se le haga pesada al usuario. Por ello se recurre a un entorno agradable pero sencillo y se reduce al mínimo posible los accesos de la página a la aplicación y a la base de datos.

El objeto del proyecto es intentar crear un sitio web optimizado para que los tiempos de carga sean lo mínimo posible y a la vez con una interfaz gráfica agradable al usuario con colores identificativos con la función que realiza. Para ello se crea una página web enlazada con una base de datos MySQL y aplicación Java.

La tecnología empleada para la codificación de la página web es el lenguaje Java (JSP y Servlets)

Todo ello se codifica en una maquina con Windows 7, con el servidor web Tomcat instalado y configurado en local y con el servidor de bases de datos MySQL Workbench también configurado en local. Con ello se pretende establecer un entorno real de ejecución.

2.3 Organización de la memoria

El presente documento se va a componer de varias partes bien diferenciadas que se explicarán en el presente apartado.

2.3.1 Introducción

En este apartado se explican los elementos de partida y enunciado del proyecto, así como las motivaciones y objetivos que se tienen que cumplir al finalizar el trabajo.

2.3.2 Los antecedentes

Aquí se van a exponer y explicar las herramientas de las que se dispone y que se tienen que adaptar al nuevo proyecto, en este caso son dos:

- El sistema de recomendación: Se trata de una aplicación escrita en Java, que se utiliza para calcular la recomendación.
- La base de datos: Se describirá el esquema de la BBDD utilizada por la web, y se explicarán de las tablas utilizadas para la recomendación.

2.3.3 Diseño de la web

En este apartado se va a describir la interfaz gráfica de la página web y se explicará su funcionamiento, el flujo de datos y las conexiones con la BBDD.

2.3.4 Arquitectura e implementación de la web

Éste es el penúltimo punto que se va a tratar en esta memoria, y uno de los más importantes. Aquí se hablará de los detalles técnicos de la implementación, de los lenguajes utilizados, las pequeñas mejoras usadas para hacer que la pagina sea lo más ligera posible, y diversos detalles técnicos.

2.3.5 Conclusiones

Y por último, como indica el título, la conclusión final del proyecto. Aquí se analizará si se han cumplido los objetivos del proyecto, se analizará el porqué, y se hará una valoración a nivel subjetiva del trabajo realizado

3.1 Sistemas de recomendación

El crecimiento de internet ha sido increíblemente acelerado, hay cantidades desorbitadas de datos casi sobre cualquier tópico, lo que hace de Internet la fuente más completa de información para casi todo. Sin embargo, hay más cantidad de películas, artículos, libros, programas de televisión, lugares turísticos para visitar entre otras cosas, de los que las personas podemos manejar efectivamente, con lo cual hay una sobrecarga de información hacia el usuario. El tiempo medio que emplea un usuario para realizar una búsqueda sobre el conjunto total de tópicos es demasiado, y los procesos de búsqueda tradicionales pueden ser infructíferos la mayoría de las veces o al menos con resultados poco exitosos de acuerdo con las preferencias del usuario.

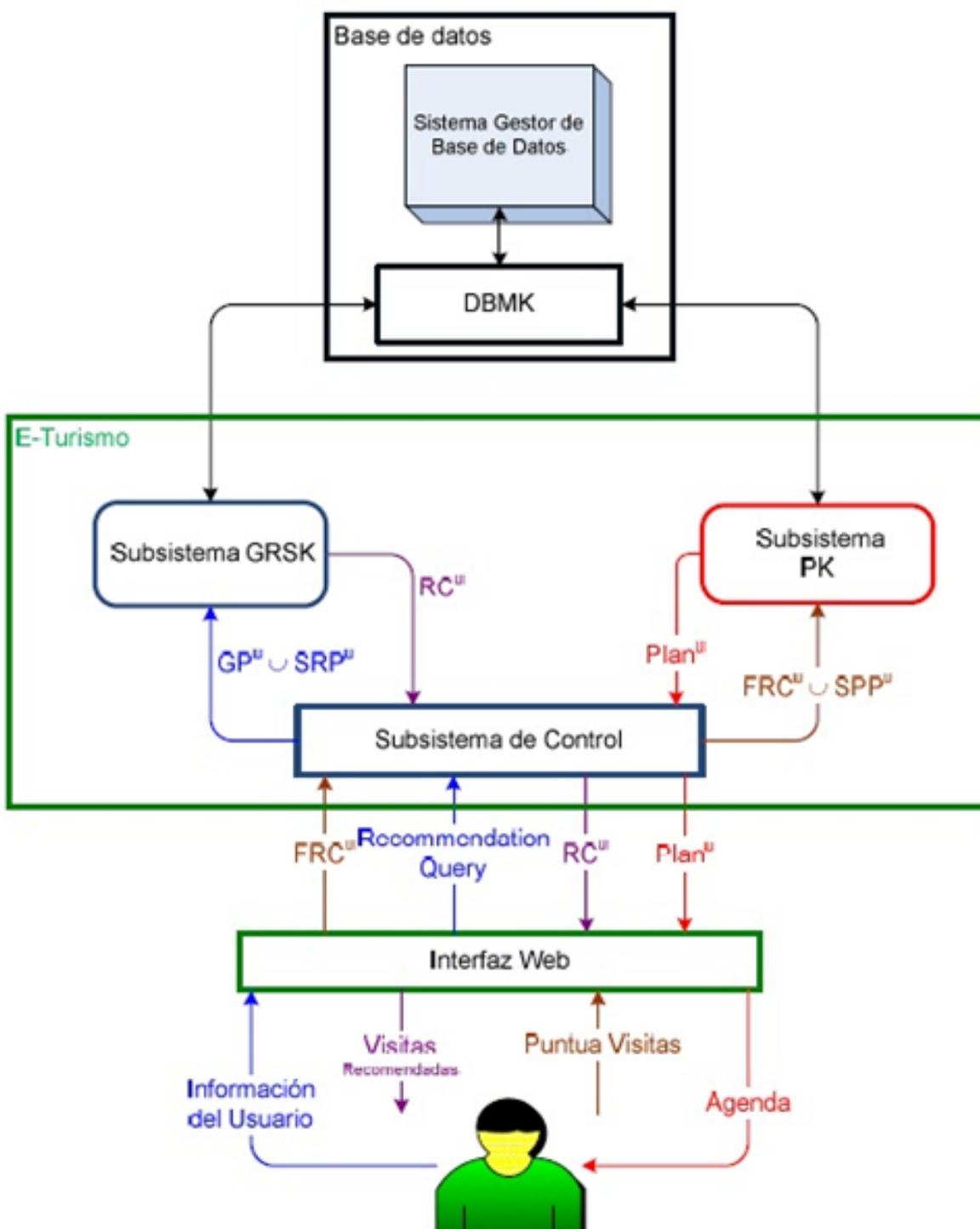
3.1.1 Definición del sistema de recomendación

Un Sistema Recomendador (SR de aquí en adelante) es una herramienta que ayuda al usuario a obtener la información que necesita de acuerdo a sus preferencias. Un SR (I) toma decisiones basadas en información disponible, (II) requiere un método sofisticado para filtrar información, y (III) generalmente se aplica en algún dominio completo, por ejemplo: películas, música, restaurantes, email, e-commerce, etc. Por tanto, los SRs tratan de ser un paso adelante en el contexto de la recuperación de información tradicional, que se da por palabras clave del tópico que se desea encontrar a través de un buscador (Google, Lycos, etc.). Estos buscadores (I) normalmente nos encuentran una información muy general y estática, es decir, se ofrece sistemáticamente la misma información a todos los usuarios independientemente del perfil de estos, y (II) muestran grandes volúmenes de información que el usuario debe procesar posteriormente para seleccionar aquellos elementos que pueden ser de su interés. Sin embargo, el objetivo de los SRs es recomendar o sugerir a los usuarios ítems o productos concretos (películas, programas de televisión, lugares turísticos para visitar, etc.) basándose en sus preferencias, es decir, ofrecer una recomendación personalizada acorde con las preferencias del usuario en particular. Por ejemplo, el viajero gastronómico situará en primer lugar sus preferencias culinarias, para luego organizar los lugares o monumentos a visitar alrededor del mismo. El usuario que viaja con niños evitará visitar museos durante un largo periodo de tiempo e incluirá en su viaje la visita a algún jardín o parque de atracciones.

3.1.2 Descripción, funcionamiento y arquitectura

El SR de rutas turísticas planificadas en la ciudad de Valencia (VCT) tiene como objetivo principal calcular un plan turístico o de ocio para un usuario, teniendo en cuenta sus preferencias. El sistema no resuelve el problema de viajar desde un lugar específico a otro, sino que se centra en recomendar una lista de las actividades que un turista puede realizar en la ciudad de Valencia.

Inicialmente el usuario debe registrarse en el sistema con sus datos personales y preferencias generales. Con esta información el sistema construye el perfil del usuario, a partir del que se planifican nuevas visitas a la ciudad de Valencia, y que se va actualizando con el historial de sus visitas anteriores.



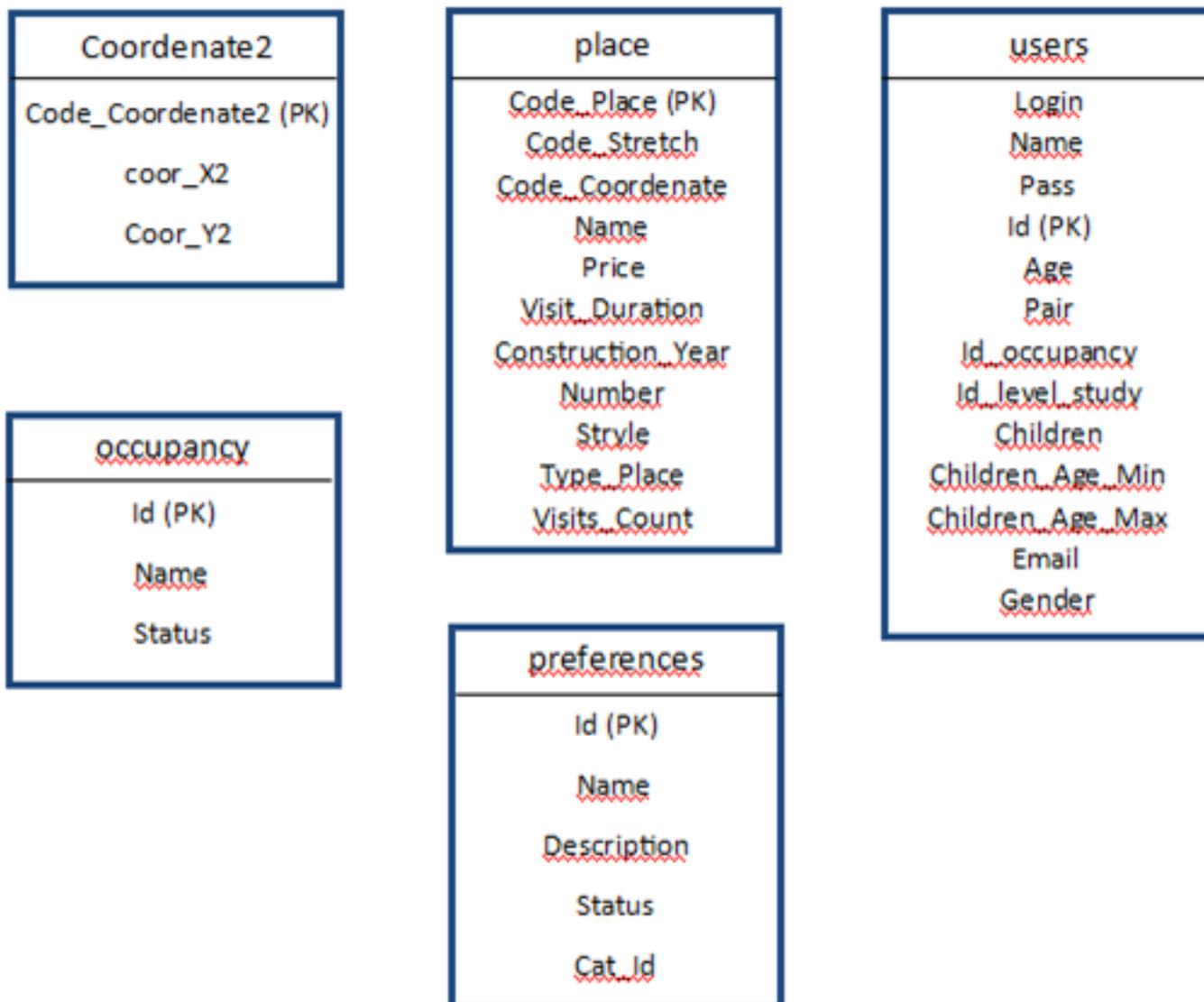
Para planificar una nueva visita el usuario define, con ayuda de sus preferencias generales, unas preferencias específicas y unos datos básicos de la visita actual, como la duración, es decir, tiempo del que dispone el usuario, si va con niños o solo, etc. A partir de esta información el sistema muestra al usuario una lista de actividades recomendadas. Esta lista de actividades recomendadas son las que se consideran más interesantes para el usuario.

El usuario debe seleccionar de esta lista de actividades, aquellas en las que está o no interesado y por consiguiente desea visitar (seleccionar lugares a visitar). Con estas restricciones y la duración de la visita (seleccionadas anteriormente) el sistema determina las actividades que le interesan al usuario y cuándo puede realizarlas, planificando de este modo la visita a los lugares turísticos seleccionados.

El SR de rutas turísticas planificadas en la ciudad de Valencia se compone de tres subsistemas: El Núcleo Generalista del Sistema Recomendador (Generalist Recommender System Kernel - GRSK), el Núcleo Planificador (Planning Kernel - PK) y el Núcleo Gestor de la Base de Datos (DBMK). GRSK es un módulo recomendador de propósito general mientras que el resto de subsistemas dependen de la aplicación (turismo en la ciudad de Valencia). Estos subsistemas son coordinados por un cuarto módulo: el subsistema de control.

El subsistema encargado de construir el perfil del usuario es el de Control, es el núcleo organizativo de VCT. Funciona como intermediario entre la interfazWeb y los demás subsistemas. Este subsistema inicia la ejecución de los demás subsistemas y centraliza el intercambio de información. Esto incluye, convertir los datos del usuario en el formato apropiado para la consulta, mostrar al usuario la lista de actividades recomendadas, registrar las actividades que son finalmente seleccionadas por el usuario, y, posteriormente, mostrar al usuario el plan programado para la visita turística.

3.2 Esquema de la BBDD



El sistema de recomendación utiliza para calcular la ruta, información almacenada en la base de datos. Parte de esta información que utiliza de la BBDD se introduce desde la web.

A continuación se explicará brevemente cada una de las tablas utilizadas por la web:

- Coordinate2: Contiene la información de las coordenadas latitud y longitud en formato DMS.
- Occupancy: Contiene la información de las diferentes profesiones.
- Place: Contiene toda la información de los lugares a visitar.
- Preferences: Contiene la descripción de los estilos y tipos de los lugares.
- Users: Contiene toda la información sobre los usuarios.

En este apartado, se va a explicar las diferentes funcionalidades disponibles en la página web. Mediante capturas se explicará, pantalla por pantalla, las opciones y los pasos a seguir para obtener una buena recomendación y un buen uso de la aplicación.

4.1 Página principal



Fig 1. Página de inicio sin sesión iniciada

En la figura 1 se puede observar la pantalla principal o de bienvenida de la página web. La página se compone de un rectángulo central situado sobre un fondo blanco, en dicho rectángulo encontraremos la información durante todo el proceso de recomendación.

Se eligió dicho formato para que la página web fuera compatible con el máximo de resoluciones posibles, tanto en formato tradicional (800x600), como en formato panorámico (1280x800), la página se visualiza correctamente.

A continuación se describirá cada uno de los apartados de la página de inicio:

- **Logotipo:** Se encuentra en la parte superior izquierda de la web y se compone de las iniciales VCT (Valencia Custom Tours), se utilizan los colores azul y rojo, que junto con el amarillo de la barra de menú forman los colores de la Señera. Este logotipo aparece en todas las pantallas de la web y es un link hacia la página home que es la que podemos ver en la figura 1 y 4.

- **Validar usuario:** En la parte superior derecha encontramos dos recuadros “text inputs” uno encima de otro con el texto “Usuario” y “Contraseña”, y un recuadro rojo justo a su derecha. Es donde los usuarios previamente registrados acceden a los datos de su perfil, al pulsar encima de cualquiera de los dos recuadros el texto que contienen desaparece, mediante un efecto Java Script. Entonces el usuario introduce su nombre de usuario y contraseña y a continuación pulsa el botón rojo.

Si la validación ha tenido éxito la página quedará como la figura 4, desaparecerán los recuadros y aparecerá el avatar del usuario y la opción de ir a su perfil o de salir.

- **Nuevo usuario:** Justo en la parte derecha de la zona de identificación de usuario podemos encontrar “Nuevo usuario” (en rojo), este es un link que nos lleva a la página de registro (figura 2) y es el primer paso que se debe dar cuando no se está registrado en la web, ya que sólo los usuarios registrados pueden solicitar una recomendación.

- **Recuperar contraseña:** Debajo del link de “Nuevo usuario” podemos encontrar el link de “¿Contraseña?”, que se trata del acceso a la herramienta de recuperar contraseña (figura 3).

- **Recordarme:** Se trata de un “input check box” situado debajo del link de recuperar contraseña y sirve para almacenar cookies sobre la página, para que los usuarios no tengan que identificarse cada vez que accedan desde un ordenador de confianza.

- **Lugares:** Se trata de un div donde se van visualizando los últimos sitios visitados. Es un “Marquee” que es una herramienta HTML para que los textos se deslicen. Éstos se paran cuando se pone el ratón encima del recuadro.

- **Fondo del recuadro:** Se utiliza la tecnología jQuery y JavaScript para hacer un efecto de desvanecimiento de la imagen de fondo cada 5000ms y que en su lugar la siguiente imagen incremente su opacidad desde 0, dando un toque de dinamismo a la página.

Existen dos tipos de usuarios, los registrados y los no registrados. Los usuarios registrados deben identificarse introduciendo sus datos en los campos habilitados para ello y pulsando el botón rojo. En ese momento ya tendrán acceso a todo el proceso de recomendación.

Por el contrario los usuarios no registrados, tendrán que ir a la sección de registro y rellenar los campos requeridos. El proceso de registro sólo es necesario la primera vez que se utiliza la web.

4.2 Registro y acceso

VCT
VALENCIA CUSTOM TOURS

[Mapa web](#)

REGISTRARSE

Obten las mejores recomendaciones de sitios para visitar en la ciudad

Es gratis y cualquiera puede utilizarlo.
Por favor, introduce toda la información que le sea posible, para obtener una mejor recomendación de ruta, y no olvide que los campos marcados con (*) son obligatorios.

Usuario*
Password*
re-Password*
-- ¿Tienes hijos? --
Edad del hijo mayor
Edad del hijo menor

Nombre completo
Tu edad
E-mail
-- Seleccione Sexo --
-- Selec. Ocupacion --
-- ¿Tienes pareja? --

Registrarse

VTC 2012 [English Castellano](#)

Fig 2. Página de registro

En la figura 2 podemos visualizar la pantalla de registro. Visualmente esta pantalla es proporcionalmente igual que la de home, pero desaparece la parte de identificación de usuario e imágenes para dar paso a un formulario de registro. La mayoría de las páginas que componen esta web tiene la misma estructura, con el fin de dar continuidad.

A continuación introduciré una breve descripción de cada uno de los campos que completan el formulario:

- **Usuario:** En este campo se debe introducir el Nick que se utilizara cada vez que el usuario quiera acceder a la página.

Este campo pasa dos controles, uno de comprobación de que no esté vacío (figura 2B), ya que se trata de un campo obligatorio (*), y otro de control de que no se encuentre registrado ya en la base de datos.

- **Password:** Se trata de otro campo obligatorio del formulario y necesario junto con el de "Usuario" para realizar posteriormente la identificación del usuario. Se debe introducir una clave para posteriormente verificar la identidad.

- **Re-Password:** Este es sólo un campo de verificación, hay que repetir la clave introducida en el campo "Password". Su utilidad es evitar un error ortográfico a la hora de introducir la contraseña. Este campo pasa dos controles, uno que comprueba que está relleno y otro que comprueba que su contenido sea igual al del campo "Password".

REGISTRARSE

Obten las mejores recomendaciones de sitios para visitar en la ciudad



En cuanto se concluyan, puedes utilizarlo.

Debes introducir un usuario

Aceptar

Re-Password

¿Tienes hijos? --

Edad del hijo mayor

Edad del hijo menor

Registrarse

E-mail

Selecione Sexo --

Selec. Ocupacion --

¿Tienes pareja? --

Fig 2B. Página de registro con alert de error

- **Tienes hijos:** Se trata de un combobox, con las opciones Sí y No.
- **Edad del hijo mayor:** En este campo se introduce la edad del hijo mayor. No se guardara el valor en la BBDD si no se pone el campo “Tienes hijos” a Sí
- **Edad del hijo menor:** En este campo se introduce la edad del hijo menor. No se guardara el valor en la BBDD si no se pone el campo “Tienes hijos” a Sí
- **Nombre:** Se trata de un campo voluntario, como todos los que no tienen el asterisco, pero que se utiliza para hacer más personal la recomendación. En él hay que introducir el nombre acompañado de los apellidos.
- **Edad:** Otro campo voluntario en el que hay que introducir la edad
- **E-mail:** se trata de un campo voluntario, pero recomendado ya que se utiliza para poder recuperar al contraseña en caso de olvido.
- **Sexo:** Se trata de un combobox con las opciones “Masculino” y “Femenino”
- **Selección de ocupación:** Se trata de un combobox que se carga directamente de la tabla “occupancy” de la base de datos y muestra las diferentes profesiones.
- **¿Tienes pareja?:** Es el último campo del formulario, se trata de un combobox con las opciones Sí y No.

La descripción de los campos del formulario se encuentra dentro de cada casilla de texto. Mediante funciones JavaScript se consigue que cuando se pulsa encima del campo que se desea rellenar, el texto que éste contenía desaparezca, y en el caso de que se deje el campo vacío aparezca otra vez la descripción.

Una vez que el usuario ha introducido todos los datos correctamente y presiona el botón de “Registrarse”, se redirige automáticamente a Home (figura 4)



Fig 3. Página de envío de contraseña

En la figura 3 se puede observar la pantalla de recuperación de contraseña. Continúa en la línea de las pantallas vistas hasta ahora. El único control que se aprecia es un Input, en el hay que introducir el usuario que se utilizó para registrarse en la web y pinchar en el botón “Enviar”.

La web ejecutará una aplicación Java que mandará un correo a la dirección que el usuario utilizó para registrarse. De los datos técnicos de esta aplicación se hablará en el siguiente capítulo de este documento.

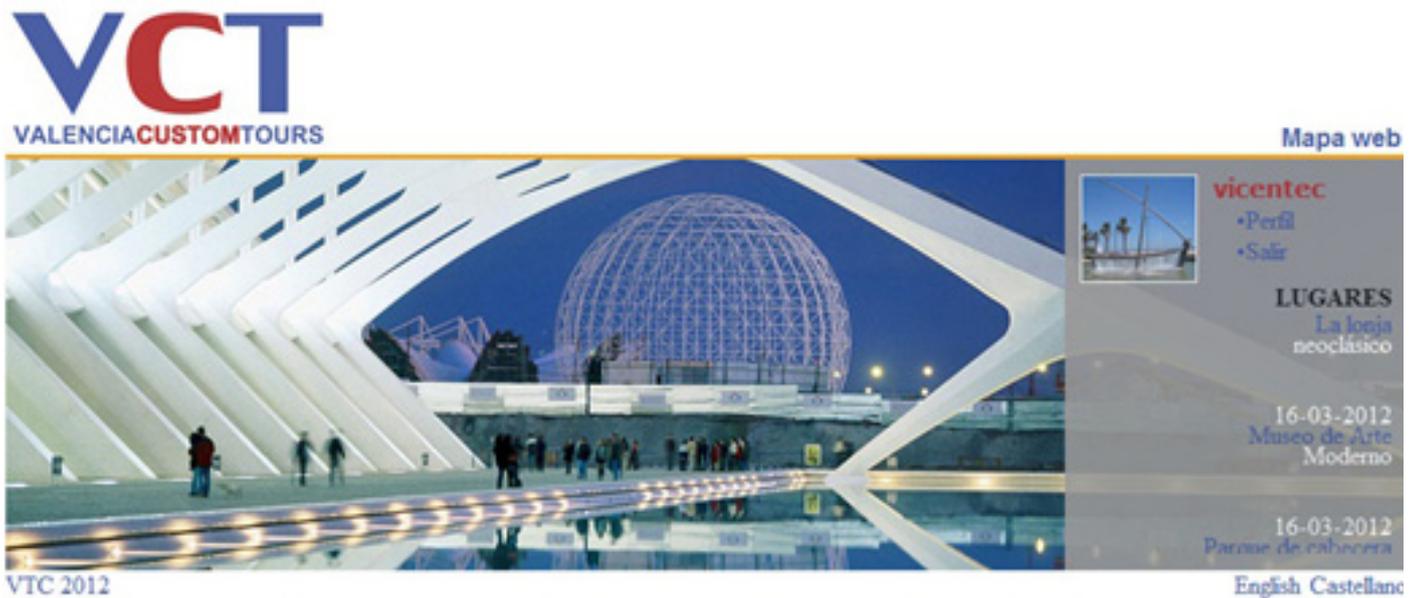


Fig 4. Página de inicio con sesión iniciada

Una vez que el usuario ha ingresado en la web, bien por introducir los datos de entrada o bien por rellenar los campos del registro, el usuario vuelve a la pantalla de Home (figura 4). En la zona de identificación, encontramos el avatar que se asocia al usuario, el nombre de usuario que le acompañará durante toda la sesión, y dos links:

- **Salir:** Este link elimina la información de la sesión. Se usa para salir de una cuenta y tener la posibilidad de registrarse o identificarse con un usuario diferente.

- **Perfil:** Este es el link que se utiliza para acceder al perfil del usuario, y así poder continuar con el proceso de recomendación.

4.3 Perfil del usuario



Fig 5. Página de valoración de perfil de usuario

Una vez que se ha pulsado el link de “Perfil” en la pantalla de inicio, accedemos a la pantalla que se muestra en la figura 5. Aquí es donde el usuario introduce la información de sus preferencias personales que se almacenarán en su perfil. Los datos rellenados aquí servirán para que después el sistema de recomendación elija los destinos que más se adecuen al perfil del solicitante.

Esta información se almacena en la base de datos con valores del 1 al 5, para cada una de las opciones. El usuario puntuará cada preferencia mediante un sistema de “Estrellitas”, que está implementado mediante el lenguaje JavaScript, y se usa para que el usuario pueda ver en tiempo real el estado de las estrellitas. Se profundizará más sobre el aspecto técnico de la implementación en el siguiente capítulo de la memoria.

Si el usuario teclea directamente la última estrellita, para dar una puntuación de 5 a alguna de las preferencias, automáticamente las cuatro anteriores se sombrearán también. Una vez el usuario ha acabado de rellenar este formulario gráfico, debe pulsar el botón “Enviar”, para registrar la información en la BBDD y continuar con el siguiente paso de la recomendación.

El usuario solamente tendrá que rellenar esta información la primera vez que acceda a su perfil, puesto que se queda almacenado en la base de datos. Las siguientes veces que acceda al perfil, las estrellas conservarán los mismos valores que se les dio por última vez. No obstante, el usuario podrá modificar dichos valores introduciendo la modificación y volviendo a darle al botón de “Enviar”.

En la página de perfil también podemos observar que la línea amarilla que había debajo del logotipo, se ha convertido en un menú, que nos acompañará a lo largo del resto del proceso de recomendación.

VCT
VALENCIA CUSTOM TOURS

Mapa web

Datos personales Perfil de usuario Visita actual Recomendación Ruta

vicentec
• Perfil
• Salir

Quiere modificar algo de su información?
Aqui puede modificar o actualizar cualquiera de los datos de su información personal que le pedimos para registrarse, si dejó algún campo pendiente de rellenar ahora puede hacerlo.

<input type="text" value="vicentec"/>	<input type="text" value="Vicente Civera Arasteny"/>
<input type="text" value="*****"/>	<input type="text" value="42"/>
<input type="text" value="*****"/>	<input type="text" value="Email"/>
<input type="text" value="-- ¿Tienes hijos? --"/>	<input type="text" value="-- Seleccione Sexo --"/>
<input type="text" value="3"/>	<input type="text" value="-- Selec. Ocupacion --"/>
<input type="text" value="8"/>	<input type="text" value="-- ¿Tienes pareja? --"/>

Modificar

Fig 6. Página de modificación de datos personales

Si pulsamos sobre la primera opción del menú “Datos personales” accederemos a una página muy parecida a la que se utilizó para efectuar el registro. En ella podemos encontrarnos los datos que rellenamos con anterioridad.

En esta pantalla podemos modificar todos los datos personales excepto el de usuario, que aparece en sombreado y con el Textbox deshabilitado para edición.

Si pulsamos una vez sobre el recuadro que queramos modificar, la información contenida por éste se borrará automáticamente para facilitarnos que insertemos el dato nuevo. Si lo queremos en blanco, al desmarcar la casilla automáticamente se pondrá en el recuadro la información que debemos ingresar en dicho campo.

Cuando se haya acabado de modificar toda la información y los datos introducidos pasen los controles necesarios para su validación, pulsaremos el botón de “Modificar” para guardar todos los cambios en la base de datos y automáticamente, si todo ha ido bien, la página se redirigirá a la siguiente opción del menú, que es la pantalla de perfil. Este punto es muy importante ya que si en lugar de pulsar el botón de “Modificar”, pulsamos el link del menú “Perfil de usuario”, la web accederá a la misma pantalla pero no se guardarán los datos modificados.

De vuelta en la pantalla de perfil de usuario, hay dos opciones para acceder a “Visita Actual” que es el siguiente paso del sistema de recomendación:

- Clicando en el menú directamente la opción “Visita actual” si no se desea introducir ningún cambio en las preferencias generales,
- Cambiando la preferencia que se quiera modificar y darle a “Enviar”.

4.4 Visita actual, recomendación y ruta

VCT
VALENCIA CUSTOM TOURS

[Mapa web](#)

Datos personales Perfil de usuario **Visita actual** Recomendación Ruta

vicentec
• Perfil
• Salir

Que vas a visitar hoy?

Por favor, introduce toda la información que le sea posible, para obtener una mejor recomendación de ruta, y no olvide que los campos marcados con (*) son obligatorios.

Preferencias generales

- Compras
- Museos
- Espacios abiertos
- Arquitectura religiosa
- Arquitectura defensiva
- Arquitectura civil
- Gastronomía
- Deportes
- Monumentos
- Ocio
- Salud y SPA
- Eventos
- Niños
- Ninguna

Horario de visita

Fecha(dd/mm/aaaa)

hora de inicio (HH:MM)

hora de fin (HH:MM)

Recomendación

VTC 2012 [English](#) [Castellano](#)

Fig 7 Pagina de visita actual

En ambos casos accederemos a la pantalla de “Visita actual”, que es la que se nos muestra en la figura 7. En ella podemos observar las mismas preferencias generales que vimos en el perfil, aunque esta vez no tienen sistema de puntuación, sino un Checkbox al lado de cada una.

Las opciones que elijamos en esta pantalla son las que se le pasarán al recomendador para que éste genere la visita actual. Además de las preferencias específicas se pide:

- **El día de visita:** fecha en la que se va a realizar la visita por la ciudad, que deberá introducir en el mismo formato que se muestra dd/mm/aaaa.

- **La hora de inicio:** Este campo se cumplimentará con la hora de inicio de la visita, en formato HH:MM

- **La hora de fin:** Éste campo se cumplimentará con la hora de finalización de la visita, en formato HH:MM

- **¿Vienes con hijos?:** Esta opción sólo le aparecerá a los usuarios que en la información personal pusieran que sí que tienen niños.

Una vez cumplimentado el formulario anterior, presionamos el botón de “Recomendación”, en este momento la página ejecutará la aplicación con los datos facilitados por el usuario y devolverá los sitios que recomienda para la visita actual.

VCT
VALENCIA CUSTOM TOURS

[Mapa web](#)

Datos personales Perfil de usuario **Visita actual** Recomendación Ruta

vicentec

- Perfil
- Salir



Que vas a visitar hoy?

Por favor, introduce toda la información que le sea posible, para obtener una mejor recomendación de ruta, y no olvide que los campos marcados con (*) son obligatorios.

Preferencias generales

- Compras
- Museos
- Espacios abiertos
- Arquitectura religiosa
- Arquitectura defensiva
- Arquitectura civil
- Gastronomía
- Deportes
- Monumentos
- Ocio
- Salud y SPA
- Eventos
- Niños
- Ninguna

Horario de visita

Fecha(dd/mm/aaaa)

hora de inicio (HH:MM)

hora de fin (HH:MM)

Recomendación

VTC 2012 [English](#) [Castellano](#)

Fig 7B. Página de visita actual con checkbox marcados

Cabe destacar que aunque aparezcan en el menú las opciones posteriores a la recomendación, como “Recomendación” o “Ruta”, éstas no serán accesibles desde la pantalla de “Visita actual”, ya que obligatoriamente se tiene que ejecutar el recomendador con los datos necesarios para acceder a ellas. Desde las opciones más avanzadas siempre se puede acceder a las pantallas anteriores, de forma que por ejemplo desde la pantalla de “Visita actual” no podríamos acceder a “Recomendación” o “Ruta”, por ser posteriores, pero desde “Ruta” podríamos acceder a cualquiera de las opciones anteriores.



Fig 8. Página de selección de lugares a visitar

Una vez ejecutado el recomendador, accedemos a la página de Recomendación (Figura 8). En esta página podremos encontrar el resultado del recomendador, que se compondrá de una lista de lugares. Estos están estructurados en columna en la página de Recomendación, y a la derecha tienen cada uno un combobox con las opciones:

- **Seleccionado:** con esta opción, elegimos el sitio para ser incluido en la ruta final

- **Rechazado:** eliminamos de la ruta final este lugar.

- **Indiferente:** el recomendador, dependiendo de varios factores, como cercanía, cantidad de sitios seleccionados... evalúa si se incluye o si se rechaza.

A la derecha del combobox, encontramos un link de mapa, que abre un pop up donde nos aparece el sitio de interés en el mapa, con su información (figura 9).

Para representar el punto en el mapa, se utiliza Java Script y la librería de Google Maps. En el mapa se introducen varios controles, como se puede observar en la figura 9, estos son:

- **De movimiento** por las zonas contiguas al punto de interés. Son las flechitas situadas en la parte superior izquierda del mapa
- **De zoom**, para poder ver el mapa con mayor o menor detalle. Se encuentra en la parte izquierda del mapa y su valor por defecto es 13.
- **De tipo de mapa.** Hay tres tipos: mapa, satélite e híbrido
- **Recuadro de zoom.** Se traza de un recuadro con un zoom mucho más alejado para que se vea la zona por la que se mueve

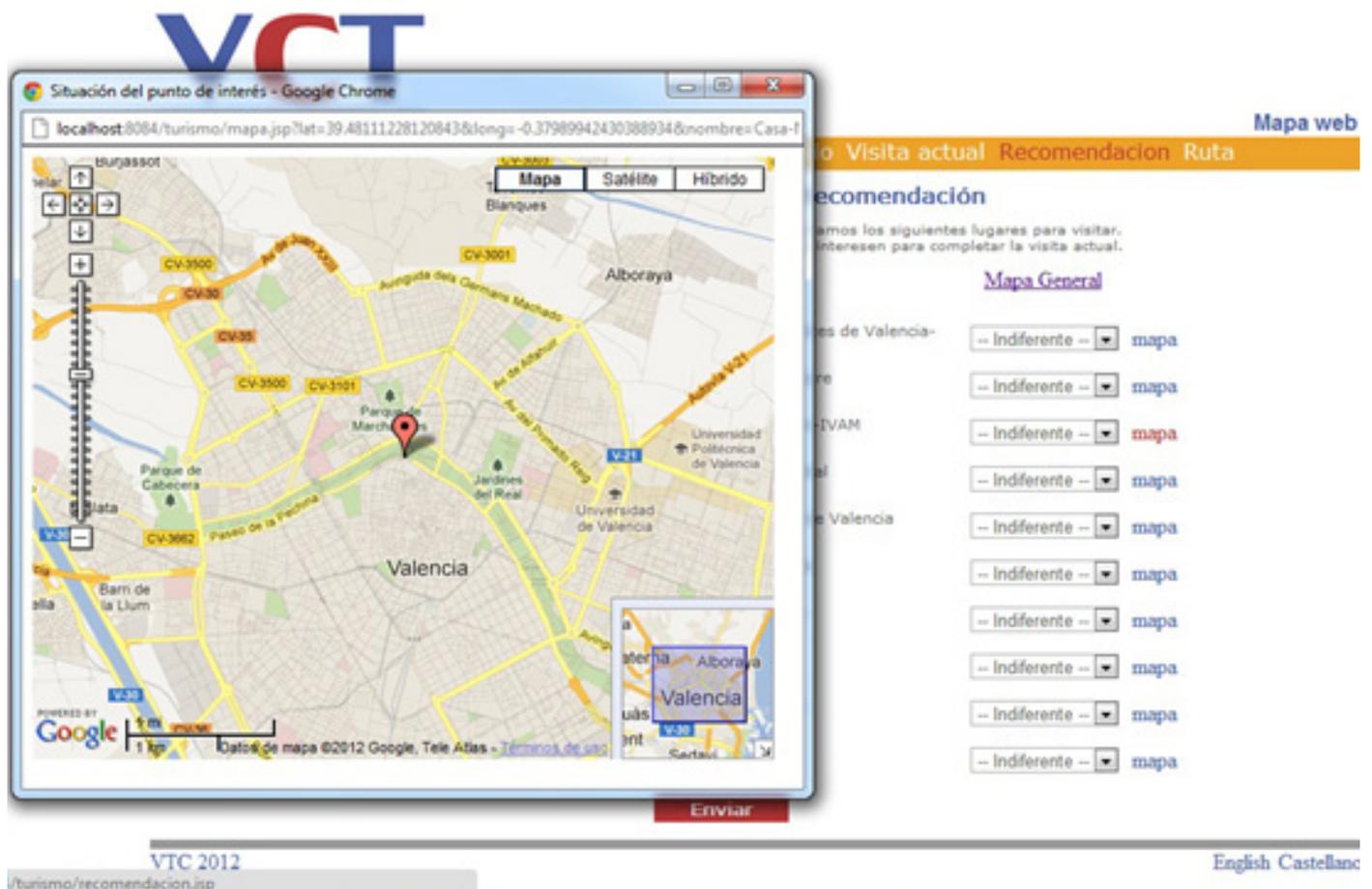


Fig 9. Pop up con un punto de interés en el mapa

En la pantalla de recomendación, también podemos observar la opción “Mapa General”. Se trata de un link que nos abre un pop up donde aparece un mapa de las mismas características que el explicado anteriormente, pero donde se dibujan todas las opciones de la ruta. Lo podemos ver en la figura 10.

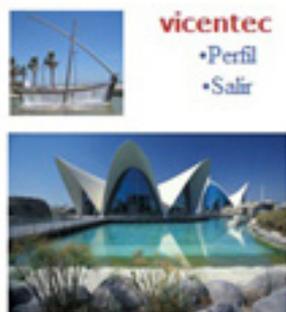
Diseño de la interfaz

Este mapa le servirá al usuario para determinar dónde se encuentra cada punto respecto a los otros. Y así ayudarle a decidir los lugares de interés que quiera visitar.

The image shows a web browser window displaying a Google Map of Valencia. The map is titled 'Situación del punto de interés - Google Chrome' and shows several red location pins. To the right of the map is a sidebar with a navigation menu: 'Inicio', 'Usuario', 'Visita actual', 'Recomendación', and 'Ruta'. The 'Recomendación' section is active, showing a list of recommended locations. Each location has a dropdown menu set to 'Indiferente' and a 'mapa' button. The locations listed are: 'Mapa General', 'Artes de Valencia', 'Benlure', 'Carmen-IVAM', 'Catedral', 'Historia de Valencia', 'Lestalla', 'Norte', 'Central', 'Colón', and 'afico'. At the bottom of the page, there is a footer with 'VTC 2012' and 'English Castellano'.

Fig 10. Popup con varios puntos de interés en el mapa

Una vez que el usuario ha seleccionado los lugares que quiere visitar, debe presionar el botón "Enviar" y así accederá a la última parte de la página.



Recomendación

vicentec, te recomendamos los siguientes lugares para visitar. Selecciona los que te interesen para completar la visita actual.



	Plaça del Reial
	2,4 km (aprox. 7 minutos)
1.	Dirigete hacia el sur en Plaça del Reial hacia Pont del Real 7 m
2.	Gira a la izquierda para continuar por Plaça del Reial 83 m
3.	Gira a la izquierda para continuar por Plaça del Reial 20 m
4.	Gira a la izquierda para continuar por Plaça del Reial 130 m
5.	Gira a la derecha hacia 450 m

Fig 11. Página con una ruta entre varios puntos

El último apartado de la web es la página de ruta, la cual se puede visualizar en las figuras 11 y 12. Esta página está formada por dos partes:

- **Mapa:** se trata de un mapa de Google Maps, donde se puede apreciar la ruta que hay que seguir por el callejero, entre cada uno de los destinos que se van a visitar. Este mapa cuenta con los mismos controles que los que he comentado anteriormente

- **Descripción de la ruta:** Contiene información de la ruta, tal como tiempo, calles, distancia... También se genera mediante el api de Google y se visualiza mediante el plasmado en un div de la información contenida en un XML. La arquitectura empleada se explicará en el siguiente capítulo de la memoria.

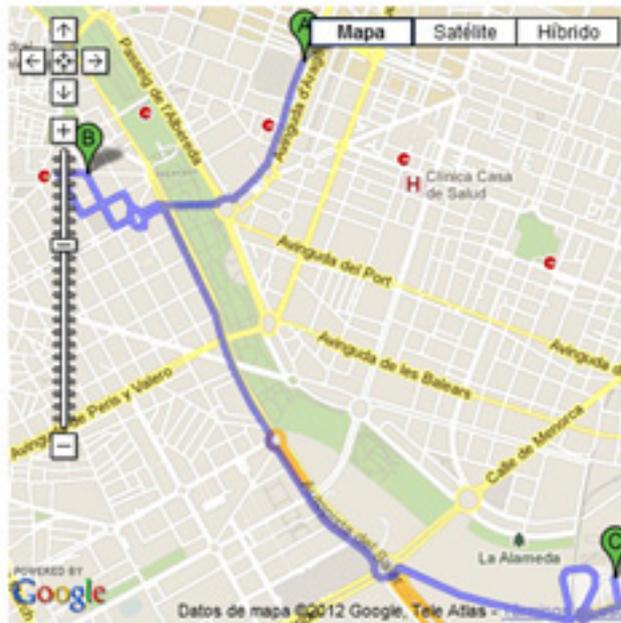

vicentec

- Perfil
- Salir



Recomendación

vicentec, te recomendamos los siguientes lugares para visitar.
 Selecciona los que te interesen para completar la visita actual.



Avinguda d'Aragó

1,7 km (aprox. 4 minutos)

1. Dirígete hacia el sur por **Avinguda d'Aragó**
Pasa una rotonda 750 m
2. En **Plaça de Saragossa**,
toma la tercera salida hacia **Pont d'Aragó** 350 m
3. Continúa por **Gran Via del Marqués del Túria**. 140 m
4. Gira a la derecha hacia **Carrer de Salamanca** 93 m
5. Gira a la derecha hacia **Gran Via del Marqués del Túria** 88 m
6. Toma la primera a la derecha 200 m

Fig 12. Página con una ruta entre tres puntos

En este punto se van a comentar los apartados técnicos de la página, se hablará de la implementación, los lenguajes de programación utilizados y cómo de la combinación entre ellos obtenemos la aplicación que se presenta

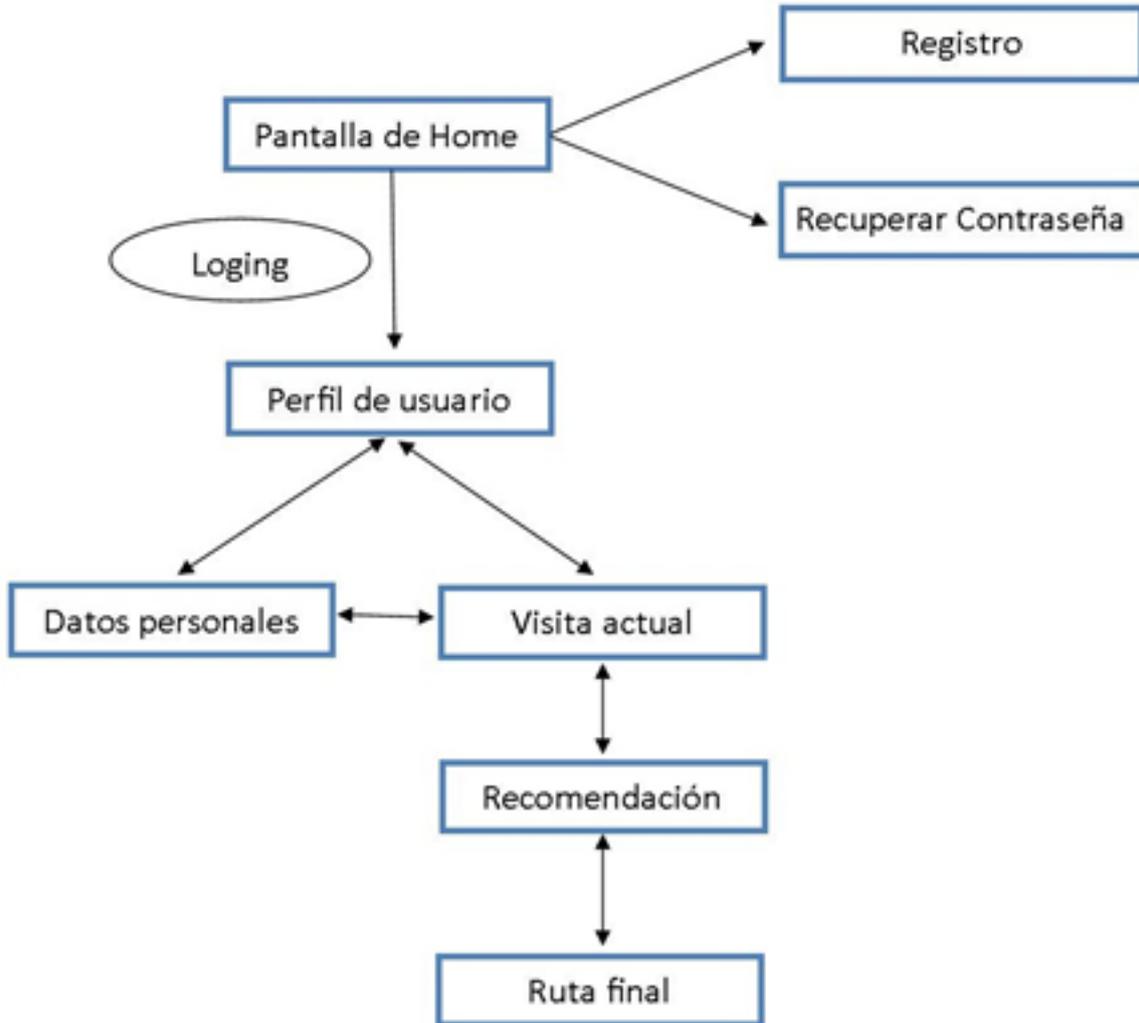


Fig. 13a Diagrama de flujo de la web

En la figura 13a se puede observar el diagrama de flujo de la página y cómo, por ejemplo, para poder acceder a las opciones de recomendación el sistema obliga a estar registrado e identificado, o cómo sólo se puede acceder a la página de recomendación a través de la página de visita actual. etc.

5.1 Detalles técnicos de implementación

Al tratarse de una página web, todo lo que podemos visualizar en la página, se encuentra escrito en lenguaje HTML, y distribuido y formateado mediante hojas de estilo CSS. Se utiliza una hoja de estilo para cada pantalla de la página. Esta no es la opción a priori más potente, ya que tiene que cargar varias, pero se ha implementado así por dos motivos.

El primero porque se trata de una web con una interfaz sencilla para acelerar el proceso de carga, con lo que las hojas de estilo asociadas a cada página son de tamaño muy reducido. Y el segundo porque todas las paginas comparten la plantilla, es decir que están hechas por el mismo patrón, con lo que si en un momento dado se quiere modificar una de las pantallas, automáticamente se modificarían todas, dividiendo el CSS en hojas de estilo más pequeñas e individualizadas. Lo que se intenta es tratar de facilitar en el futuro el cambio de alguna de las páginas. Más adelante se explicará la estructura del CSS y la plantilla “hueca” del código HTML.

Pero tanto el HTML como el CSS son lenguajes estáticos, que sólo permiten mostrar elementos en un navegador. En este proyecto había que crear una interfaz para un sistema recomendador que tiene que utilizar datos de una aplicación Java e intercambiar información con una base de datos, así que para “pintar” el HTML anterior utilizamos plantillas Java, en concreto paginas JSP.

Como se comenta en el párrafo anterior se utilizan paginas JSP para la implementación de la web. Se utiliza Java porque es un lenguaje potente a la hora de interactuar con bases de datos o con aplicaciones Java y la página web necesita constantemente estar manipulando información de la base de datos y, cuando ya tiene todos los datos que necesita, ejecutar el sistema de recomendación para obtener la lista con las opciones de la visita.

Otro motivo importante del uso del lenguaje Java es porque en los estudios de Ingeniería Técnica en Informática de Gestión, el lenguaje que más se profundiza, es precisamente éste, con lo que ya tenía experiencia con el tanto en la codificación de páginas web como en la implementación de aplicaciones.

Para explicar todos los apartados técnicos de la página se va a dividir el capítulo en tres partes; en la primera explicaré la plantilla HTML con la disposición de los Divs de la web, en la segunda explicaré brevemente algunas de las funciones JavaScript y en la última y más importante explicaré las paginas JSP más importantes y los detalles que son comunes a todas ellas. Dentro de este punto en concreto en la parte de JavaScript no se va a explicar el uso del API de Google Maps, porque se explicará en otro capítulo de este documento.

5.1.1 Plantilla HTML/CSS

Para la creación gráfica de la página web se crean dos plantillas, una con la estructura de las capas (Divs) de la página de inicio y la otra con la estructura del resto de páginas, las cuales coinciden.



FIG 13b. Esquema de las capas de Inicio.jsp

```

<HTML>
  <head>
    ...
  </head>
  <body>
    <div id="fondo">
      <div id="cabecera">
        ...
      </div>
      <div id="linia"></div>
      <div id="imagenes">
        ...
      <div id="capa">
        <div id="zonaLogging">
          ...
        </div> <!-- fin de zonaLogging -->
        <div id="lugares">
          ...
        </div> <!-- FIN DE LUGARES -->
      </div> <!-- FIN DE CAPA -->
    </div> <!-- FIN DE IMAGENES -->
  </div><!-- FIN DE FONDO -->
</body>
</HTML>
    
```

En la figura 13b podemos ver cómo están posicionadas las capas de la página de inicio de la web. Y a continuación tenemos el “esqueleto” HTML. El término “esqueleto” se utiliza porque se ha eliminado del código toda información que no tenga que ver con la estructura de la página. Este código HTML junto a la hoja de estilo (CSS) forman visualmente la página de inicio.

Todas las capas utilizadas tienen la propiedad de posición “relativa”, de forma que la introducción de nuevos elementos en la página en zonas determinadas no causaría una desfiguración de ésta.

En la figura 13b se puede observar que a cada recuadro o capa le corresponde un número. Estos números se van a utilizar para la correcta comprensión de la imagen junto al código escrito a continuación.

La primera capa de la página, que es la que contiene la información sobre el fondo es el recuadro de color azul (1) y en el código corresponde al div con el id “Fondo”. Ésta a su vez, contiene dos capas que son el rectángulo rojo (2), que es el div de id “Cabecera” y que contiene el logo de la página, y el rectángulo verde (3), que es el div de id “Imágenes” que contiene las imágenes y la zona de autenticación. Estas dos últimas están separadas por un div (“Línea”) que contiene la franja amarilla que podemos ver debajo del logotipo.

Hasta aquí la página de inicio es igual al resto de páginas, con la única excepción que el div “Línea” en el resto de páginas se hace más ancho y pasa a contener el menú.

Dentro del div “Imágenes” (3) en la parte derecha, tenemos el div “Capa” que se corresponde con el rectángulo naranja (4). Los datos más relevantes del diseño de este div es que a la propiedad z-index se le pone un valor más elevado para que se muestre encima de las imágenes y a la propiedad opacity se le da un valor cercano al 80% para que se transparente, con esto conseguimos que se aprecie que la imagen va cambiando debajo de la zona de autenticación y de últimas visitas.

Para finalizar la plantilla de la página de inicio, mencionar que el div “Capa” (4) contiene a su vez los divs “ZonaLogin” y “Lugares” que son el rectángulo rosa (5) y celeste (6) respectivamente.

Y para finalizar el apartado de implementación HTML y CSS, se expondrá la estructura de la plantilla de las páginas de contenido de la web.



FIG 14. Esquema de capas general de la web

```

<HTML>
  <head>...</head>
  <body>
    <div id="fondo">
      <div id="cabecera">...</div>
      <div id="linia">...</div>
      <div id="imagenes">
        <div id="descripcion">
          <div id="logeado">...</div>
          <div id="logeado2">...</div>
          <div id="img1">...</div>
        </div>
        <div id="formulario">
          <div id="encima">...</div>
          <div id="encima2">...</div>
          <div id="izquierda">...</div>
          <div id="derecha">...</div>
        </div>
      </div> <!-- FIN DE IMAGENES -->
      <div id="liniaCierre"></div>
      <div id="vtc2012">VTC 2012</div>
      <div id="castellano">Castellano</div>
      <div id="ingles">English</div>
    </div><!-- FIN DE FONDO -->
  </body>
</HTML>

```

Las capas “Fondo”, “Cabecera” e “Imágenes” siguen correspondiendo a los rectángulos 1, 2 y 3 respectivamente, pero entre los dos últimos podemos ver que el div “Línea”, que es el rectángulo número 14, se ha hecho más grande. Esto es porque en las páginas de datos esta capa contiene el menú.

Ahora la capa de “Imágenes” se divide en 2 capas, que son “Descripción”, que es el número 4 y “Formulario” que es el número 5. A su vez el div descripción se divide en tres:

- “Logeado”: se trata del rectángulo número 6 y contiene el logotipo del usuario
- “Logeado2”: que es el número 7 y contiene la opción de perfil y salir
- “Img1”: contiene una imagen decorativa

El div “Formulario” se divide en cuatro capas que se van a explicar a continuación:

- “Encima”: se trata del número 9 y es donde aparece el título
- “Encima2”: se corresponde con el número 10 y contiene la descripción
- “Izquierda”: contiene los datos situados en la izquierda, es el numero 12
- “Derecha” : contiene los datos situados en la derecha, es el numero 13

5.1.2 Estructura y codificación de las paginas JSP

En este apartado vamos a explicar la estructura y codificación de las paginas JSP más importantes. Explicaremos las funciones de validación comunes a todas las páginas y la división de algunos fragmentos de código en diversos archivos JSP diferentes para luego cargarlos en otros con includes.

5.1.2.1 Includes.jsp

Este archivo sólo contiene código Java. Junto con el archivo ControlSesion.jsp, que se explicará más adelante, contiene código necesario para todas las páginas de la web.

Esta página contiene los “Imports” necesarios para que compilen el resto de páginas Java :

```
<%@page import="Java .io.IOException" %>
<%@page import="Java .sql.Connection" %>
...
<%@page import="Java x.servlet.http.HttpServletRequest" %>
<%@page import="Java x.servlet.http.HttpServletResponse" %>
```

Pone a ‘true’ la propiedad de Java “Session” para que se pueda llevar un control de la sesión y guardar variables:

```
<%@page session="true"%>
```

Inicializa la variable “SesionActiva” a false para que luego se valide en el ControlSesion.jsp

```
boolean sesionActiva=false;
```

Y por último inicializa la variable sesión para que podamos usarla para incluir y modificar variables de sesión en todas las páginas:

```
HttpSession sesion=request.getSession();
```

5.1.2.2 ControlSesion.jsp

Esta página también contiene únicamente código Java, y es la que se encarga de validar si el usuario está identificado o no. En caso de no estarlo, este script espera las variables “user” y “pass”, las contrasta con la información de la base de datos y le da el valor apropiado a la variable “SesionActiva” (‘True’ si ha tenido éxito o ‘False’ en el caso contrario). A continuación se presenta un esquema en pseudocódigo:

```
sesionActiva=false
Si ( sesión(usuario) no existe)
  Si ( User y pass existen ) {
    Conectamos a la base de datos
    Si ( existen en la base de datos )
      sesionActiva=true
    }
  }
//existe usuario, por lo que hay sesión activa
Si no sesionActiva=true
```

5.1.2.3 Inicio.jsp

Esta página se compone fundamentalmente de código HTML, que básicamente es el que se explica en el punto anterior, con el añadido de algunas líneas de código necesarias para hacer que la imagen principal haga un efecto de desvanecimiento en la transición.

```
<div id="slideshow">
  
  
  
  
</div>
```

Esta página incluye "Includes.jsp" (para utilizar las librerías de Java y tener la sesión activa), "ControlSesion.jsp" (para dar valor a la variable "SesionActiva") y "Scripts.jsp".

Scripts.jsp es una página que únicamente contiene el código Java Script (jQuery) necesario para hacer el efecto de transición de las imágenes de la página de inicio. Sólo se utiliza en "Inicio.jsp" y se implementó en un JSP aparte para simplificar la página de inicio.

El único código Java que contiene esta página es el que se encarga de evaluar la variable SesionActiva, ya que si esta está a "True" la página "pinta" en la zona de identificación el logotipo del usuario, su nombre de usuario, la opción de ir a perfil y la opción de cerrar sesión. Y si por el contrario se encuentra a "False" se "pinta" en la zona de identificación de usuario un formulario para ingresar el usuario y la contraseña.

5.1.2.4 Registro.jsp

Esta página incluye un formulario en HTML para captar los datos del usuario. Ese formulario envía los datos a “CompruebaRegistro.jsp” que es la que se encarga de hacer las comprobaciones oportunas y dar de alta en la base de datos al usuario en caso de que todo esté correcto.

La página contiene los includes “Includes.jsp” para ejecutar cualquier código Java y mantener la variable de sesión activa y “ControlSesion.jsp” para darle valor a la variable booleana “SesionActiva”. Acto seguido se hace una comprobación:

```
<% if(sesionActiva){ response.sendRedirect("inicio.jsp");}%>
```

En el caso de que ya exista una sesión activa, la página se redirecciona a Inicio. Esto se implementa para que un usuario ya autenticado no se pueda registrar, por ejemplo, poniendo la url completa en el navegador.

La última parte de código Java que contiene la aplicación, carga (si existe) el parámetro “error” y hace una comprobación.

- Si la variable “error” tiene contenido, el código html que se escribe es:

```
<Body onLoad=\"alert(‘‘+error+’’)\">
```

- Si por el contrario la variable error es “Null” o tiene un String vacío escribe la etiqueta body normal:

```
<Body>
```

El uso de la comprobación es para mostrar el error por el cual el usuario no se ha podido registrar en la página, ya que si “CompruebaRegistro.jsp” encuentra un error en la validación de la información pasada desde “registro.jsp”, ésta se redirige a “Registro.jsp” con la variable “error” cargada con el error detectado.

Resumiendo el párrafo anterior, la página “Registro.jsp” no mostrará error siempre que provenga de Inicio.jsp y sí que mostrara error cuando sea desde “CompruebaRegistro.jsp”.

Esta página contiene una función JavaScript, que es “deseleccionar(valor)” que se encarga de comprobar si el campo del formulario está vacío, y en el caso de estarlo, pone por defecto su descripción.

5.1.2.5 CompruebaRegistro.jsp

Esta página no contiene código HTML, se trata de un script en Java que capta la información enviada mediante "Get" desde la página "Registro.jsp".

Para empezar se inicializan todas las variables que se esperan, y luego se realizan una serie de comprobaciones para validar la información. Se comprueba:

- Que el campo usuario no esté vacío
- Que los campos de contraseña tampoco estén vacíos
- Que los campos de contraseña coincidan

Si alguna de las comprobaciones anteriores falla, la página se redirige a "Registro.jsp" con la información del error en la variable "error".

Después de esas comprobaciones se estima que la información es correcta y está lista para ser insertada en la base de datos, pero como la mayoría de los campos son voluntarios, no se puede insertar toda la información a la vez, así que se opta por hacer un "Insert" de los datos que son obligatorios, y después se van comprobando los campos uno por uno. Los que no tienen información se ignoran y con los que sí que la tienen, se hace un "Update" para insertar la información en la tabla "Users".

Después se crea una variable de sesión con el usuario, para que la próxima vez que se ejecute el "ControlSesion.jsp" este usuario aparezca ya como usuario registrado y autenticado. Por último la página se redirige al siguiente paso para el cálculo de la ruta que es "Perfil.jsp"

5.1.2.6 Iniconexion.jsp y Ficonexion.jsp

"Iniconexion.jsp" y "Ficonexion" contienen una secuencia de líneas de código comunes al momento de establecer una conexión con el servidor de base de datos (MySQL). Esto tiene una doble función:

1. En el caso de cambiar la base de datos, no es necesario modificar página por página la información.
2. Se simplifica el proceso de conexión y el código es más legible, ordenado y reutilizable.

En Iniconexion.jsp instanciamos la clase del driver Java para abrir la conexión con la BBDD y creamos la variable Connection con los datos de la base de datos que estemos usando.

```
Try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection db=DriverManager.getConnection("jdbc:mysql://localhost:6969/eturismo_
pruebas","root","11851185");
    ...
}
```

En `Ficonexion.jsp` cerramos la conexión a la base de datos, esto es muy importante hacerlo siempre ya que si no las conexiones se quedarían abiertas y el servidor acabaría dando un error por múltiples conexiones.

```
db.close();  
}catch(Exception e){out.print(e);}
```

Entre ambas funciones se ejecuta el código necesario para, hacer una consulta a la base de datos, un "Insert" o "Delete".

5.1.2.7 Password.jsp, Rec_pass.jsp y MensajeEnviado.jsp

En "Password.jsp" únicamente podemos ver código HTML. Solamente se ejecuta (como en casi todas) el "ControlSesion.jsp", para controlar que un usuario que ya esté autenticado no pueda pedir la contraseña.

Después de escribir en el único campo disponible el nombre que ha extraviado la contraseña y darle a enviar, ésta se redirige a "Rec_pass.jsp".

Éste sólo contiene el código Java necesario para enviar por e-mail la contraseña perdida. Primero los includes (Includes.jsp) y luego los "Imports" de "EnviarMail.Java" y "Java x.activation.*". Estos dos últimos necesarios para utilizar la aplicación Java de envío de mail.

Acto seguido se recupera la variable "User" proveniente de "Password.jsp" y se hace una consulta a la base de datos para comprobar que exista. Después de la conexión a la base de datos se pueden dar tres posibilidades:

1. Que el usuario exista y tenga su e-mail registrado, en cuyo caso se crea un objeto de la clase "EnviarMail" al que se le pasa el E-mail obtenido de la base de datos y la contraseña, para que la aplicación envíe el correo con el Password al usuario. Después se redirige la página a MensajeEnviado.jsp con la variable "fin=0".

2. Que el usuario exista, pero que no ingresara el mail en el registro, se redirige a la página MensajeEnviado.jsp con la variable "fin=1".

3. Y que no exista el usuario, se redirige directamente a la página MensajeEnviado.jsp con la variable "fin=2".

La página "MensajeEnviado.jsp" es exactamente igual que "Password.jsp", con la diferencia de que donde en "Password.jsp" tenemos el formulario para enviar el usuario, en "MensajeEnviado.jsp" tenemos un condicional con el mensaje a mostrar dependiendo del resultado de la variable fin; 0 se ha enviado bien, 1 no se ha enviado porque el usuario no insertó su e-mail y 2 si el usuario no existe.

Con esta forma de codificación el usuario cuando aprieta el botón "Enviar" visualiza instantáneamente el mensaje resultado, sin que nada más aparentemente cambie en el resto.

5.1.2.8 Perfil.jsp

Esta página combina el código HTML necesario para visualizar la página correctamente en el navegador, el código Java para realizar las comprobaciones necesarias y el código Java Script para manipular la estrellitas de puntuación.

Primero se valida que el usuario está identificado, sino se reenvía a la página de registro, después se inicializan todas las variables que se van a evaluar a 0.

Después de hacer una consulta a la base de datos para recuperar los datos del usuario, se “dibujan” las estrellitas conforme a los valores extraídos de la base de datos. De esa forma, cuando el usuario se conecta para modificar su perfil, puede ver las estrellitas tal y como las guardó la última vez. La función JavaScript encargada de esta tarea es “Inicializar()”

En Perfil.jsp se implementa mediante HTML un formulario, no visible al usuario, con los datos del perfil. Cada vez que marcamos una estrella para establecer un valor de un campo, la función JavaScript “selección(objeto)” modifica el valor del formulario “invisible” al usuario con el valor que éste ha marcado en las estrellitas. De esta forma cuando el usuario presiona el botón “Enviar”, la pagina hace un “submit” del formulario con los valores marcados con las estrellitas.

5.1.2.9 CompruebaPerfil.jsp

Esta página contiene únicamente código Java y se encarga de obtener los valores marcados por el usuario en la página de “Perfil.jsp” e insertarlos en la base de datos. Para ello realiza una consulta a la tabla “Preferences” para a partir del “Nombre” (que es lo que obtiene de “Perfil.jsp”), obtener el campo “Id” que será necesario para luego insertar la información en la tabla de “Users_preferences”.

Antes de actualizar la información en la base de datos, se hace una consulta para saber si los campos que se pretenden registrar se encuentran en la base de datos o no. Si se encuentran ya en la base de datos se actualizan mediante “Update”; si por el contrario no están en la base de datos, se realiza un “Insert”. Y para finalizar, la página se redirige al siguiente paso de la recomendación, que es “Preferencias.jsp”.

5.1.2.10 Preferencias.jsp

Se compone del código HTML necesario para crear un formulario con catorce checkbox con las preferencias específicas, donde el usuario marcará las que más le interesen, y tres Textbox para que el usuario inserte la fecha, hora y duración. Este formulario pasará los datos a “CargaAplicacion.jsp”, donde se llamará a la aplicación.

Por lo que hace referencia al código Java implementado en esta página, en primer lugar se incluye la línea de código que, como en la mayoría de páginas de esta web, comprueba si el usuario está o no autenticado, en el caso de no estarlo lo envía a la página de registro.

Como la página de registro, ésta también espera el parámetro “error”. Este parámetro le viene de “CargaAplicacion.jsp”, en el caso de no ser “Null” se muestra el texto mediante un Alert. Esto se implementa para poder tratar el caso en el que el usuario introduzca mal la fecha, hora o duración y así no dar pie a que se pueda producir un error en tiempo de ejecución al lanzar la aplicación.

Por último, se hace una llamada a la base de datos, para saber si el usuario tiene hijos; en el caso de tenerlos se muestra un combobox adicional para que el usuario especifique si va a traerlos al viaje o no.

5.1.2.11 CargaAplicacion.jsp

Esta página se compone únicamente de código Java, y es una de las más importantes de la web. Lo primero que se codifica son los includes y el control del usuario registrado.

Después se realizan las comprobaciones necesarias para asegurarse de que los datos de los Textbox provenientes de “Preferencias.jsp” son correctos; en el caso de no serlo se redirige la página a Preferencias.jsp y se le da a la variable “Error” el valor del error; si son correctos se trata la información.

Cuando se habla de tratar información, se refiere a formatear correctamente los datos para pasárselos a la aplicación y que esta no provoque una excepción. Para ello se crea un array auxiliar donde se almacena el valor de los catorce campos de los checkbox, después se recorre el array auxiliar almacenando en otro array los valores no nulos. De esta forma, el resultado que nos queda es un array de “Size n” con los “n” valores no nulos. Es decir, un array con los valores que el usuario marcó.

Después se realiza una de las funciones más importantes de la web, que se trata de llamar a la aplicación, pasándole las variables tratadas anteriormente, y ejecutando el código necesario para que ésta devuelva un array con el identificador de los sitios que recomienda visitar al usuario.

Finalmente se almacenan los resultados de la aplicación en variables de sesión, para que estén disponibles sin necesidad de que se vuelva a lanzar la aplicación durante la sesión.

Por último, se redirige la página a “Recomendacion.jsp”, que es el siguiente paso.

5.1.2.12 Recomendación.jsp

En esta página, se combina código JavaScript para abrir popups con los mapas de Google, Java para inicializar todas las variables con la información sobre los diferentes sitios y HTML para la visualización de la misma en el navegador.

Empezamos con los includes de los Imports y con el control de si el usuario está autenticado o no. Después de implementan dos funciones JavaScript :

- **Mapa():** Abre un popup de 580x480 con la página “Mapa.jsp” pasándole a ésta mediante el método get, los argumentos necesarios para la carga del mapa. Argumento[0]=latitud, Argumento[1]=longitud, Argumento[2]=nombre, Argumento[3]=descripción. Esta función se utiliza para representar solamente un punto en el mapa.

- **MapaGeneral():** Abre otro popup del mismo tamaño que el anterior, pero éste no requiere ningún parámetro, ya que al mostrar todos los puntos en el mismo mapa, utiliza las variables de sesión para obtener la información que necesita.

Con lo que respecta a la parte HTML, dentro de la “plantilla” explicada inicialmente, se implementa en el cuerpo dos columnas:

- La columna de la izquierda contendrá los nombres de los sitios

- La columna de la derecha contendrá un combobox para –elegir, rechazar, indiferente- y un link a “mapa.jsp” donde se le mostrara un popup con el punto seleccionado en el mapa, todo ello incluido en un formulario que enviará los datos seleccionados a “Recomendación_final.jsp”.

Para finalizar se explica la codificación Java utilizada para dotar de funcionalidad y dinamismo a la web. Para empezar se recuperan en variables locales el contenido de las variables de sesión que se cargaron con los identificadores de los sitios en la página “CargaAplicacion.jsp” y se inicializan cuatro variables para cada identificador: nombre del lugar, latitud, longitud y descripción. En este punto tenemos las variables con los Identificadores cargadas con los resultados de la aplicación y el resto de variables “vacías”.

Por último se implementa un Include con “CargarDatosSitios.jsp”, cuya funcionalidad final es la de cargar las variables inicializadas anteriormente con los datos almacenados en la base de datos para poder ser visualizadas en la página y usadas como argumentos en las funciones JavaScript. El detalle de su implementación se explicará más adelante en el punto 5.1.2.14 del presente documento.

5.1.2.13 Mapa.jsp y MapaGeneral.jsp

La página “Mapa.jsp” está preparada para una resolución de 550x450px y se compone del código HTML básico para mostrar el script de visualización del mapa de Google.

Contiene código Java para poder asignar las variables que se le pasan como parámetro (longitud, latitud, nombre y descripción) que se utilizan para cargar el mapa.

La página “MapaGeneral.jsp” contiene el mismo código HTML que la página anterior. Por lo que respecta a la codificación Java, “MapaGeneral.jsp” no recibe los parámetros mediante “Request” como “Mapa.jsp”, sino que inicializa todas las variables necesarias para representar los 10 puntos en mapa y las carga mediante variables de sesión.

5.1.2.14 CargarDatosSitios.jsp

Sólo contiene código Java y se trata de un script para la carga de los datos de los lugares propuestos por la aplicación.

La función de este Jsp es simplificar el código de la página “Recomendación.jsp”, por ello se codifica en un Jsp aparte y luego se carga en “Recomendación.jsp” mediante un Include. Se asume en esta página que las variables con los IDs de los sitios propuestos por la aplicación ya están cargadas con sus respectivos valores, y que el resto de variables se encuentran vacías pero ya inicializadas.

Por ello directamente se hace una consulta a la base de datos, para a partir de los IDs de los sitios obtener el resto de información necesaria. Así las variables inicializadas en “Recomendación.jsp” se cargan con sus valores.

Por último se almacenan todas las variables dentro de la sesión, para que no sea necesario hacer la carga con su correspondiente llamada a la base de datos en otros puntos de la web.

5.1.2.15 ModificacionUsuario.jsp

No se va profundizar mucho en este apartado debido a que la página es prácticamente igual que la página de registro, así que en este apartado sólo vamos a explicar las pequeñas diferencias que esta tiene con la anterior.

El código HTML es el mismo, pero a diferencia de “Registro.jsp” sí que contiene código Java, además del tratamiento de errores explicado en la página de registro y del control de sesión. En “ModificacionUsuario.jsp” se realiza una consulta a la base de datos para cargar todos los valores que en su día el usuario registró en la base de datos.

Los valores cargados se insertan directamente en el formulario, para que el usuario substituya directamente el valor que quiera cambiar. Sólo hay un dato que aparece deshabilitado de modificación que es el “Usuario”.

Este formulario envía los datos a “CompruebaModificacion.jsp” para que se modifiquen en la base de datos.

5.1.2.16 CompruebaModificacion.jsp

Esta página también comparte su mayoría de código con “CompruebaRegistro.jsp”. Primero se evalúan las variables provenientes del formulario y se verifica que son correctas. Si no lo son, se redirige a la página “ModificacionUsuario.jsp” junto con la variable “Error” inicializada con el valor del error, mostrándose en ésta el error al usuario.

En el caso de que esté todo correcto, se modifican los campos requeridos en la base de datos.

5.1.2.17 SalirSesion.jsp, Scripts.jsp

La página “SalirSesion.jsp” es la que se encarga de eliminar la información del usuario de las variables de sesión:

```
request.getSession().removeAttribute(“usuario”);
```

Y redirige la página a “Inicio.jsp”. Al eliminar la variable de sesión “Usuario”, el script de comprobación de sesión que contiene la mayoría de las páginas de la web devolverá false cuando intente acceder a cualquiera de ellas. Por lo tanto el usuario se tendrá que volver a autenticar o registrar para poder acceder a los servicios de la web.

La página “Scripts.jsp” únicamente incluye el código JavaScript necesario para hacer el efecto de la transición de las fotografías de la página de inicio. Se codificó en un JSP aparte para simplificar el código de la página de inicio.

Contiene la función SlideSwitch() que utiliza a su vez la librería jQuery para hacer un efecto de desvanecimiento en la foto actual y después hacer un efecto de apareamiento de la foto siguiente, todo ello modificando la propiedad CSS “opacity”. También contiene la función “lanzadora” que se encarga de ejecutar la anterior cada 6000ms.

5.1.2.18 Recomendación_final.jsp

Para finalizar el presente apartado de la memoria, donde se describe el código en cada uno de los ficheros que componen la web, se va a explicar la codificación del fichero Recomendación_final.jsp. Este se compone del código HTML necesario para representar la página, de acuerdo con la plantilla anteriormente presentada en este documento.

Y por último, por lo que respecta al código Java , la web comienza con el “Include” de los “Imports” de Java , seguida de la comprobación de sesión.

En el caso de que la comprobación de inicio de sesión haya sido la correcta, se cargan las variables de sesión creadas anteriormente, es decir, se cargan todos los datos de los puntos a representar en el mapa.

Después se crea un array tridimensional con el valor dado al sitio en el combo box de la página “Recomendación.jsp”, la latitud y la longitud.

Por último se crea un String con el formato adecuado y se recorre el array anterior, insertando en el String la información necesaria para que luego la función JavaScript que se encarga de cargar el mapa con la ruta “Load()” lo haga de forma correcta.

El funcionamiento de las funciones JavaScript y de todo lo necesario para el uso del api de Google Maps se explicará en el capítulo siguiente de la memoria.

5.2 Uso del API de Google Maps

Una de las partes fundamentales de este proyecto es enlazar todos los datos de la base de datos y de los clientes con alguna API que permita situarlos en un mapa de forma dinámica y proporcionar información sobre los mismos. Para ello en el presente proyecto se decide utilizar el API de Google Maps.

Existen tres páginas donde se utiliza el código de Google: “Mapa.jsp”, “MapaGeneral.jsp” y “Recomendación_final.jsp”. A continuación se explicará las características de las variables JavaScript del API de Google Maps necesarias para representar el mapa.

En todas las páginas se utilizan los mismos controles:

- GMapTypeControl: Este control incluye tres botones para cambiar entre distintos tipos de mapa (mapa político de calles y carreteras, mapa satélite y mapa híbrido -calles y carreteras sobre el fondo de satélite)

- GLargeMapControl: Muestra una interfaz para hacer zoom y moverse por el mapa, con una especie de regla donde se pueden escoger de las distintas aproximaciones de la vista. Este control aparece en la parte superior izquierda.

- GScaleControl: Este es un control que simplemente informa sobre la escala actual del mapa. Con este control no se puede interaccionar, es meramente informativo.

- GOverviewMapControl: Con este control aparece un pequeño mapita en la esquina inferior derecha donde aparece una vista general con menos zoom del mapa actual, a través del cual también podemos movernos.

El detalle de la implementación es prácticamente igual en las tres páginas donde se utiliza el presente API, la única que se diferencia sería la de “Recomendación_final.jsp” que en lugar de representar puntos en el mapa, representa una ruta entre dos o varios puntos.

La implementación comienza con la creación de una función Load() que contendrá todo el código Java Script de Google y que se ejecutara al cargar el Body de la página.

```
if (GBrowserIsCompatible()) {  
var map = new GMap2(document.getElementById("map"));  
map.setCenter(new GLatLng(<%out.print(lat);%>, <%out.print(longi);%>), 13);
```

Si el navegador es compatible procedemos con la carga de la aplicación, nos creamos la variable Map, que es la que va a contener el mapa con todos sus componentes y le pasamos como parámetro del constructor el id del div donde va a ir el mapa. Una vez tenemos la variable inicializada le asignamos como centro la latitud y longitud del punto que vamos a representar en él.

```
map.addControl(new GMapTypeControl());  
map.addControl(new GLargeMapControl());  
map.addControl(new GScaleControl());  
map.addControl(new GOverviewMapControl());
```

Después le añadimos los controles que hemos descrito anteriormente en el presente apartado.

```
function addtag(point, address) {  
    var marker = new GMarker(point);  
    GEvent.addListener(marker, "click", function() {  
        marker.openInfoWindowHTML(address); });  
    return marker;  
}
```

Esta función se encuentra dentro de la anterior y su funcionalidad es crear un oyente que detecte un clic del ratón dentro del mapa y que lo asigne como punto de elección.

```
var coord_sitio = new GLatLng(<%out.print(lat);%>,<%out.print(longi);%>);  
var des_sitio = '<b><%out.print(nombre);%></b><br/><i><%out.print(descripcion);%></i><br />';  
var marker_sitio = addtag(coord_sitio, des_sitio);  
map.addOverlay(marker_sitio);
```

Por último y más importante nos creamos un punto, con la misma latitud y longitud que hemos utilizado antes para centrar el mapa entorno a él y le añadimos descripción y título.

De esta manera cuando el usuario abre la página visualiza el mapa con sus controles y el punto de interés.

Esta codificación explicada anteriormente corresponde con lo implementado en las páginas "Mapa.jsp" y "MapaGeneral.jsp".

En la página "RecomendaciónFinal.jsp" se han utilizado los mismos controles de Google Maps que hemos explicado anteriormente. La diferencia con las otras dos páginas reside en la forma de introducir el punto de interés.

Utilizando la función "Gdirload("punto de interés");" podemos representar un punto tal como si se buscara en el buscador de Google Maps. Pero lo que tiene que hacer este apartado, no es buscar un punto sino en representar una ruta.

Para solucionar esto se utilizan las palabras reservadas de Google "From y To". De esta forma le pasamos a la función explicada anteriormente un String con el siguiente formato:

```
Ruta="From c1,c2 to c3,c4 to ... to cn,cn+1"
```

Donde "cx" es la latitud del punto y "cy" la longitud del punto.

De esta manera cuando el usuario abre la página se carga el mapa con los controles inicializados y con la ruta y su descripción.

5.3 Entornos IDE usados para la web

En este punto se va a explicar los diferentes entornos de programación utilizados para el desarrollo del presente proyecto. Aunque en este apartado se va a explicar programa por programa, la funcionalidad realizada por cada uno y el lenguaje utilizado, cabe hacer una pequeña clasificación por tareas y los programas utilizados para hacer cada una de las mismas.

La creación de la página web se puede dividir en tres grandes partes. Fase de diseño, en ella se realizan labores de creación gráfica, y se utiliza el software Adobe Dreamweaver y adobe Photoshop. Fase de implementación Java , donde se usa el Netbeans y por último fase de pruebas donde se utiliza el Apache Tompcat, servidor Sqlplus y Bluej

1. Fase de diseño

1.1 - Adobe Dreamweaver

Se trata de un entorno de programación potente para el desarrollo y codificación de código HTML, CSS y JavaScript.

Este programa ofrece muy buenas características para el diseño de la página, ya que en tiempo real y sin cargar la página en el servidor se puede ir viendo el resultado gráfico. También es muy cómodo por el sistema que tiene de cerrado de etiquetas HTML, por el método de inserción de propiedades CSS y por el recordatorio de variables en CSS.

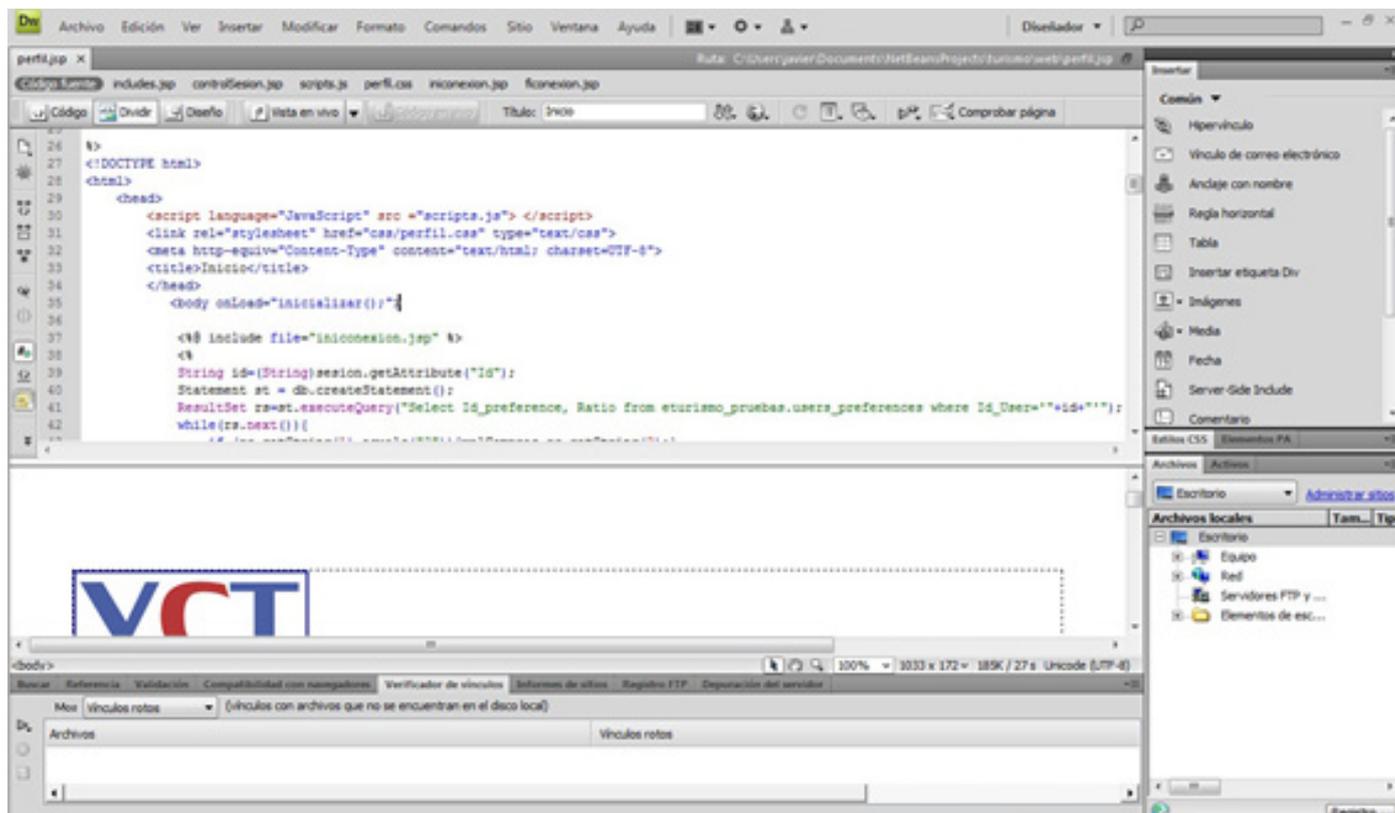


Fig 15. Dreamweaver

1.2 - Adobe Photoshop

Antes del proceso de codificación del apartado gráfico de la web, el proyecto pasa por una fase de aceptación, en la que mediante este programa de diseño fotográfico, se crean todos los logos e imágenes que va a contener la página, se editan y se guardan en formatos para que pesen lo mínimo a la hora de su carga y por último se crea una “maqueta”. Esta consiste en la realización de un trucaje fotográfico de cómo serían las pantallas más significativas de la web ya finalizada.

2. Fase de implementación Java

Para la creación del proyecto web, y todo lo referente al dinamismo de la página codificado en Java se utiliza el Netbeans, se trata de uno de los principales entornos de programación para Java, es gratuito y tiene muchas herramientas de “ayuda” a la hora de realizar proyectos Java . Este programa se enlaza directamente con el servidor web Apache Tomcat e integra perfectamente tanto las aplicaciones Java, las cuales se compilan desde el mismo entorno, como las paginas JSP. Otras características importantes son su buen funcionamiento bajo Windows y la posibilidad de compresión del proyecto en un archivo .war.

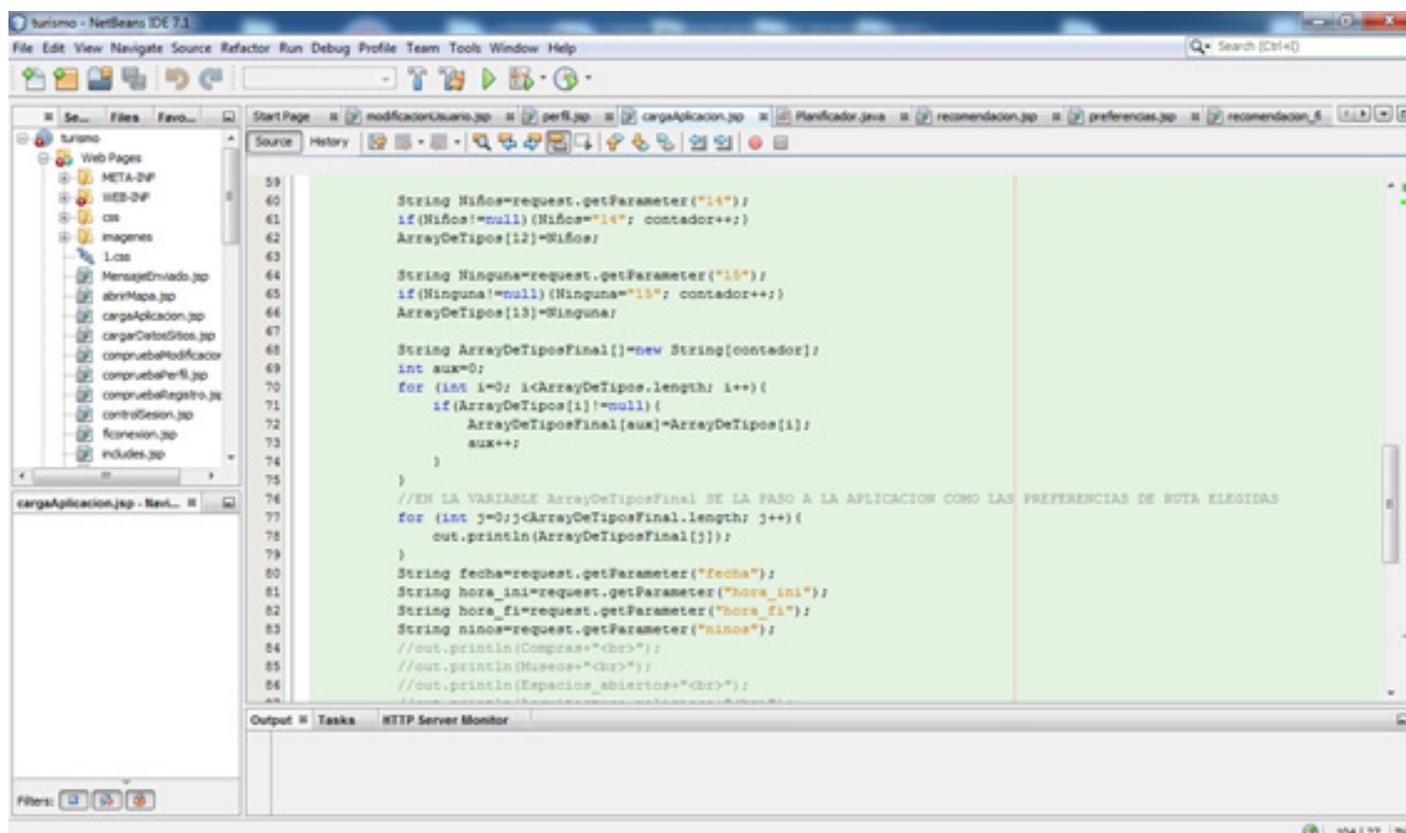


Fig 16. Netbeans

3. Fase de pruebas

3.1 - Apache Tompcat

La página web puede ser alojada por cualquier servidor con soporte para la ejecución de contenedores Java. Para comprobar el correcto funcionamiento de la web, se monta en un servidor Tompcat versión 7.0 y se prueba en local.

3.2- servidor Sqlplus

Una de las partes más importantes de la web, que gestiona todo lo referente a la información de los usuarios, los sitios... es la base de datos. Para poder probar la página se monta un servidor de Mysql y en él se carga la base de datos.

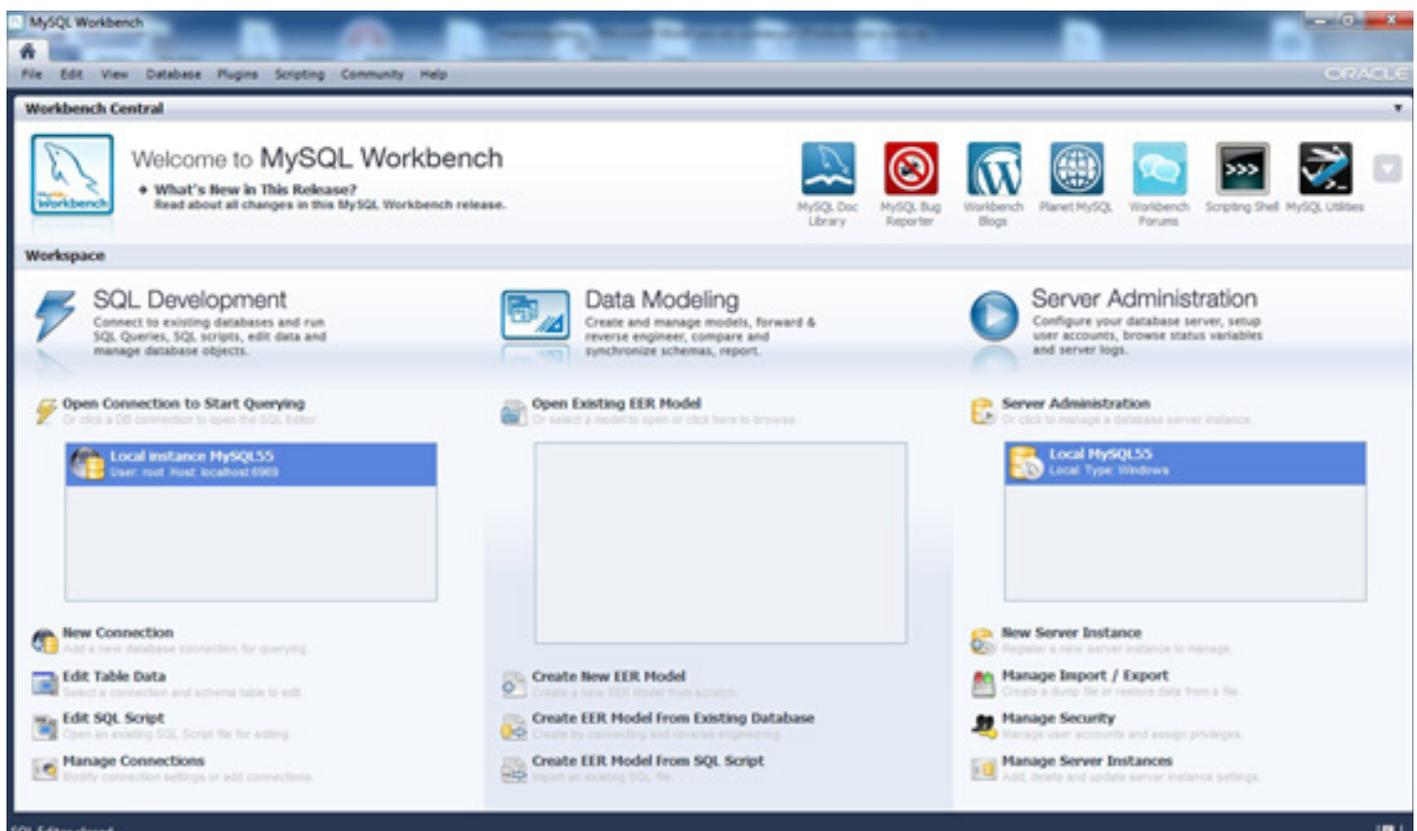


Fig 17. MySQL Workbench

3.3- Bluej

La web utiliza dos aplicaciones Java , una que se encarga de enviar por mail la clave perdida por el usuario y la aplicación principal que es el recomendador de rutas. Para comprobar que las aplicaciones compilan correctamente y no tienen excepciones en modo ejecución se prueban con este sencillo entorno de programación Java. Este proceso no es realmente necesario, ya que el Netbeans también tiene compilador de aplicaciones Java, pero para comprobar el funcionamiento de las aplicaciones, se crean y se destruyen constantemente clases Java de prueba y como cuando estas son depuradas desde el Netbeans, el programa crea los .class en sus carpetas correspondientes y las enlaza en el fichero web.xml. Se considera esta la opción más limpia al asegurarse que no queda ningún tipo de información residual en el proyecto.

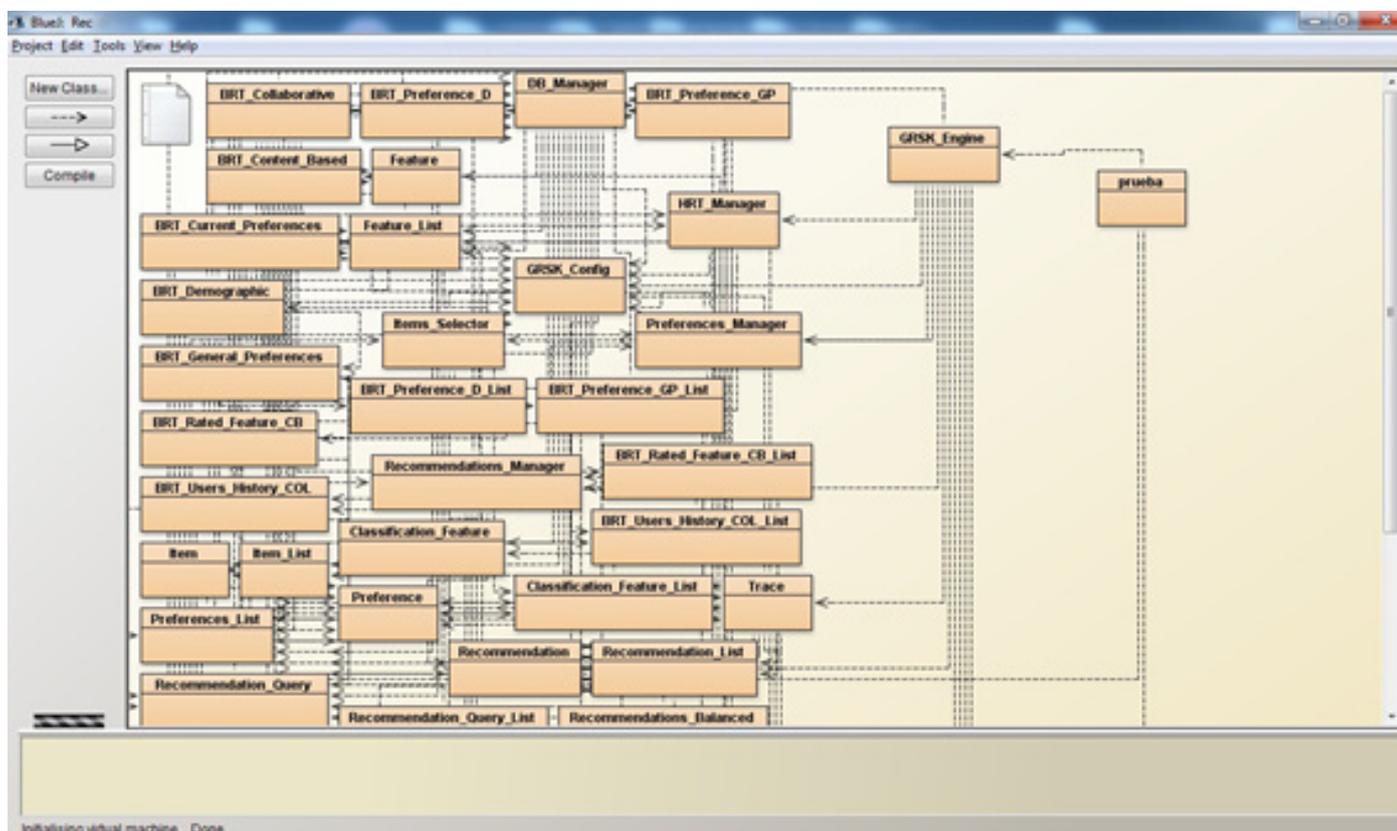


Fig 18. Bluej

El sistema operativo utilizado para la ejecución de los programas explicados anteriormente es Windows 7.

Como se ha visto en el presente proyecto, he realizado una página web como interfaz de un Sistema de Recomendación para el cálculo de rutas por la ciudad de Valencia. La página web además de presentar gráficamente la ruta al usuario, se encarga de gestionar los datos de los usuarios en la base de datos y de ejecutar la aplicación de Recomendación.

Las pruebas realizadas han puesto de manifiesto el buen funcionamiento de la misma, realizando el proceso de validación o registro, modificación de datos de perfil y cálculo de ruta con unos buenos tiempos de carga. Además he realizado la interfaz de forma sencilla e intuitiva para hacérselo lo más fácil posible al usuario.

De cara a futuras versiones se contemplan algunas mejoras como:

- 1- La creación de un apartado en la web para que permita al usuario reportar posibles fallos o mejoras.
- 2- Inserción del campo “Preferencia de transporte”, que contendrá los tres posibles del API de Google Maps (Caminando, Coche particular o Transporte público).
- 3- Creación de una nueva página con los lugares que el el usuario ha visitado ya, donde se le dará la posibilidad de valorarlos para poder crear un ranking.

- Guia online de programación utilizando jsp, servlets y MySql
web: http://www.programacionfacil.com/java_jsp/start
- Foro general de programación web
web: <http://www.desarrolloweb.com>
- Artículos sobre diferentes tecnicas de diseño web
web: www.programacion.com
- Manuales de codigo HTML
web: <http://www.quackit.com/html/>
- Libro: Java for the Web with Servlets, JSP, and EJB 992 páginas
Editor: New Riders Publishing; Edición: Pap/Cdr (12 de abril de 2002)
Colección: Landmark (New Riders)
Idioma: Inglés
- Libro: My SQL and JSP Web Applications 576 páginas
Editor: Sams Publishing; Edición: 0004 (27 de marzo de 2002)
Colección: Kaleidoscope
Idioma: Inglés
- Libro: MySQL Workbench: Data Modeling & Development 480 páginas
Editor: McGraw-Hill Osborne Media; Edición: 1 (3 de mayo de 2013)
Idioma: Inglés
- Web de turismo:
<http://www.turisme.gva.es/opencms/opencms/turisme/es/files/pdf/recomendación/DocumentoBasePEGTCV.pdf>
- Mapas de google:
<http://www.desarrolloweb.com/articulos/controles-mapas-google.HTML>



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

PROYECTO FINAL DE CARRERA | 2013
Javier Crespo Tapia