



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA TÉCNICA INFORMÁTICA DE GESTIÓN

MyPaths

**GESTIÓN DE RUTAS CATEGORIZADAS EN LA NUBE USANDO
GWT, DATASTORE, APPENGINE Y API GOOGLE MAPS GWT**

**Realizado por
MIGUEL CENTENO MUÑOZ
28633254W**

**Dirigido por
SERGIO SEGURA RUEDA**

**Departamento
LENGUAJES Y SISTEMAS INFORMÁTICOS**

Sevilla, Septiembre de 2014

Resumen

EL origen de esta idea surgió como una necesidad particular de un familiar quien, planificando un viaje turístico, necesitaba de la experiencia de personas que ya hubieran estado en diferentes lugares de España. Dicha necesidad se convirtió³ en idea de negocio.

Entre las necesidades que busca cubrir esta aplicación se encuentran dos preguntas que frecuentemente abordan a las personas en la planificación de sus actividades favoritas o el sacar partido a su tiempo libre: ¿Qué hacer y qué ofrece un destino concreto? y, sobre todo, ¿Cómo puedo saber si una actividad en concreto que consulto en la red es realmente lo que busco y cubre mis necesidades?

Posteriormente a esta primera interpretación del problema, surge otro factor determinante en la funcionalidad de *MyPaths*: Ofrecer alternativas, dentro de los gustos de los usuarios, cerca de la zona en la cual residen e, incluso, cerca de la geolocalización en la que se encuentran en un momento concreto.

MyPaths es una aplicación que busca cubrir las necesidades de los usuarios para realizar actividades categorizadas, presentarlas haciendo uso de las posibilidades de *Google Maps*, y, sobre todo, poder compartir opinión sobre dichas actividades así como poder consultar las opiniones de usuarios o amigos que ya las conozcan.

Los usuarios de *MyPaths* pueden crear sus propias actividades, gestionarlas, compartirlas con grupos de amigos (tanto privadas como públicamente) y, sobre todo, ofrece a los usuarios la posibilidad de organizarse para realizar sus actividades favoritas, así como sacar el máximo provecho a su tiempo libre.

Para el desarrollo de *MyPaths* se ha tenido muy en cuenta la posibilidad de que la aplicación sea escalable, tenga un posible despliegue multidispositivo en el futuro, y su código sea mantenible en el tiempo.

Entre las tecnologías y herramientas usadas con este fin se encuentran Projetsii, GWT(Google Web Toolkit), Google App Engine, API Google Maps para GWT, Git, Sonar y JUnit.

Tabla de contenidos

Resumen	2
---------------	---

Parte I.- Introducción y planificación	7
--	---

1.- Introducción	9
------------------------	---

1.1.- Motivación	9
1.2.- Objetivos del proyecto	10
1.2.1.- Objetivos docentes	10
1.2.2.- Objetivos técnicos	10
1.3.- Desarrollo del proyecto.....	10
1.3.1.- Proyectosii	10
1.4 Estructura del documento.....	11

2.- Planificación	12
-------------------------	----

2.1.- Planificación temporal	12
2.1.1. Estimación de tiempos	12
2.1.2.- Desviación respecto estimación de tiempos	13
2.2.- Planificación de costes	13
2.2.1.- Estimación de costes.....	13
2.2.2.- Desviación respecto estimación de costes	13

Parte II.- Materias relacionadas.....	14
---------------------------------------	----

3.- Materias relacionadas.....	16
--------------------------------	----

3.1.- Google Web Toolkit (GWT)	16
3.2.- AJAX y cómo GWT hace uso de él	18
3.2.1.- ¿Por qué AJAX?	18
3.2.2.- ¿Qué hace GWT por JavaScript y AJAX?.....	18
3.3.- RPC como herramienta de comunicación Cliente-Servidor.....	19
3.4.- MVP aplicado a Gwt: No terminado	21
3.5.- Datastore y Google App Engine (uso de JDO)	22
3.6.- Control de versiones	22
3.6.1.- Git.....	22
3.6.2. Comandos básicos de Git	23

3.6.3.- Uso de Git.....	24
Parte III.- Sistema desarrollado	27
4.- Dominio del problema	29
4.1.- Descripción del problema	29
4.2.- Descripción del sistema actual.....	30
4.3.- Glosario de términos.....	30
5.- Visión general del sistema desarrollado.....	30
5.1.- Participantes en el proyecto	30
5.1.1.- Participantes	30
5.1.2.- Organizaciones.....	30
5.2.- Objetivos del sistema	31
6.- Elicitación de requisitos	32
6.1.- Requisitos de información	32
6.2.- Reglas de negocio y restricciones	33
6.3.- Requisitos funcionales	34
6.3.1 Diagrama de subsistemas del sistema	34
6.3.2.- Diagramas de casos de uso	34
6.3.2.- Definición de actores	35
6.3.3.- Casos de uso del sistema	36
6.3.- Requisitos no funcionales	40
6.4.- Matriz de rastreabilidad requisitos-objetivos.....	40
6.5.- Matriz de rastreabilidad casos de uso-objetivos	40
7.- Análisis de requisitos.....	40
7.1.- Modelo estático del sistema	40
7.1.1.- Diagramas de tipos.....	40
7.1.2.- Escenario de prueba.....	40
7.1.3.- Tipos.....	40
7.2.- Modelo dinámico del sistema	40
7.2.1.- Subsistema de gestión de usuarios	40
7.2.2.- Subsistema de gestión de rutas	40
7.3.- Prototipos de interfaz de usuario	40

7.3.1.- Pantalla de inicio y búsqueda	41
7.3.2.- Pantalla de login.....	41
7.3.3.- Pantalla de registro de usuario	42
7.3.4.- Pantalla Mis Rutas.....	43
7.3.5.- Pantalla Añadir Ruta	44
7.3.6.- Pantalla Editar Perfil.....	45
7.3.7.- Pantalla Búsqueda (Usuario logado).....	46
7.3.8.- Pantalla Resultados Búsqueda (Usuario logado)	46
7.3.9.- Pantalla Resultados Búsqueda (Usuario NO logado)	47

Parte I.- Introducción y planificación

1.- Introducción

En la actualidad, las personas cada vez gozan de un tiempo libre más limitado debido principalmente a las situaciones laborales y personales de cada individuo. Desde hace algún tiempo, cada vez es más frecuente que las personas hagan uso de servicios ofrecidos por las diferentes tecnologías y dispositivos que nos rodean para organizarse y planear dicho tiempo libre en base a los gustos y aficiones particulares.

Dicho tiempo libre es muy utilizado para hacer deporte (jogging, mountain bike...), viajar y hacer turismo, conocer la cultura y gastronomía de diferentes puntos del mundo, acudir a espectáculos y eventos y todo tipo de actividades.

1.1.- Motivación

La idea surge de un familiar, muy aficionado tanto al jogging como a la mountain bike, cuando se encontraba planificando sus vacaciones y descubrió que existía un vacío a la hora de poder disfrutar del turismo, conocer una ciudad, su gastronomía y, a la vez, poder hacer deporte en lugares que fueran realmente interesantes, de modo que pudiera hacer su tiempo más provechoso.

En *MyPaths* se busca, haciendo uso de la geolocalización y de las posibilidades ofrecidas por el plugin de Google para Eclipse (principalmente su librería Google Maps), representar en un mapa rutas y senderos con información detallada, estando englobadas en diferentes categorías.

MyPaths ofrece la posibilidad a un usuario de crear sus rutas personales y que, tanto el propio usuario como sus círculos de amistades dentro de la aplicación, puedan disfrutar de actividades diversas cercanas al lugar donde se encuentran.

Como consecuencia de lo anterior, me decidí por utilizar esa necesidad patente como un proyecto emprendedor.

La idea es desarrollar la aplicación y, mediante el uso de herramientas en redes sociales y posicionamiento online, hacer que llegue al máximo de usuarios posibles. Una vez alcanzada cierta masa de usuarios (previo a estudio de mercado/comparación competencia), así como cierta cantidad de rutas en las principales ciudades de España, comenzar a incluir publicidad geolocalizada. Obviamente, la consecución de dicha publicidad supone un fuerte esfuerzo comercial que habría que evaluar. Pongamos un ejemplo: En la Sierra Norte de Sevilla hay 15 rutas de mountain bike en nuestra aplicación. Pues es obvio que a una tienda especializada en mountain bike en la ciudad de Sevilla le aporta beneficios aparecer en nuestra app cada vez que alguien busca o ve dichas rutas.

Otra motivación importante es el uso que el framework GWT (Google Web Toolkit) puede aportarme en el desarrollo continuo de mi carrera y mis conocimientos. GWT aporta múltiples funcionalidades orientadas a la integración de servicios en el desarrollo de aplicaciones web, haciendo el uso de AJAX mucho más sencillo, debido al uso de Java que, como ya es conocido, es un lenguaje robusto y continuamente actualizado. Con GWT se puede desarrollar bajo cualquier IDE de java siendo el código compilado a JavaScript y HTML.

1.2.- Objetivos del proyecto

1.2.1.- Objetivos docentes

- Uso de metodologías estudiadas para la planificación, diseño y ejecución de un proyecto software
- Practicar el proceso de planificación y organización de un proyecto software, fundamental para un ingeniero informático.
- Uso de tecnologías no estudiadas durante la etapa universitaria que puedan aportarme en mi carrera profesional.
- Afianzar el uso de Java como lenguaje y plataforma de programación para proyectos software.

1.2.2.- Objetivos técnicos

- Utilización de patrones de diseño y análisis de las ventajas del uso de los mismos durante el desarrollo del proyecto.
- Estudiar y profundizar en el uso de tecnologías en las cuales no se profundiza durante la carrera, y hacer uso de ellas para el desarrollo del proyecto.
- Hacer simulación profesionalizada del ciclo de vida del software y funciones de consultoría del software, análisis de costes y de tiempos.
- Ser capaz de adaptar el proceso de desarrollo de un proyecto software a necesidades reales de clientes.

1.3.- Desarrollo del proyecto

La idea a desarrollar surge como consecuencia de las primeras reuniones entre el tutor y el alumno. En la decisión tomada sobre el proyecto a desarrollar se unen necesidades reales de conocidos (usuarios-clientes) con el desarrollo de una aplicación web con integración de servicios y la búsqueda de mejorar la experiencia con tecnologías bastante demandadas en la actualidad entre profesionales del sector de las TIC.

Las reuniones entre tutor y alumno se han ido sucediendo de manera muy habitual con la intención de concretar objetivos y generar tareas a realizar entre las mismas. A las tareas se les ha asignado un tiempo de investigación y ejecución razonables, tiempo tenido en cuenta a la hora de realizar la estimaciones temporales en el desarrollo del proyecto. En las reuniones posteriores se decide las modificaciones necesarias a realizar además de las nuevas tareas.

1.3.1.- Projeetsii

Esta herramienta ofrecida por la Universidad de Sevilla se ha utilizado para la gestión de tareas. Las tareas se añaden al sistema asignándole un número de horas estimado y, posteriormente, se le

asignan las horas reales utilizadas en el desempeño de las mismas. Esto se utiliza posteriormente en el apartado X.X. de la documentación para la estimación temporal del proyecto.

Para el resto de pasos o etapas superadas para el desarrollo del proyecto, se indica el proceso seguido para la realización de cada apartado o sección en su apartado correspondiente en este documento.

1.4 Estructura del documento

Para una mejor comprensión de los pasos seguidos en el desarrollo del proyecto, se resume en este apartado las partes de las que la documentación consta:

PARTE I – Introducción y planificación. En esta primera parte, se resumen las motivaciones y los orígenes de la idea a desarrollar, así como se realiza una introducción a la metodología a seguir en su desarrollo, estimación de los tiempos y estimación de los costes de llevar a cabo el proyecto.

PARTE II – Materias relacionadas. En la segunda parte, se detallan las tecnologías empleadas en el desarrollo del proyecto. (MÁS)

PARTE III – Sistema desarrollado. Esta es la parte principal del documento, en ella se incluye: documentación de la aplicación desarrollada: elicitación y análisis de requisitos, diseño, implementación, pruebas y manuales.

PARTE IV – Comentarios y conclusión.

Memoria

Parte I – Introducción y planificación

- 1.- Introducción
- 2.- Planificación

Parte II – Materias relacionadas

- 3.- Tecnologías Web

Parte III – Sistema desarrollado

- 4.- Dominio del problema
- 5.- Visión general del sistema
- 6.- Elicitación de requisitos
- 7.- Análisis de requisitos
- 8.- Diseño
- 9.- Implementación
- 10.- Pruebas
- 11.- Manual de instalación
- 12.- Manual del usuario

Parte II – Materias relacionadas

- 13.-Cumplimiento de los objetivos del proyecto
- 14.- Conclusiones

2.- Planificación

2.1.- Planificación temporal

La planificación temporal consiste en la distribución de tiempos estimados para la realización de determinados conjuntos de tareas (los llamaremos HITOS), de manera que se consigue una mejora en la estimación temporal para el desarrollo del proyecto en su conjunto.

En todas las funciones del mundo real en las que se trabaja mediante proyectos, la minimización de los costes y, por ende, de los tiempos de desarrollo de los mismos es fundamental, además de ser una cualidad muy valorable para un buen ingeniero de software.

En el caso de este proyecto en cuestión, la tarea es algo más sencilla pues el equipo de desarrollo consta de un único miembro, ahorrándonos la planificación para desarrollo de tareas de forma paralela.

El uso de tecnologías, frameworks y herramientas, que no son conocidas en profundidad por el autor del proyecto, dificulta la tarea pues tiene un trabajo importante previo de documentación y prácticas con las mismas.

La división del trabajo a realizar en hitos queda definido de la siguiente manera:

HITO	DESCRIPCIÓN
H1	Estudio y prácticas realizadas con las tecnologías a utilizar
H2	Documento de elicitación de requisitos
H3	Documento de análisis de requisitos
H4	Documento de diseño
H5	Desarrollo de la aplicación
H6	Instalación e implantación final de la aplicación
H7	Fin de proyecto

2.1.1. Estimación de tiempos

La fecha en la que se comenzó la realización del proyecto fue en torno al 15 de Marzo de 2014, y la fecha fin del mismo es en el mes de Septiembre, a principios del mismo, pero debido a que el mes de agosto no es lectivo, se estimará la fecha de finalización para documentación, desarrollo, pruebas e implantación en torno al 30 de Julio.

La estimación en horas de trabajo para el único miembro del equipo de trabajo será de **3 horas al día 5 días a la semana**, lo que hace un total de **15 horas semanales**. Estas 15 horas semanales es fruto de una ponderación, pues hay fechas en las que la capacidad de dedicación en horas es inferior a dicha fecha, así como hay épocas en las que es bastante mayor, haciendo uso de horas en fin de semana.

Otro factor a tener en cuenta es la cantidad de horas mínima que el PFC tiene como asignatura de Ingeniería Técnica Informática de Gestión, que son **225 horas**.

En la tabla que a continuación se presenta, realizamos una división del desarrollo del proyecto en tareas y una estimación de horas divididas por cada una de ellas.

TAREA	DÍAS
T1.- Estudio del problema y prácticas previas realizadas	10
T2.- Familiarización con el entorno de desarrollo y herramientas	71
T3.- Desarrollo de la aplicación	
T3.1. Elicitación de requisitos	15
T3.2. Análisis de requisitos	15
T3.3. Diseño del sistema	5
T3.4. Implementación de la aplicación	30
T3.5. Pruebas Unitarias	3
T3.6. Implantación del sistema	3
T4.- Documentación del proyecto	14
T5.- Manual de usuario	5
TOTAL	100 días

Según el reparto de días en función a tareas realizado anteriormente, nos salen 120 días.

Como ya hemos comentado, asignamos 3 horas/día, quedándonos una dedicación total en horas:
100 días x 3 horas/día = 300 horas.

Diagrama de gantt: TOMSPANNER

2.1.2.- Desviación respecto estimación de tiempos

Pendiente

2.2.- Planificación de costes

Pendiente

2.2.1.- Estimación de costes

Pendiente

2.2.2- Desviación respecto estimación de costes

Pendiente

Parte II.- Materias relacionadas

3.- Materias relacionadas

En el capítulo tercero, se repasan las tecnologías y herramientas de las que se ha hecho uso durante el desarrollo de *MyPaths*. En ellas presentaremos brevemente sus características y las ventajas que ofrece su uso.

3.1.- Google Web Toolkit (GWT)



Google Web Toolkit (GWT) es un framework, un conjunto de APIs y widgets que permiten desarrollar aplicaciones AJAX en Java para después compilar el código fuente a Javascript de manera muy optimizada. Este código JavaScript generado corre perfectamente en todo tipo de navegadores web incluyendo navegadores móviles de Android o iOS.

Una ventaja que tiene GWT es que, a pesar de estar basado en una serie de Widgets en su capa más alta de presentación, permite la inserción de código Javascript estándar por parte del usuario. Digamos que no limita su uso a sus puntos fuertes, dejando hueco a la creatividad del desarrollador.

Otra de las principales ventajas que ofrece GWT es la posibilidad de hacer depuración en código Javascript, debido a que se implementa en Java, además de que puede ser usado mediante un IDE (Eclipse en nuestro caso) con las ventajas y comodidades que ofrece.

En el momento del despliegue de la aplicación es cuando el código desarrollado en Java es compilado mediante el compilador GWT generando un fichero Javascript y ficheros HTML.

Las aplicaciones desarrolladas en GWT pueden ser, a su vez, ejecutadas de dos modos diferentes: *Dev-Mode* o en modo desarrollo, ejecutándose como código bytecode de Java dentro de la Máquina Virtual de Java (JVM) o en modo Web, ejecutándose la aplicación como código JavaScript y HTML puro.

Probablemente, la ventaja de mayor peso de todas las que ofrece este framework es la siguiente: GWT resuelve, de manera sencilla, la parte del cliente y de comunicación remota usando un contenedor de Servlets en el lado del cliente. Además, GWT es sencillo de combinar con tecnologías maduras de Java, como pueden ser JPA, EJB, Spring Framework...

Google ofrece desde hace bastante tiempo, un plugin oficial para eclipse llamado Googl Plugin for Eclipse. Con él en sencillos pasos podemos dar comienzo al desarrollo de un proyecto GWT con Eclipse.

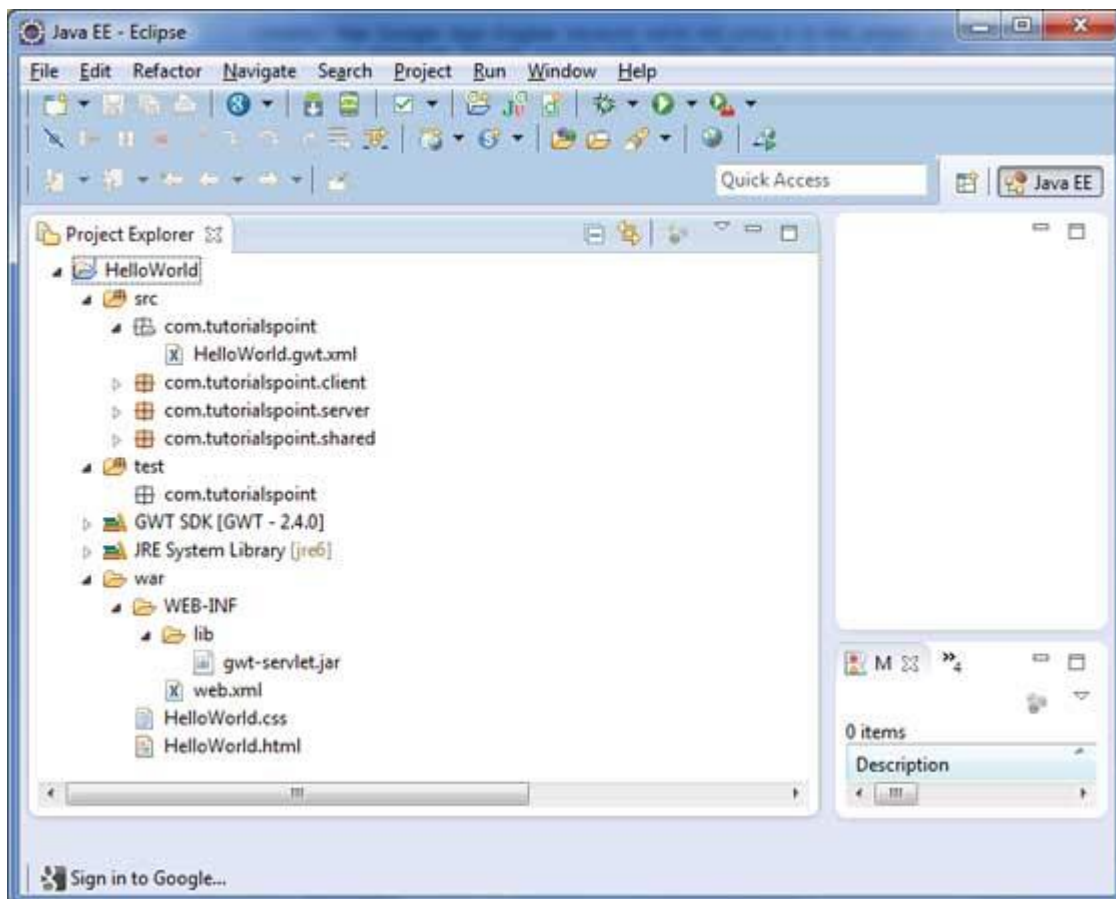


Imagen 1: Estructura de un proyecto GWT en Eclipse.

En el paquete contenedor del código Java (src) categorizamos el código en 3 paquetes básicos: client, shared, server.

Client contiene el código del lado del cliente, shared contendrá clases útiles en ambas direcciones, y server contendrá clases con métodos que se ejecutarán del lado del servidor, así como el mapeo de la Base de Datos.

Otros componentes a destacar en la estructura de un proyecto base GWT:

Módulo XML (cuyo nombre será: *MyPaths.gwt.xml*). En él se definen el módulo GWT, los recursos del proyecto o shared package y también define cuál es la clase EntryPoint.

Página host (*MyPaths.html*). Página que ejecuta la aplicación. Referencia las hojas de estilos CSS. También referencia el código JavaScript que será un mezcla de elementos estáticos y dinámicos.

Css de la aplicación.

La Clase Entrypoint (más correctamente, clase que implementa EntryPoint) es una clase con el nombre de la APP o módulo y que contiene los widgets y elementos a utilizar por la misma.

En GWT la comunicación entre el código que se ejecutará en el lado del cliente, y el que lo hará en el lado del servidor, se realiza mediante llamadas RPC (Remote Procedure Calls).

3.2.- AJAX y cómo GWT hace uso de él

AJAX es el acrónimo de Asynchronous JavaScript and XML. Es un conjunto de técnicas que incluyen JavaScript, CSS, llamadas asíncronas HTML y que ofrecen en su conjunto una experiencia mejorada para el usuario de una aplicación software en el lado del cliente.

AJAX se aplica hoy día a cualquier aplicación construida bajo estándares abiertos y comunicación (cliente-servidor) asíncrona. Las aplicaciones desarrolladas bajo AJAX con responsive y parecen aplicaciones de escritorio en su funcionamiento a nivel cliente.

Con AJAX las interfaces de usuario son muy mejoradas y toman importancia en el software. La masa de usuarios es enorme y, por ende, la documentación y referencias.

3.2.1.- ¿Por qué AJAX?

- Porque su núcleo es JavaScript, lo que lo hace muy potente.
- Es una técnica de desarrollo, no un lenguaje.
- Puede usarse JavaScript nativo para mejorar el trabajo con AJAX.
- Recorta en gran medida el uso de peticiones HTTP así como el tráfico de datos cuando se compara con aplicaciones sólo HTML.
- Implementación de JavaScript por parte de los navegadores no es consistente, aunque mejora mucho en estos últimos años.
- Hitos JavaScript: Aprenderlo y manejarlo con soltura, manejo de la seguridad y la memoria.

3.2.2.- ¿Qué hace GWT por JavaScript y AJAX?

- GWT permite codificar el lado del cliente en Java.
- GWT compila en Java y genera JavaScript -> Es cargado por el navegador.

Hitos de las Apps AJAX	Resolución de GWT
Aprender y dominar JavaScript	GWT usa Java para desarrollar apps AJAX
Fácil pérdida de memoria de apps JavaScript	GWT genera JavaScript optimizado que limita o elimina este problema
Difícil usar marcadores en apps JavaScript	GWT ofrece soporte completo de histórico de navegación, incluidos marcadores.
Apps AJAX difíciles de depurar	Apps GWT codificadas en Java, pudiéndose usar depuradores para Java, incluido el que viene por defecto con Eclipse.
Apps JavaScript suelen ser de gran tamaño. Su inicialización es muy lenta.	GWT divide y encapsula apps AJAX dentro de trozos que sólo son descargados cuando se necesitan
Los errores de sintaxis en JavaScript sólo pueden descubrirse en tiempo de ejecución	GWT es Java, por lo que no permite errores de sintaxis al compilar.
Sintaxis diferente para CSS y JavaScript en cada navegador	GWT contiene dichas diferencias y genera código diferente para navegadores diferentes.

3.3.- RPC como herramienta de comunicación Cliente-Servidor

Las aplicaciones tradicionales (o aplicaciones que no usan AJAX), cuando realizan una petición de información al server, bloquean la interfaz de usuario (IU) hasta recibir respuesta por parte de este. Una vez recibida la respuesta, proceden a actualizar la IU. Esto es lo conocido como llamadas síncronas, al suceder los procesos en secuencias cronológicas, una cosa cada vez.

GWT realiza llamadas asíncronas al server, lo que hace que no sean bloqueantes y que puedan realizarse varios procesos en paralelo. Estas llamadas son las llamadas RPC o Remote Procedure Calls.

Las ventajas del uso de esta herramienta de comunicación Cliente-Servidor son:

- Mejor experiencia de usuario. La interfaz de usuario no se bloquea al operar y puede realizar otras tareas mientras espera respuesta.
- La recepción de las respuestas se realiza mediante un método callback.
- Permite compartir clases de datos (véase DTO's) entre el cliente y el servidor.

La implementación de las llamadas asíncronas es más compleja que las llamadas síncronas, pero GWT agiliza este proceso:

- 1) Creación en el paquete `com.nombreapp.client` de interfaz que extienda `RemoteService`

```
@RemoteServiceRelativePath("userAccountService")
public interface UserAccountService extends RemoteService{
    public UserAccountDTO login(String email, String Password);
}
```

- 2) Creación, en el paquete `com.nombreapp.server` de una clase que implemente dicha interfaz y extienda `RemoteServiceServlet`.

```
public class UserAccountServiceImpl extends RemoteServiceServlet
implements UserAccountService{
    public UserAccountDTO login(String email, String password){
        new AppMisc().populateDataStoreOnce();
        return UserAccount.toDTO(UserAccount.getDefaultUser());
    }
}
```

- 3) Informar al compilador de la existencia del Servlet en `\WEB-INF\web.xml`

```

...
<servlet>
    <servlet-name>UserAccountService</servlet-name>
    <servlet-class>
        com.myApp.server.UserAccountServiceImpl
    </servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>UserAcoountService<servlet-name>
    <url-pattern>/myApp/userAccountService</url-pattern>
</servlet-mapping>
...

```

4) Crear interfaz asíncrona en com.nombreapp.client

```

public interface UserAccountServiceAsync{
    void login(String email, String password,
        AsyncCallback<UserAccountDTO> callback);
}

```

5) Llamada/Uso del servicio creado

```

private void login(){
    userService.login("Email", "password",
        new AsyncCallback <UserAccountDTO>() {
            public void onFailure(Throwable caught){
                Window.alert("An error occurred");
            }
            Public void onSuccess(UserAccountDTO user){
                currentUser = user;
                displayApp();
            }
        }
    }
}

```

3.4.- MVP aplicado a Gwt: **No terminado**

A la hora de desarrollar una aplicación en GWT tenemos la posibilidad de usar el patrón MVP (modelo vista presentador). ¿Qué ventajas tiene su uso?:

1º desacopla el desarrollo en una dirección permitiendo trabajar a varios desarrolladores a la vez.

2º minimiza el uso de GWTTestCases (se realizan en navegador, siendo más costosas) y permite la realización de pruebas de software basadas en JRE (véase Junit) lo cual es más rápido y eficiente, desechando el uso de un navegador para testear el software.

A la hora de usar el patrón MVP, existen dos conceptos fundamentales:

a) Se particiona la aplicación en componentes:

Model

View

Presenter

AppController

b) Interactuación de los componentes:

Enlazar Presenter – Views

Eventos y Bus de Eventos

Histórico y transición entre las vistas.

Pruebas (testing)

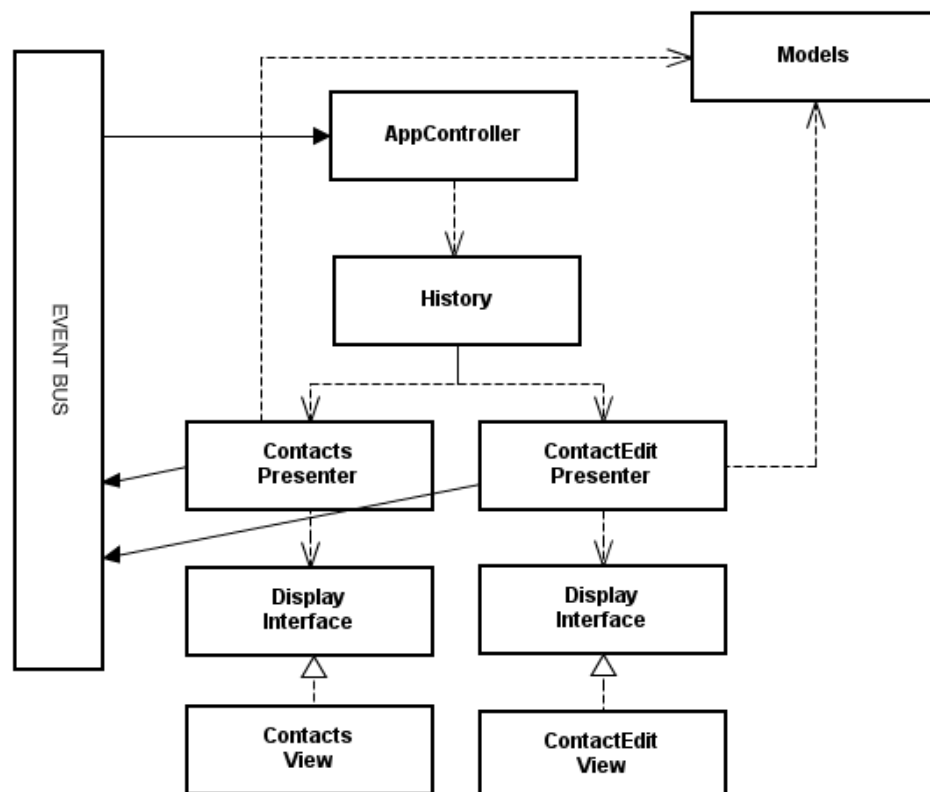


Imagen 2: Diagrama patrón MVP en Gwt en App de prueba de gestión de contactos.

3.5.- Datastore y Google App Engine (uso de JDO)

Pendiente de pasar. Documento creado.

3.6.- Control de versiones

En este apartado se presenta la herramienta usada para la gestión del código fuente de la aplicación así como el control de versiones.

Un sistema de control de versiones gestiona los cambios que se realizan sobre el código de un programa o aplicación, así como de los ficheros de configuración del mismo. Con los sistemas de control de versiones podemos también controlar las modificaciones o actualizaciones que se realizan sobre el código.

La alternativa elegida para la gestión del código y control de versiones es Git.

3.6.1.- Git



Git es un sistema de control de versiones distribuido (scvd) escrito en C que permite la creación de una historia o histórico para una colección de archivos e incluye la funcionalidad de revertir la colección a otro estado. “Otro

estado” es cualquiera de las dos opciones siguientes: a) una colección de archivos diferente b) un contenido diferente de la colección de archivos.

En el caso de un proyecto de desarrollo de software, una colección de archivos la componen el código fuente de la misma, así como los ficheros de configuración de la aplicación.

Que Git sea un sistema de control de versiones distribuido quiere decir que no requiere un repositorio central para los ficheros, como sucedería en el caso de subversión, sino que se crea siempre una copia local del código en cada dispositivo conectado al mismo.

Los cambios en el código (añadir, eliminar o modificar/actualizar ficheros) los realizo sobre esa copia local que tenemos. Posteriormente se harán esos cambios relevantes al control de versiones (se añadirán los ficheros al *Staging Index*). Posteriormente, los cambios se realizarán haciendo *commit*.

Características principales de *Git*:

- Los cambios en *Git* se realizan al repositorio local
- Luego, pueden sincronizarse los cambios con otros repositorios.
- *Git* permite clonar repositorios completos, incluido el histórico del mismo
 - *Push*: Transfiere los cambios al repositorio remoto
 - *Pull*: Obtiene cambios del repositorio remoto
- Soporte para *branching*. Esto quiere decir que, dentro del repositorio, se pueden tener diferentes versiones del proyecto, de manera que, por ejemplo, puede crearse un *branch* para modificar cierta parte del código sin afectar al proyecto, y una vez comprobado su funcionamiento, puede hacerse *merge* (unir los *branches*).

- Varias implementaciones: puede usarse por línea de comandos o a través de editores gráficos.

3.6.2. Comandos básicos de Git

<code>git init:</code>	Al usar <code>git init</code> , transformamos el directorio en el que nos encontramos en un repositorio <code>git</code> , comenzando a registrar versiones del proyecto.
<code>git init --bare:</code>	Inicializa repositorio <code>git</code> vacío, pero omitiendo el directorio de trabajo. Así imposibilitamos la edición/actualización de ficheros. Útil para crear repositorios centrales, marcándolo como un almacén. A partir de ahí pueden crearse ramas (<i>branches</i>) sí editables.
<code>git clone:</code>	Clona un repositorio <code>git</code> ya existente. Puede equipararse al comando <code>svn checkout</code> . La copia creada localmente es completamente aislado del repositorio remoto, con sus propios ficheros, y su propia historia de versiones. Al clonar se crea el origen, o conexión remota que apunta a repositorio original.
<code>git config:</code>	Configura la instalación de <code>git</code> desde la línea de comandos. Define desde información de usuario hasta preferencias de comportamiento de un repositorio.
<code>git add:</code>	<p>Añade un cambio en el directorio de trabajo al área de preparación. Le dice a Git que quieres incluir actualizaciones en un archivo en particular en el siguiente commit. Sin embargo, <code>git add</code> no afecta al repositorio de forma importante, pues los cambios no se guardan hasta que se ejecuta <code>git commit</code>.</p> <p>Junto con estos comandos, también es necesario <code>git status</code> para ver el estado del directorio de trabajo y del área de preparación.</p>
<code>git commit:</code>	El comando <code>git commit</code> confirma la instantánea preparada a la historia del proyecto. Las instantáneas confirmadas pueden considerarse versiones "seguras" de un proyecto—Git nunca las modificará a menos que lo pidas explícitamente. Junto con <code>git add</code> , este es uno de los comandos más importantes de Git.
<code>git status:</code>	El comando <code>git status</code> muestra el estado del directorio de trabajo y del área de preparación. Permite ver qué cambios se han preparado, cuáles no, y qué archivos no llevan seguimiento de Git. La salida del comando <i>no</i> muestra ninguna información sobre la historia del proyecto confirmada. Para esto, se ha de usar <code>git log</code> .
<code>git log:</code>	El comando <code>git log</code> muestra las instantáneas confirmadas. Permite hacer un listado de la historia del proyecto, filtrarlo y buscar cambios específicos. Mientras que <code>git status</code> permite inspeccionar el directorio de trabajo y el área de preparación, <code>git log</code> solo opera en la historia confirmada.

3.6.3.- Uso de Git

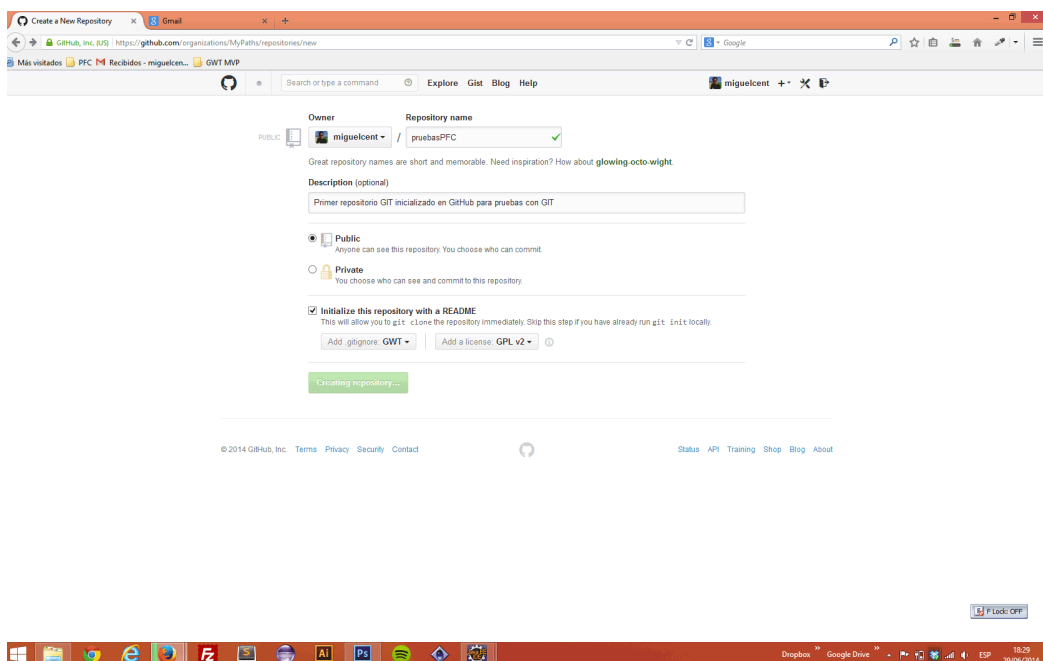


GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails.

El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

En este apartado mostraremos cómo se crea un proyecto Git usando Egit (plugin que implementa la funcionalidad de Git y que es contenido en las instalaciones base de Eclipse Java EE).

El primer paso es crear nuestro perfil en GitHub y, a la vez, se crea un repositorio remoto asociado a él.

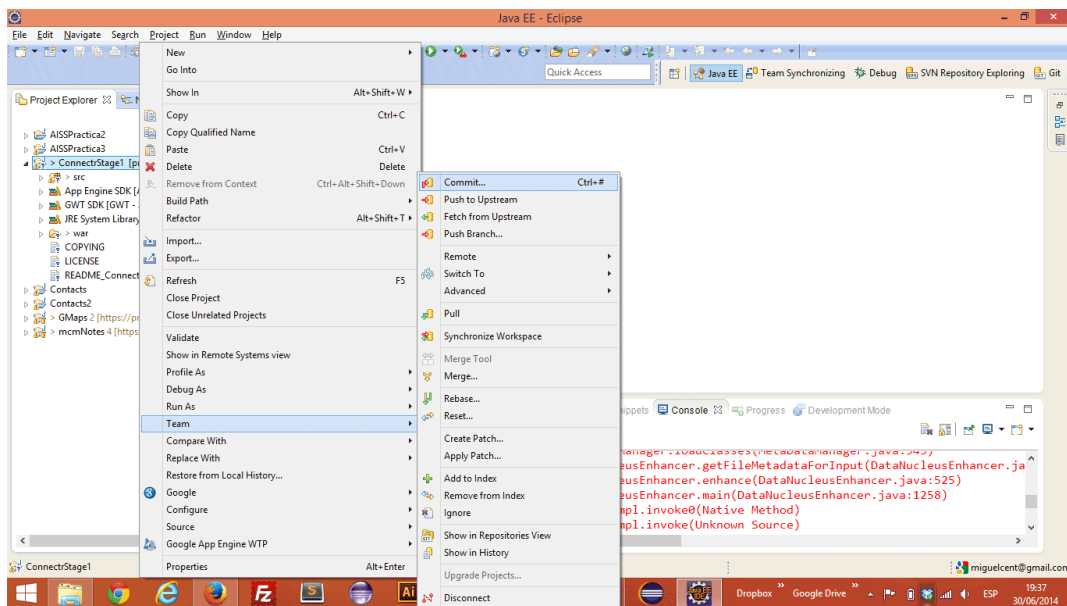


También, podemos usar la app GitHub for Windows y hacemos git clone para crear el repositorio local en nuestro PC (asociado, mediante el índice origin, al repositorio remoto creado anteriormente)

La aplicación GitHub for Windows nos permite realizar todas las acciones necesarias sobre el código y su control de versiones usando Git, pero en este apartado nos centraremos en el uso de Egit, plugin para Eclipse de Git que viene incorporado en la instalación base del mismo.

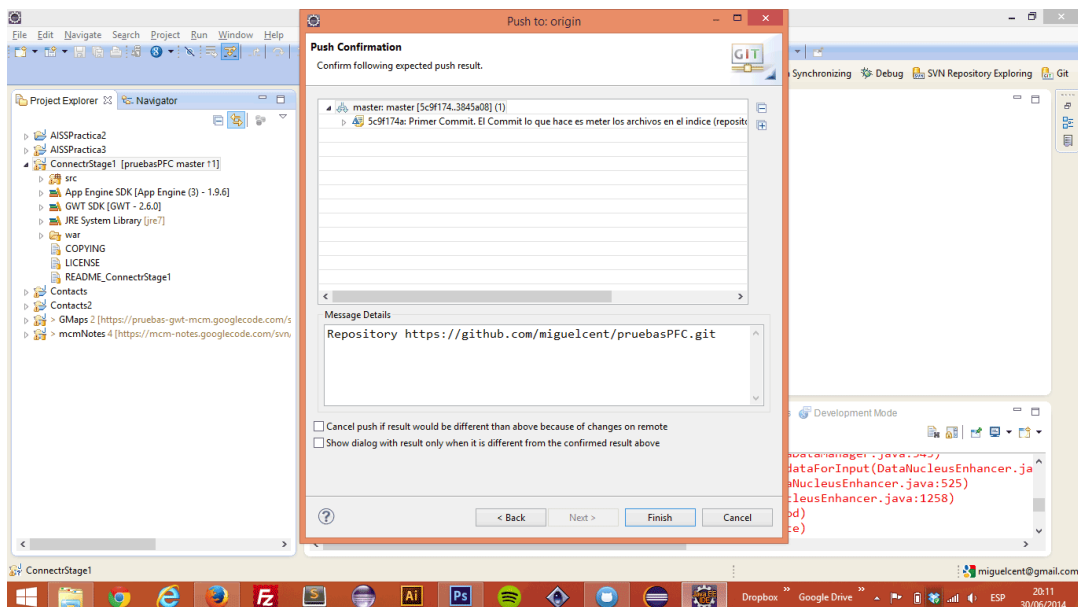
En Eclipse, podemos abrir la vista Git, donde añadiremos el repositorio local a través de las opciones que la vista ofrece.

Una vez que indicamos a eclipse donde se halla nuestro repositorio local, podemos añadir, haciendo git commit, cualquiera de los proyectos que tengamos en Eclipse al repositorio local.



Hasta aquí, tendríamos nuestro repositorio local, que apunta al remoto creado en GitHub, y nuestro proyecto en él tras nuestro primer commit.

A continuación, habría que realizar git push, para que todo lo que se encuentra en el índice origen (los ficheros añadidos, eliminados o actualizados se referencian en el índice tras hacer commit) se actualicen en el repositorio remoto.



A partir de aquí cada vez que hagamos algún cambio que queramos hacer definitivo, primero lo agregamos al repositorio local, haciendo que el índice origen se actualice, mediante git commit y luego, con git push, los hacemos visibles en el repositorio remoto.

Parte III.- Sistema desarrollado

4.- Dominio del problema

4.1.- Descripción del problema

A menudo cuando salimos de viaje a un lugar nuevo resulta difícil encontrar información sobre las alternativas que ofrece para realizar nuestras actividades favoritas.

Si bien la información relacionada con documentos y eventos culturales es amplia, no ocurre lo mismo con la información relacionada con otro tipo de actividades como aquellas relacionadas con la gastronomía, deportes, bares con encanto, etc.

Así, por ejemplo, imaginemos que se encuentra por trabajo 2 días, jueves y miércoles, en Toledo. Para todos los que hemos estado allí es una de las ciudades con más que ver en toda España. Fue capital, de allí es El Greco, está el museo de la guerra de España, tiene restos de 3 culturas que vivieron (incluso convivieron) allí: cristianos, árabes y judíos.

Perfecto hasta aquí, pero tenemos 3 horas el miércoles y 3 el jueves, nos gusta el turismo, la gastronomía y, además hacemos triatlones y, por lo tanto, me he traído la bici.

MyPaths busca cubrir este tipo de necesidades: rutas turísticas, gastronómicas, deportivas... representadas sobre el mapa y con la precisión y facilidad que ofrecen los servicios de Google.

--Antiguo--

En la actualidad, se pueden encontrar muchas aplicaciones orientadas a cubrir las necesidades que tienen las personas de cubrir su, en muchos casos, escaso tiempo libre. Dichas aplicaciones suelen ser específicas de un tipo de actividad, como puede ser la gastronomía, el deporte o el turismo de una localidad, con lo cual buscan solucionar necesidades puntuales para una actividad o un ámbito de necesidades concreto.

Son herramientas pensadas para llegar al mayor número de usuarios posibles y, sobre todo, son orientadas con desde el punto de vista del comercio o local, más que en la creación de comunidad de usuarios. Bien es verdad que la mayoría de las mismas permiten el uso de herramientas sociales pero no el crear comunidad como tal.

También cada vez es más frecuente tener conocido por las personas las actividades que hacer en su lugar habitual de residencia, pero actualmente muchos trabajamos o vivimos temporalmente en sitios que no nos son familiares.

Con *MyPaths* se busca solucionar la necesidad de las personas de realizar sus actividades preferidas --o sencillamente ofrecer algo que hacer en su tiempo libre-- allá donde se encuentre. Actividades que son compartidas por los mismos usuarios de la aplicación y, de todos es sabido, que las recomendaciones y sugerencias de las personas que han realizado ya alguna actividad, han comido en algunos restaurantes o simplemente han paseado por una ciudad, es la mejor fuente de conocimiento y garantía de confianza.

4.2.- Descripción del sistema actual

El principal objetivo es no tener que visitar un elevado alto de sitios web, donde en dos webs diferentes puedes encontrar opiniones o sugerencias contrapuestas para un mismo lugar, una misma actividad.

Nace así la idea de desarrollar una herramienta web en la cual exista una variedad de categorías de actividades a realizar y, además, puedan consultarse en el mapa en forma de ruta, para poder hacer más accesible, y en definitiva más útil, la información a los usuarios.

4.3.- Glosario de términos

pendiente

5.- Visión general del sistema desarrollado

5.1.- Participantes en el proyecto

En este apartado presentamos brevemente a las diferentes personas y organizaciones involucradas en el proyecto.

5.1.1.- Participantes

Participante	Miguel Centeno Muñoz
Organización	Alumnado
Rol	Desarrollador
Es desarrollador	Sí
Es cliente	No
Es usuario	Sí
Comentarios	miguelcent@gmail.com

Participante	Sergio Segura Rueda
Organización	Dpto. Lenguajes y Sistemas Informáticos
Rol	Tutor
Es desarrollador	No
Es cliente	No
Es usuario	Sí
Comentarios	sergiosegura@us.es

5.1.2.- Organizaciones

Organización	Dpto. Lenguajes y Sistemas Informáticos
Dirección	Avenida Reina Mercedes S/N 41012 Sevilla
Teléfono	954 557 139
Fax	954 557 139
Comentarios	E-mail: lsi@lsi.us.es Web: www.lsi.us.es

Organización	Alumnado
Dirección	Luis Montoto, 135 - 2º C
Teléfono	617673924
Fax	
Comentarios	Organización a la que pertenece el alumno desarrollador y el cliente.

5.2.- Objetivos del sistema

OBJ-0001	Gestión Usuarios Aplicación
Versión	1.0 (22/06/2014)
Autores	<ul style="list-style-type: none"> Miguel Centeno Muñoz
Fuentes	<ul style="list-style-type: none"> Sergio Segura Rueda
Descripción	El sistema deberá <i>gestionar toda la información referente a los usuarios de la aplicación y sus perfiles, incluyendo la capacidad de Alta de usuario, así como la modificación del perfil de un usuario.</i>
Subobjetivos	<ul style="list-style-type: none"> [OBJ-0002] Alta de usuario: El sistema deberá <i>gestionar el alta de un nuevo usuario en el sistema</i> [OBJ-0003] Modificación perfil de usuario: El sistema deberá <i>gestionar la modificación del perfil de los usuarios.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

OBJ-0004	Gestión de Rutas
Versión	1.0 (23/06/2014)
Autores	<ul style="list-style-type: none"> Miguel Centeno Muñoz
Fuentes	<ul style="list-style-type: none"> Sergio Segura Rueda
Descripción	El sistema deberá <i>gestionar toda la información referida a las rutas, incluyendo la creación, modificación de las mismas.</i>
Subobjetivos	<ul style="list-style-type: none"> [OBJ-0005] Alta de una Ruta: El sistema deberá <i>gestionar el alta de una nueva ruta creada por un usuario en el sistema.</i> [OBJ-0006] Modificación de una Ruta: El sistema deberá <i>gestionar la modificación de los atributos propios de una ruta.</i> [OBJ-0007] Eliminación de una Ruta: El sistema deberá <i>gestionar la capacidad de un usuario para eliminar una ruta creada por el del sistema.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

6.- Elicitación de requisitos

Se conoce como Elicitación de requisitos a todos los procesos que se realizan anteriormente a la creación del diseño y la implementación de un proyecto software. Un requisito es cada una de las funciones que debe realizar el sistema que se quiere realizar. Normalmente se presenta en forma de lista de requisitos.

Para la recopilación de estos requisitos, existen técnicas variadas, desde las más básicas, aplicables a todo tipo de proyectos, como pueden ser las entrevistas, los talleres, las reuniones en grupo... hasta otras herramientas más orientadas a aspectos tecnológicos, aspectos por los que, además pueden ser categorizados los requisitos. Por ejemplo, habrá requisitos de interfaz, requisitos funcionales, o requisitos de información a almacenar.

Los requisitos y los procesos por los cuáles sean obtenido quedan recogidos habitualmente en un documento de Elicitación de requisitos. Si partimos que el objetivo final del proyecto es, a grandes rasgos, el desarrollo de un aplicación web que presenta rutas en mapas para el desarrollo de actividades diversas, veremos que a partir de este concepto muy generalista nos surgirán requisitos más concretos, pudiéndolos seccionar.

6.1.- Requisitos de información

Las tablas que se presentan a continuación, hacen referencia a requisitos referentes a la información que el sistema debe almacenar.

IRQ-0001 Información sobre el perfil del usuario	
Versión	1.0 (23/06/2014)
Autores	<ul style="list-style-type: none">Miguel Centeno Muñoz
Fuentes	<ul style="list-style-type: none">Sergio Segura Rueda
Dependencias	Ninguno
Descripción	El sistema deberá almacenar la información correspondiente a <i>datos identificativos de cada usuario</i> . En concreto:
Datos específicos	<ul style="list-style-type: none">Nombre de usuarioPasswordEmailNombreApellidosPaísFoto de perfil
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

IRQ-0002	Información sobre las Rutas
Versión	1.0 (23/06/2014)
Autores	<ul style="list-style-type: none"> Miguel Centeno Muñoz
Fuentes	<ul style="list-style-type: none"> Sergio Segura Rueda
Dependencias	Ninguno
Descripción	El sistema deberá almacenar la información correspondiente a <i>datos identificativos e informativos de cada una de las rutas del sistema..</i> En concreto:
Datos específicos	<ul style="list-style-type: none"> Título de la Ruta (<i>El usuario tiene la capacidad de nombrar cada ruta.</i>) Categoría de la Ruta: Mountain Bike, Turística, Gastronómica, Senderismo o Jogging. Descripción de la ruta. (<i>El usuario puede escribir una descripción explicativa sobre cada una de sus rutas</i>) Fichero de la Ruta (<i>la ruta específica al fichero físico, así como el fichero en sí.</i>)
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

6.2.- Reglas de negocio y restricciones

CRQ-0001	Unicidad de los nombres de usuario
Versión	1.0 (23/06/2014)
Autores	<ul style="list-style-type: none"> Miguel Centeno Muñoz
Fuentes	<ul style="list-style-type: none"> Sergio Segura Rueda
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>La información que el sistema almacene como nombr de usuario de un usuario debe ser única.</i>
Importancia	vital
Urgencia	hay presión
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

6.3.- Requisitos funcionales

A continuación se representan los requisitos funcionales que deberá satisfacer la aplicación a desarrollar. Se categorizan en Subsistemas diferenciados por la información con la que tratan.

6.3.1 Diagrama de subsistemas del sistema

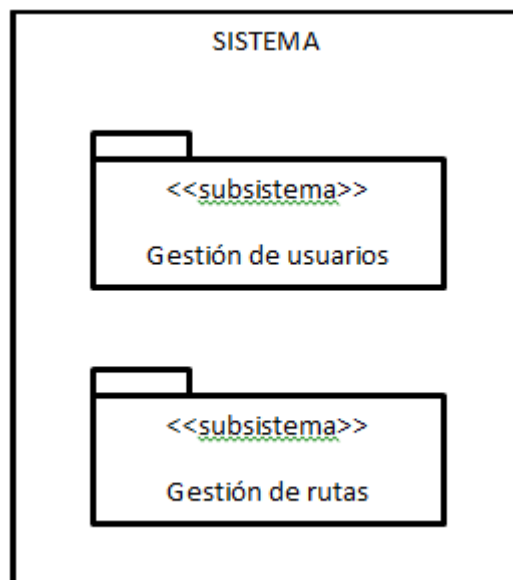


Imagen X: Diagrama subsistemas de la aplicación

6.3.2.- Diagramas de casos de uso

6.3.2.1.- Subsistema de Gestión de Usuarios

En el Subsistema de Gestión de Usuarios se gestiona toda la información referente al perfil de los usuarios de la aplicación, incluyendo modificación de la contraseña y los datos identificativos.

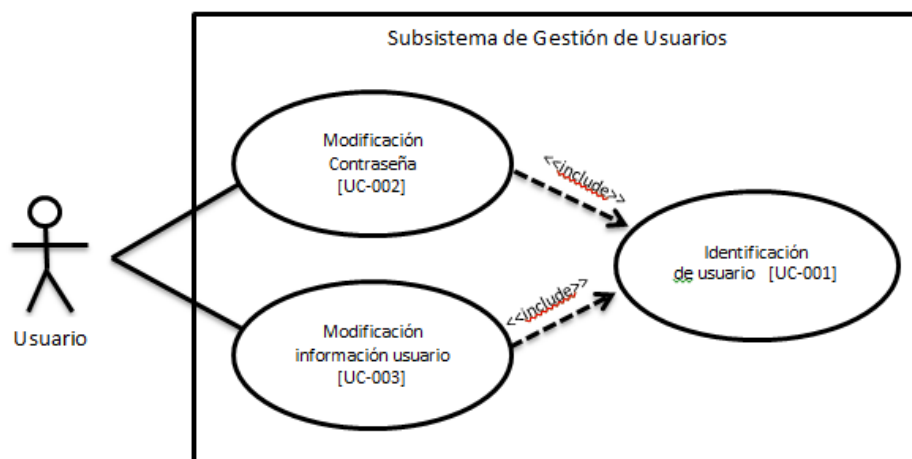


Imagen X: Subsistema de gestión de usuarios

6.3.2.2.- Subsistema de Gestión de Rutas

En el Subsistema de Gestión de Rutas se gestiona toda la información referente a las rutas de la aplicación, incluyendo alta, modificación y eliminación de las mismas.

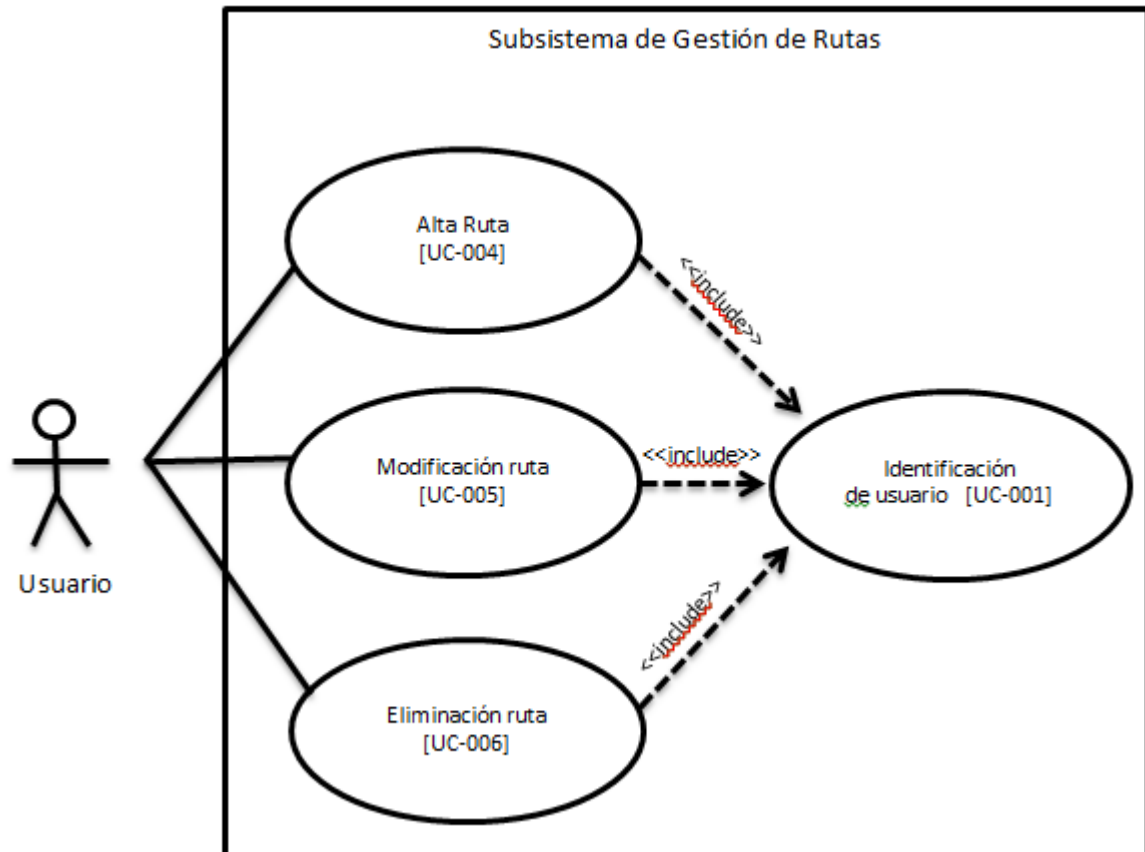


Imagen X: Subsistema de gestión de rutas

6.3.2.- Definición de actores

ACT-0001	Usuario
Versión	1.0 (21/06/2014)
Autores	<ul style="list-style-type: none">Miguel Centeno Muñoz
Fuentes	<ul style="list-style-type: none">Sergio Segura Rueda
Descripción	Este actor representa <i>el usuario registrado en el sistema</i>
Comentarios	Ninguno

6.3.3.- Casos de uso del sistema

6.3.3.1.- Subsistema de gestión de usuarios

UC-0001	Identificación de usuario	
Versión	1.0 (21/06/2014)	
Autores	<ul style="list-style-type: none"> Miguel Centeno Muñoz 	
Fuentes	<ul style="list-style-type: none"> Sergio Segura Rueda 	
Dependencias	<ul style="list-style-type: none"> [OBJ-0001] Gestión Usuarios Aplicación [IRQ-0001] Información sobre Usuario 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso abstracto durante la realización de los siguientes casos de uso: [UC-0002] Modificación contraseña, [UC-0003] Modificación información del usuario, [UC-0004] Alta ruta, [UC-0005] Modificación ruta, [UC-0006] Eliminación ruta	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El sistema <i>solicita los datos de identificación del usuario</i>
	2	El actor Usuario (ACT-0001) <i>introduce los datos de identificación</i>
	3	El sistema <i>valida los datos introducidos por el usuario mostrando un mensaje de confirmación</i>
Postcondición	Usuario reconocido por la aplicación, asignándosele sus respectivos permisos.	
Excepciones	Paso	Acción
	3	Si alguno de los datos aportados por el usuario no es correcto, el sistema muestra un mensaje informando de la no-correcta introducción de los datos de usuario.El sistema permite modificar los datos introducidos erroneamente., a continuación este caso de uso continúa
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	alta	

UC-0002	Modificación contraseña										
Versión	1.0 (21/06/2014)										
Autores	<ul style="list-style-type: none"> Miguel Centeno Muñoz 										
Fuentes	<ul style="list-style-type: none"> Sergio Segura Rueda 										
Dependencias	<ul style="list-style-type: none"> [OBJ-0001] Gestión Usuarios Aplicación [IRQ-0001] Información sobre Usuario 										
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>cuando un usuario solicite cambiar su contraseña.</i>										
Precondición	Ninguna										
Secuencia normal	<table> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>1</td><td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td></tr> <tr> <td>2</td><td>El sistema <i>solicita al usuario una nueva contraseña (por duplicado para evitar confusiones)</i></td></tr> <tr> <td>3</td><td>El actor Usuario (ACT-0001) <i>introduce la nueva contraseña</i></td></tr> <tr> <td>4</td><td>El sistema <i>Registra y asigna nueva contraseña al usuario y muestra confirmación por pantalla</i></td></tr> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema <i>solicita al usuario una nueva contraseña (por duplicado para evitar confusiones)</i>	3	El actor Usuario (ACT-0001) <i>introduce la nueva contraseña</i>	4	El sistema <i>Registra y asigna nueva contraseña al usuario y muestra confirmación por pantalla</i>
Paso	Acción										
1	Se realiza el caso de uso Identificación de usuario (UC-0001)										
2	El sistema <i>solicita al usuario una nueva contraseña (por duplicado para evitar confusiones)</i>										
3	El actor Usuario (ACT-0001) <i>introduce la nueva contraseña</i>										
4	El sistema <i>Registra y asigna nueva contraseña al usuario y muestra confirmación por pantalla</i>										
Postcondición	El sistema asigna una nueva contraseña al usuario										
Importancia	vital										
Urgencia	inmediatamente										
Estado	en construcción										
Estabilidad	alta										

UC-0003	Modificación información del usuario												
Versión	1.0 (25/06/2014)												
Autores	<ul style="list-style-type: none"> Miguel Centeno Muñoz 												
Fuentes	<ul style="list-style-type: none"> Sergio Segura Rueda 												
Dependencias	<ul style="list-style-type: none"> [IRQ-0001] Información sobre Usuario [OBJ-0003] Modificación perfil de usuario 												
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario solicita la modificación/actualización de los datos de su perfil</i>												
Precondición	Ninguno												
Secuencia normal	<table> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>1</td><td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td></tr> <tr> <td>2</td><td>El sistema <i>solicita al usuario los datos del perfil que desea actualizar</i></td></tr> <tr> <td>3</td><td>El actor Usuario (ACT-0001) <i>rellena los datos del perfil que desea actualizar</i></td></tr> <tr> <td>4</td><td>Si <i>dato identificativo a modificar es foto de perfil</i>, el actor Usuario (ACT-0001) <i>provee el fichero con la foto de perfil, buscándolo en su equipo mediante explorador de archivos</i></td></tr> <tr> <td>5</td><td>El sistema <i>registra/actualiza los datos identificativos facilitados mostrando confirmación del proceso por pantalla</i></td></tr> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema <i>solicita al usuario los datos del perfil que desea actualizar</i>	3	El actor Usuario (ACT-0001) <i>rellena los datos del perfil que desea actualizar</i>	4	Si <i>dato identificativo a modificar es foto de perfil</i> , el actor Usuario (ACT-0001) <i>provee el fichero con la foto de perfil, buscándolo en su equipo mediante explorador de archivos</i>	5	El sistema <i>registra/actualiza los datos identificativos facilitados mostrando confirmación del proceso por pantalla</i>
Paso	Acción												
1	Se realiza el caso de uso Identificación de usuario (UC-0001)												
2	El sistema <i>solicita al usuario los datos del perfil que desea actualizar</i>												
3	El actor Usuario (ACT-0001) <i>rellena los datos del perfil que desea actualizar</i>												
4	Si <i>dato identificativo a modificar es foto de perfil</i> , el actor Usuario (ACT-0001) <i>provee el fichero con la foto de perfil, buscándolo en su equipo mediante explorador de archivos</i>												
5	El sistema <i>registra/actualiza los datos identificativos facilitados mostrando confirmación del proceso por pantalla</i>												
Postcondición	Ninguna												
Importancia	vital												
Urgencia	inmediatamente												
Estado	en construcción												
Estabilidad	alta												

6.3.3.2.- Subsistema de gestión de rutas

UC-0004	Alta ruta										
Versión	1.0 (25/06/2014)										
Autores	<ul style="list-style-type: none"> Miguel Centeno Muñoz 										
Fuentes	<ul style="list-style-type: none"> Sergio Segura Rueda 										
Dependencias	<ul style="list-style-type: none"> [OBJ-0004] Gestión de Rutas [OBJ-0005] Alta de una Ruta [IRQ-0002] Información sobre Rutas 										
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario solicita que desea registrar una nueva ruta</i>										
Precondición	Ninguna										
Secuencia normal	<table> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>1</td><td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td></tr> <tr> <td>2</td><td>El sistema <i>solicita al usuario información referente a la ruta que se desea añadir</i></td></tr> <tr> <td>3</td><td>El actor Usuario (ACT-0001) <i>provee la información necesaria para registrar una nueva ruta</i></td></tr> <tr> <td>4</td><td>El sistema <i>registra el alta de la nueva ruta así como muestra un mensaje por pantalla de confirmación</i></td></tr> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema <i>solicita al usuario información referente a la ruta que se desea añadir</i>	3	El actor Usuario (ACT-0001) <i>provee la información necesaria para registrar una nueva ruta</i>	4	El sistema <i>registra el alta de la nueva ruta así como muestra un mensaje por pantalla de confirmación</i>
Paso	Acción										
1	Se realiza el caso de uso Identificación de usuario (UC-0001)										
2	El sistema <i>solicita al usuario información referente a la ruta que se desea añadir</i>										
3	El actor Usuario (ACT-0001) <i>provee la información necesaria para registrar una nueva ruta</i>										
4	El sistema <i>registra el alta de la nueva ruta así como muestra un mensaje por pantalla de confirmación</i>										
Postcondición	Nueva ruta queda registrada en el sistema										
Importancia	vital										
Urgencia	inmediatamente										
Estado	en construcción										
Estabilidad	alta										

UC-0005	Modificación ruta										
Versión	1.0 (25/06/2014)										
Autores	<ul style="list-style-type: none"> Miguel Centeno Muñoz 										
Fuentes	<ul style="list-style-type: none"> Sergio Segura Rueda 										
Dependencias	<ul style="list-style-type: none"> [OBJ-0006] Modificación de una Ruta [OBJ-0004] Gestión de Rutas 										
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario solicita modificar la información referente a una ruta</i>										
Precondición	Ninguna										
Secuencia normal	<table> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>1</td><td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td></tr> <tr> <td>2</td><td>El sistema <i>solicita los atributos de la ruta que el usuario desea modificar</i></td></tr> <tr> <td>3</td><td>Si <i>el atributo de la ruta que desea modificar es fichero ruta</i>, el actor Usuario (ACT-0001) <i>busca el nuevo fichero a registrar en su equipo mediante el explorador de archivos</i></td></tr> <tr> <td>4</td><td>El sistema <i>registra los cambios realizados por el usuario a la ruta mostrando un mensaje de confirmación</i></td></tr> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema <i>solicita los atributos de la ruta que el usuario desea modificar</i>	3	Si <i>el atributo de la ruta que desea modificar es fichero ruta</i> , el actor Usuario (ACT-0001) <i>busca el nuevo fichero a registrar en su equipo mediante el explorador de archivos</i>	4	El sistema <i>registra los cambios realizados por el usuario a la ruta mostrando un mensaje de confirmación</i>
Paso	Acción										
1	Se realiza el caso de uso Identificación de usuario (UC-0001)										
2	El sistema <i>solicita los atributos de la ruta que el usuario desea modificar</i>										
3	Si <i>el atributo de la ruta que desea modificar es fichero ruta</i> , el actor Usuario (ACT-0001) <i>busca el nuevo fichero a registrar en su equipo mediante el explorador de archivos</i>										
4	El sistema <i>registra los cambios realizados por el usuario a la ruta mostrando un mensaje de confirmación</i>										
Postcondición	Ninguna										
Importancia	vital										
Urgencia	inmediatamente										
Estado	en construcción										
Estabilidad	alta										

UC-0006	Eliminación ruta																
Versión	1.0 (25/06/2014)																
Autores	<ul style="list-style-type: none"> Miguel Centeno Muñoz 																
Fuentes	<ul style="list-style-type: none"> Sergio Segura Rueda 																
Dependencias	<ul style="list-style-type: none"> [IRQ-0002] Información sobre Rutas [OBJ-0007] Eliminación de una Ruta [OBJ-0004] Gestión de Rutas 																
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario solicita que desea eliminar una ruta</i>																
Precondición	Ninguna																
Secuencia normal	<table> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>1</td><td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td></tr> <tr> <td>2</td><td>El actor Usuario (ACT-0001) <i>selecciona la ruta que desea eliminar haciendo click en el botón designado para eliminar ruta en la pantalla de gestión de las rutas del usuario</i></td></tr> <tr> <td>3</td><td>El sistema <i>solicita confirmación de que se quiere eliminar la ruta seleccionada</i></td></tr> <tr> <td>4</td><td><i>Si cliente quiere eliminar la ruta , el actor Usuario (ACT-0001) confirma la eliminación de la ruta en cuestión</i></td></tr> <tr> <td>5</td><td><i>Si cliente ha confirmado la eliminación de la ruta, el sistema elimina la ruta en cuestión mostrando un mensaje de confirmación</i></td></tr> <tr> <td>6</td><td><i>Si cliente no quiere eliminar la ruta en cuestión, el actor Usuario (ACT-0001) cancela el proceso de eliminación haciendo uso del botón designado a tal función</i></td></tr> <tr> <td>7</td><td><i>Si cliente cancela eliminación de la ruta, el sistema muestra un mensaje confirmando la no-eliminación de la ruta</i></td></tr> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El actor Usuario (ACT-0001) <i>selecciona la ruta que desea eliminar haciendo click en el botón designado para eliminar ruta en la pantalla de gestión de las rutas del usuario</i>	3	El sistema <i>solicita confirmación de que se quiere eliminar la ruta seleccionada</i>	4	<i>Si cliente quiere eliminar la ruta , el actor Usuario (ACT-0001) confirma la eliminación de la ruta en cuestión</i>	5	<i>Si cliente ha confirmado la eliminación de la ruta, el sistema elimina la ruta en cuestión mostrando un mensaje de confirmación</i>	6	<i>Si cliente no quiere eliminar la ruta en cuestión, el actor Usuario (ACT-0001) cancela el proceso de eliminación haciendo uso del botón designado a tal función</i>	7	<i>Si cliente cancela eliminación de la ruta, el sistema muestra un mensaje confirmando la no-eliminación de la ruta</i>
Paso	Acción																
1	Se realiza el caso de uso Identificación de usuario (UC-0001)																
2	El actor Usuario (ACT-0001) <i>selecciona la ruta que desea eliminar haciendo click en el botón designado para eliminar ruta en la pantalla de gestión de las rutas del usuario</i>																
3	El sistema <i>solicita confirmación de que se quiere eliminar la ruta seleccionada</i>																
4	<i>Si cliente quiere eliminar la ruta , el actor Usuario (ACT-0001) confirma la eliminación de la ruta en cuestión</i>																
5	<i>Si cliente ha confirmado la eliminación de la ruta, el sistema elimina la ruta en cuestión mostrando un mensaje de confirmación</i>																
6	<i>Si cliente no quiere eliminar la ruta en cuestión, el actor Usuario (ACT-0001) cancela el proceso de eliminación haciendo uso del botón designado a tal función</i>																
7	<i>Si cliente cancela eliminación de la ruta, el sistema muestra un mensaje confirmando la no-eliminación de la ruta</i>																
Postcondición	La ruta queda eliminada del sistema																
Importancia	vital																
Urgencia	inmediatamente																
Estado	en construcción																
Estabilidad	alta																

6.3.- Requisitos no funcionales

6.4.- Matriz de rastreabilidad requisitos-objetivos

6.5.- Matriz de rastreabilidad casos de uso-objetivos

7.- Análisis de requisitos

7.1.- Modelo estático del sistema

7.1.1.- Diagramas de tipos

7.1.1.1.- Subsistema de gestión de usuarios

7.1.1.2.- Subsistema de gestión de rutas

7.1.2.- Escenario de prueba

7.1.3.- Tipos

7.1.3.1.- Subsistema de gestión de usuarios

7.1.3.2.- Subsistema de gestión de rutas

7.2.- Modelo dinámico del sistema

7.2.1.- Subsistema de gestión de usuarios

7.2.2.- Subsistema de gestión de rutas

7.3.- Prototipos de interfaz de usuario

A continuación mostraremos las vistas más importantes que conformarán nuestra aplicación. Para la realización de estos diseños hemos usado el software *Moqups* en su versión de prueba. Estos prototipos han sido obtenidos tras varias reuniones con el cliente y a partir de los requisitos obtenidos. Serán utilizados como medio de comunicación con el cliente, como una forma sencilla para él de comprobar que los requisitos que nos está pidiendo aparecen todos en las interfaces mostradas. Por nuestra parte, serán usados para cerciorarnos de que lo que nos está pidiendo se adecua a lo que hemos entendido a partir de los requisitos, y también para comprobar si existen inconsistencias en los requisitos obtenidos o si es necesario añadir o eliminar alguno.

7.3.1.- Pantalla de inicio y búsqueda

texto

A screenshot of a web browser window showing the home page of 'logotipo'. The browser's address bar displays 'http://pathgenerator.com'. In the top right corner, there are links for 'Login' and 'Registro'. The main heading 'logotipo' is centered. Below it is a search bar with the placeholder text 'Busca una ruta', a dropdown menu for 'Tipo de ruta', and a 'Buscar!' button. At the bottom left, there is a banner for Moqups that says 'Created with Moqups (Free Version)' and 'Upgrade your account to remove this banner'.

7.3.2.- Pantalla de login

Texto

A screenshot of a web browser window showing the login page of 'logotipo'. The browser's address bar displays 'http://pathgenerator.com/login'. In the top right corner, there are links for 'Inicio' and 'Registro'. The main heading 'logotipo' is centered. Below it, there are input fields for 'User:' and 'Password:'. Below the password field, there is a link that says '¿Olvidaste tu usuario o contraseña? Click aquí'. At the bottom left, there is a banner for Moqups that says 'Created with Moqups (Free Version)' and 'Upgrade your account to remove this banner'.

7.3.3.- Pantalla de registro de usuario

texto



The screenshot shows a web browser window with the address bar displaying `http://pathgenerator.com/register`. The page has a header with the word "logotipo" on the left and links for "Inicio" and "Login" on the right. The main content area is divided into two sections: "Datos de Usuario" and "Datos Personales".

Datos de Usuario

- Username (*):
- Password(*):
- Repetir Password (*):

Datos Personales

- Email (*):
- Nombre:
- Apellidos:
- Pais:
- Foto:

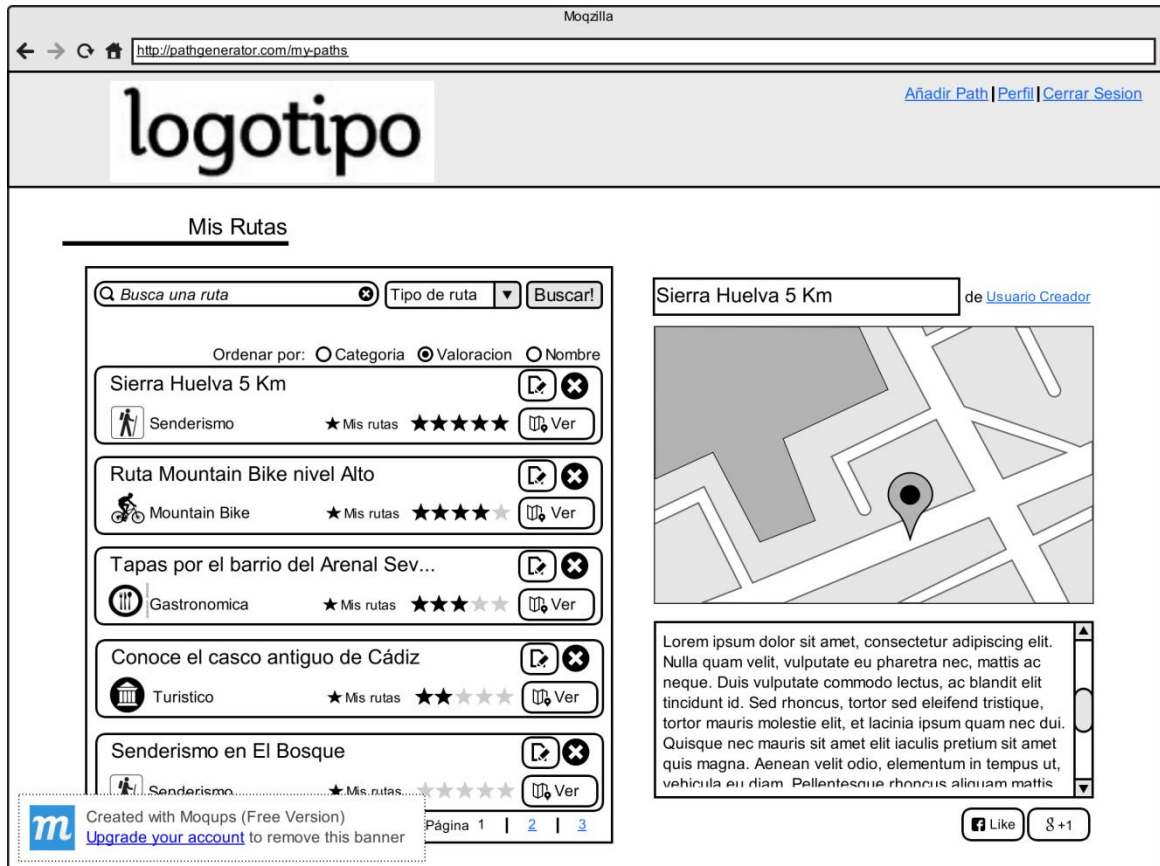
Below the "Datos Personales" section, there is a checkbox labeled "Acepto los" followed by a link "términos de privacidad".

At the bottom right, there is a button labeled "Enviar" with a paper plane icon.

At the bottom left, there is a banner for Moqups: "Created with Moqups (Free Version) Upgrade your account to remove this banner".

7.3.4.- Pantalla Mis Rutas

texto



7.3.5.- Pantalla Añadir Ruta

Texto

Mozilla

← → ↻ 🏠 <http://pathgenerator.com/user/add-path>

logotipo [Añadir Path](#) | [Perfil](#) | [Cerrar Sesión](#)

[Perfil](#)
[My Paths](#)
[Buscar](#)
[Añadir Path](#)

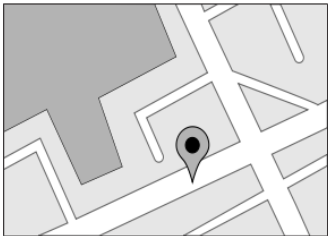
Añadir Path

Título (*)

Categoría(*)

Descripción (*)

Fichero (*)



Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner

7.3.6.- Pantalla Editar Perfil

Texto

Mozilla

← → ↻ 🏠 <http://pathgenerator.com/user/edit-profile>

logotipo [Añadir Path](#) | [Perfil](#) | [Cerrar Sesión](#)

[Perfil](#)
[My Paths](#)
[Buscar](#)
[Añadir Path](#)

Datos de Usuario

Username

Password

Nuevo Password


Datos Personales

Email


Nombre


Apellidos


País



Foto

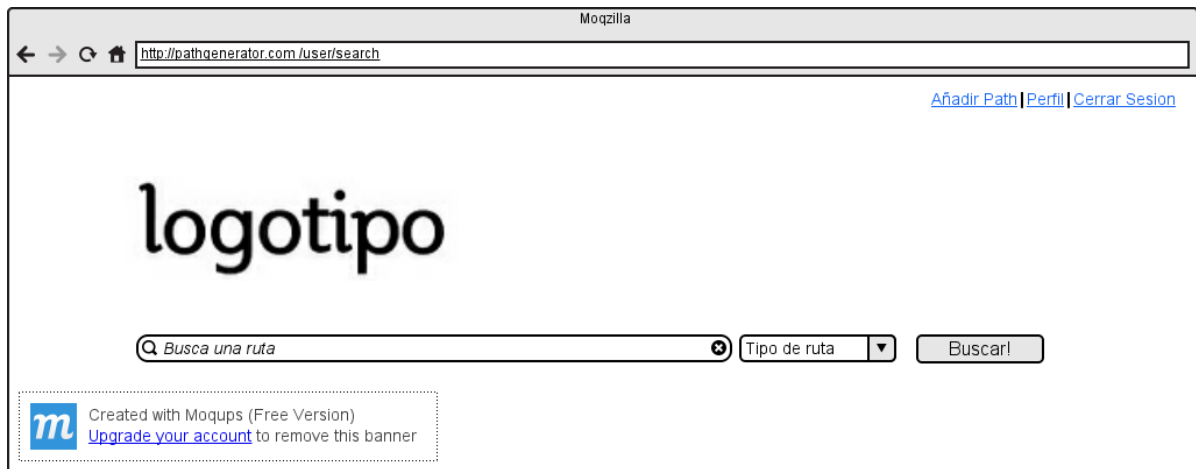
 [Cambiar...](#)

 **Terminar**

 Created with Moqups (Free Version)
[Upgrade your account](#) to remove this banner

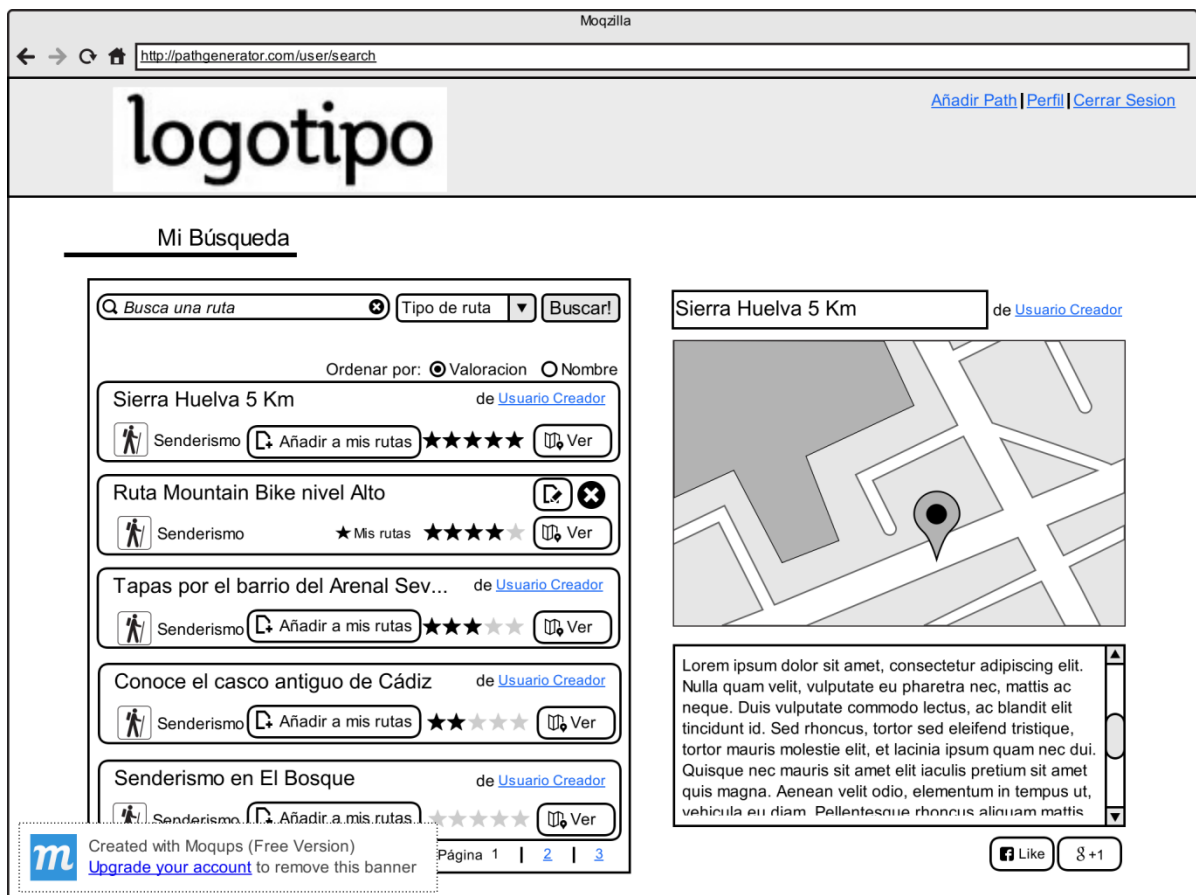
7.3.7.- Pantalla Búsqueda (Usuario logado)

Texto



7.3.8.- Pantalla Resultados Búsqueda (Usuario logado)

Texto



7.3.9.- Pantalla Resultados Búsqueda (Usuario NO logado)

Texto

