



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA TÉCNICA INFORMÁTICA DE GESTIÓN

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

Realizado por
MIGUEL CENTENO MUÑOZ
28633254W

Dirigido por
SERGIO SEGURA RUEDA

Departamento
LENGUAJES Y SISTEMAS INFORMÁTICOS

Sevilla, Septiembre de 2014

Resumen

Al final (hacerlo más motivador)

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
1.4 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 (faltan referencias)	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

3.7.- Projetsii	26
Parte III.- Sistema desarrollado	28
4.- Dominio del problema	30
4.1.- Descripción del problema	30
4.2.- Descripción del sistema actual.....	30
4.3.- Glosario de términos.....	31
5.- Visión general del sistema desarrollado.....	31
5.1.- Participantes en el proyecto	31
5.1.1.- Participantes	31
5.1.2.- Organizaciones.....	31
5.2.- Objetivos del sistema	32
6.- Elicitación de requisitos	34
6.1.- Requisitos de información	34
6.2.- Reglas de negocio y restricciones	37
6.3.- Requisitos funcionales	39
6.3.1 Diagrama de subsistemas del sistema	39
6.3.2.- Diagramas de casos de uso	40
6.3.2.- Definición de actores	42
6.3.3.- Casos de uso del sistema	42
6.3.- Requisitos no funcionales	51
6.4.- Matriz de rastreabilidad requisitos-objetivos.....	52
6.5.- Matriz de rastreabilidad casos de uso-objetivos	52
7.- Análisis de requisitos.....	53
7.1.- Modelo estático del sistema	53
7.1.1.- Diagramas de tipos.....	54
7.1.2.- Tipos.....	55
7.2.- Modelo dinámico del sistema	60
7.2.1.- Subsistema de gestión de usuarios.....	60
.....	61
.....	62
7.2.2.- Subsistema de gestión de rutas	63

.....	64
.....	65
7.3.- Prototipos de interfaz de usuario	67
7.3.1.- Pantalla de inicio y búsqueda	67
7.3.2.- Pantalla de login.....	67
7.3.3.- Pantalla de registro de usuario	68
7.3.4.- Pantalla Mis Rutas.....	69
7.3.5.- Pantalla Añadir Ruta	70
7.3.6.- Pantalla Editar Perfil.....	71
7.3.7.- Pantalla Búsqueda (Usuario logado).....	72
7.3.8.- Pantalla Resultados Búsqueda (Usuario logado)	72
7.3.9.- Pantalla Resultados Búsqueda (Usuario NO logado)	73
8.- Diseño del sistema	74
8.1.- Arquitectura del sistema.....	74
8.2.- Grafo de Navegabilidad.....	76
8.3.- Diseño detallado	77
8.3.1.- Capa de presentación	77
8.3.2.- Capa de Lógica	79

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, el cual se encontraba 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, se llevó 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. *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.

1.4 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 (faltan referencias)

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 Google Plugin for Eclipse. Con él en sencillos pasos podemos dar comienzo al desarrollo de un proyecto GWT con Eclipse.

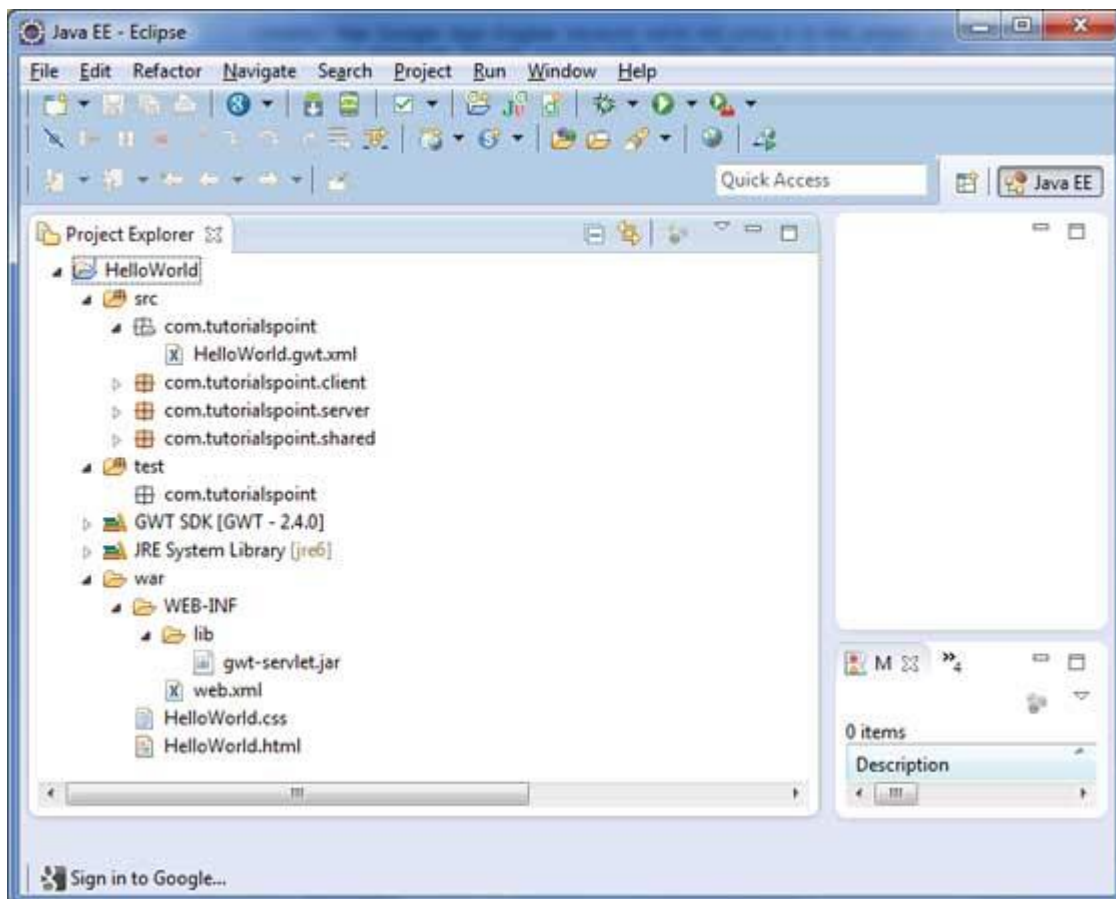


Imagen 1: Estructura de un proyecto GWT en Eclipse.

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. AJAX debe comprenderse como una técnica de desarrollo, no un lenguaje.

Otro motivo es que puede usarse JavaScript nativo para mejorar el trabajo con AJAX.

Funcionalmente, el uso de AJAX recorta en gran medida el uso de peticiones HTTP así como el tráfico de datos cuando se compara con aplicaciones sólo HTML.

Un posible pero es que la implementación de JavaScript por parte de los navegadores no es consistente, aunque mejora mucho en estos últimos años.

¿Cuáles serían los hitos de 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 Fácil pérdida de memoria de apps JavaScript	GWT usa Java para desarrollar apps AJAX GWT genera JavaScript optimizado que limita o elimina este problema
Difícil usar marcadores en apps JavaScript Apps AJAX difíciles de depurar	GWT ofrece soporte completo de histórico de navegación, incluidos marcadores. 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/Usa 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)

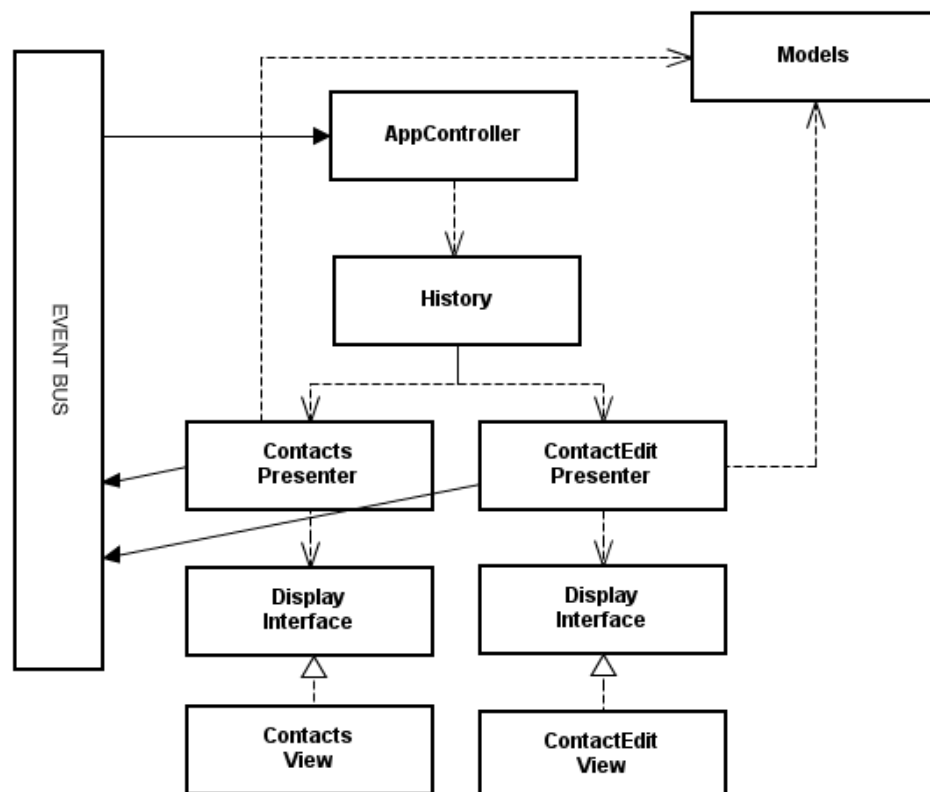


Imagen 2: Diagrama patrón MVP en Gwt en App de prueba de gestión de contactos.

3.5.- Google AppEngine

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

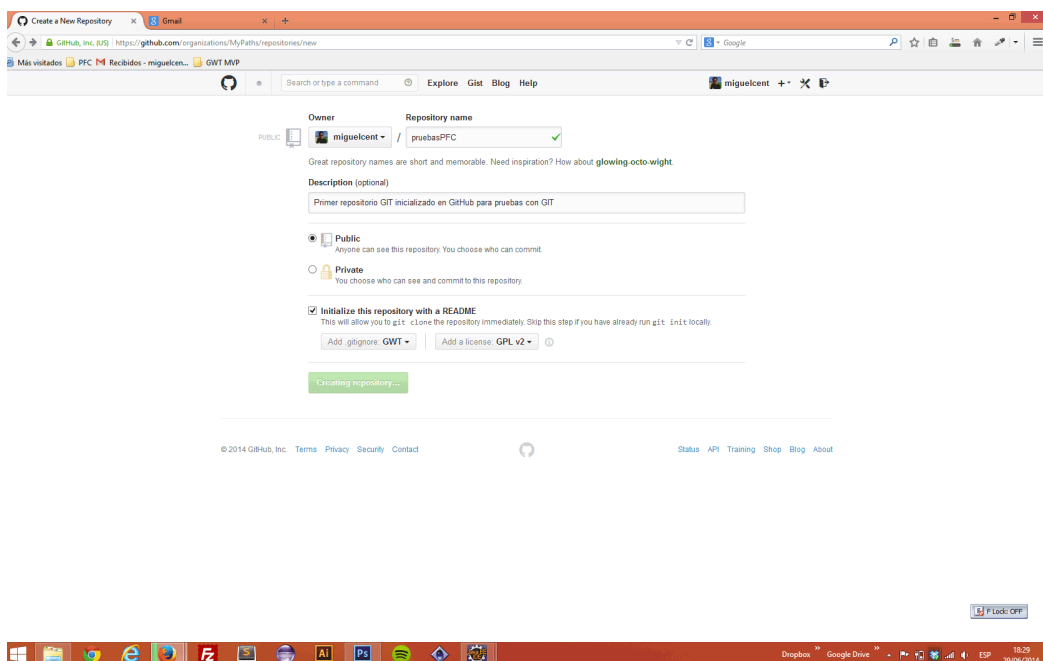


GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails.

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.

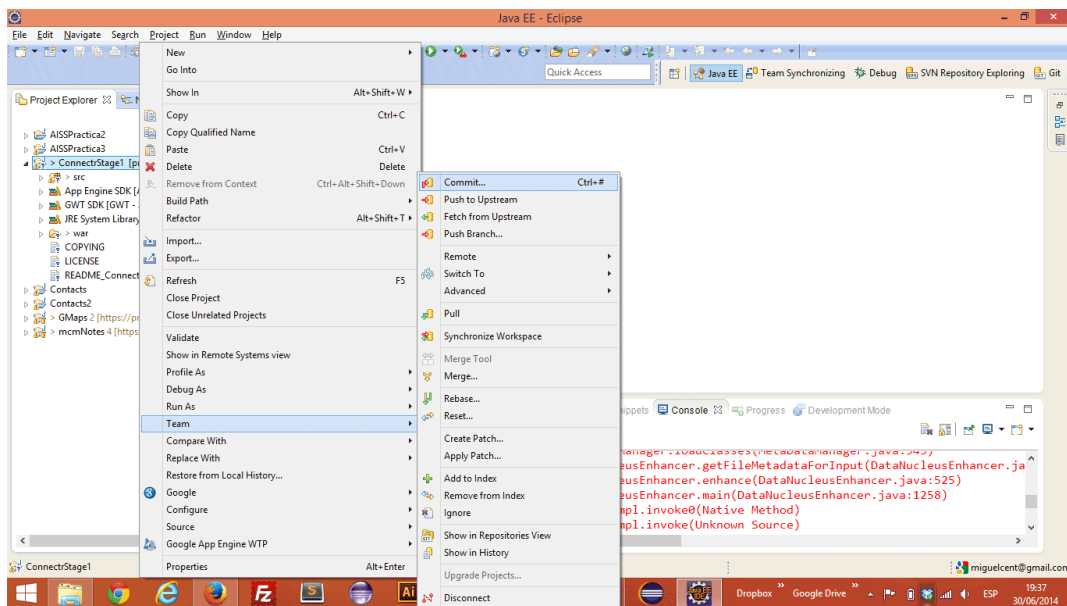


También, podemos usar la app GitHub for Windows y hacemos git clone para crear el repositorio local en nuestro PC (asociado, mediante el índice origin, al repositorio remoto creado anteriormente)

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.

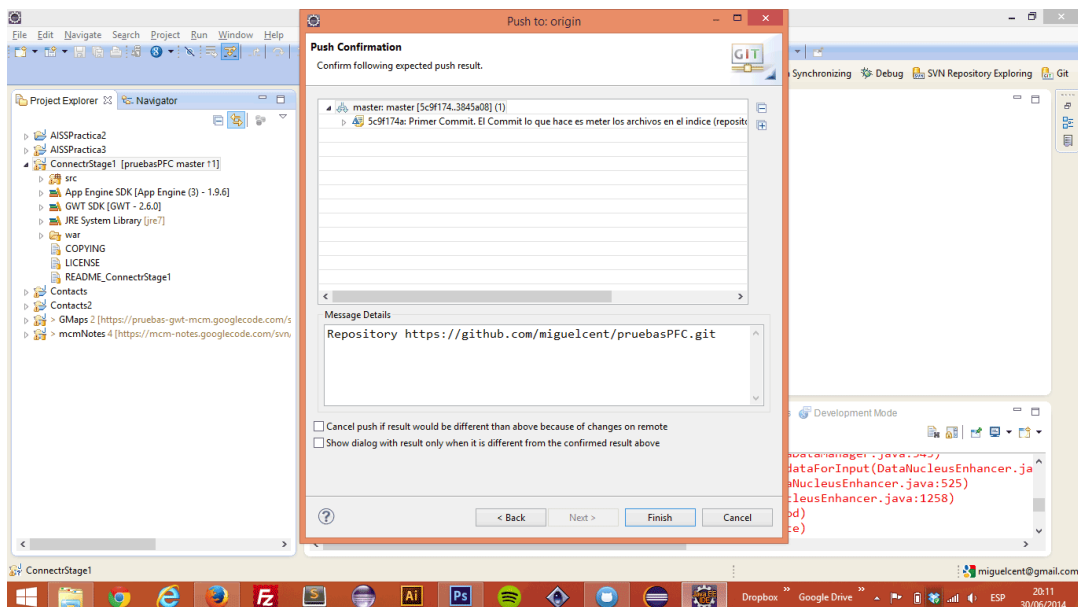
En Eclipse, podemos abrir la vista Git, donde añadiremos el repositorio local a través de las opciones que la vista ofrece.

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.

3.7.- Projetsii

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.

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.

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.

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. (Añadir comparativa)

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 de Usuarios

Versión	1.0 (22/06/2014)
Autores	Miguel Centeno Muñoz
Fuentes	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 identificarse en la aplicación, dar de alta un usuario, así como la modificación de la contraseña y el perfil de usuario y la baja de un usuario.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

OBJ-0002 Gestión de Rutas

Versión	1.0 (23/06/2014)
Autores	Miguel Centeno Muñoz
Fuentes	Sergio Segura Rueda
Descripción	El sistema deberá <i>gestionar toda la información referida a las rutas, incluyendo la creación, modificación y eliminación de las mismas.</i>
Subobjetivos	[OBJ-0003] Gestión Valoración Rutas [OBJ-0004] Gestión Comentarios sobre Rutas
Importancia	vital
Urgencia	inmediatamente

Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

OBJ-0003 Gestión Búsqueda Rutas

Versión	1.0 (27/07/2014)
Autores	Miguel Centeno Muñoz
Fuentes	Sergio Segura Rueda
Descripción	El sistema deberá <i>gestionar la información necesaria para poder mostrar los resultados en las búsquedas de rutas realizadas por los usuarios</i>
Subobjetivos	Ninguno
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	Miguel Centeno Muñoz
Fuentes	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	Nombre de usuario Password Email Nombre Apellidos Ciudad Código Postal País Foto 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	Miguel Centeno Muñoz
Fuentes	Sergio Segura Rueda
Dependencias	Ninguno
Descripción	El sistema deberá almacenar la información correspondiente a datos identificativos e informativos de cada una de las rutas del sistema. En concreto:

Datos específicos	Título de la Ruta (El usuario tiene la capacidad de nombrar cada ruta.) Fecha de creación de la ruta. Descripción de la ruta. Fichero de la Ruta (mapa)
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

IRQ-0003 Información sobre las Categorías de Ruta

Versión	1.0 (23/06/2014)
Autores	Miguel Centeno Muñoz
Fuentes	Sergio Segura Rueda
Dependencias	Ninguno
Descripción	El sistema deberá almacenar la información correspondiente a las categorías a las que pertenecen las rutas. En concreto:
Datos específicos	Nombre de la categoría Imagen descriptiva de la categoría de la ruta.
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

IRQ-0004 Información sobre las Valoraciones de Ruta

Versión	1.0 (23/06/2014)
Autores	Miguel Centeno Muñoz
Fuentes	Sergio Segura Rueda
Dependencias	Ninguno
Descripción	El sistema deberá almacenar la información correspondiente a las valoraciones que reciben las rutas. En concreto:
Datos específicos	Número de valoraciones recibidas Suma de valoraciones recibidas
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

IRQ-0005 Información sobre los Comentarios de Ruta

Versión	1.0 (23/06/2014)
Autores	Miguel Centeno Muñoz

Fuentes	Sergio Segura Rueda
Dependencias	Ninguno
Descripción	El sistema deberá almacenar la información correspondiente a los comentarios que reciben las rutas. En concreto:
Datos específicos	Comentario sobre la ruta
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

IRQ-0006 Información sobre los Mapas

Versión	1.0 (23/06/2014)
Autores	Miguel Centeno Muñoz
Fuentes	Sergio Segura Rueda
Dependencias	Ninguno
Descripción	El sistema deberá almacenar la información correspondiente los mapas que conforman cada una de las rutas
Datos específicos	Tipo de Mapa Imagen de mapa
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	Miguel Centeno Muñoz
Fuentes	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 nombre de usuario de un usuario debe ser única.</i>
Importancia	vital
Urgencia	hay presión
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

CRQ-0002 Usuario puede valorar ruta una sola vez

Versión	1.0 (25/07/2014)
Autores	Miguel Centeno Muñoz
Fuentes	Sergio Segura Rueda
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>Un usuario sólo podrá valorar una ruta en una ocasión.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

CRQ-0003 Un usuario puede subir un máximo de diez rutas al día

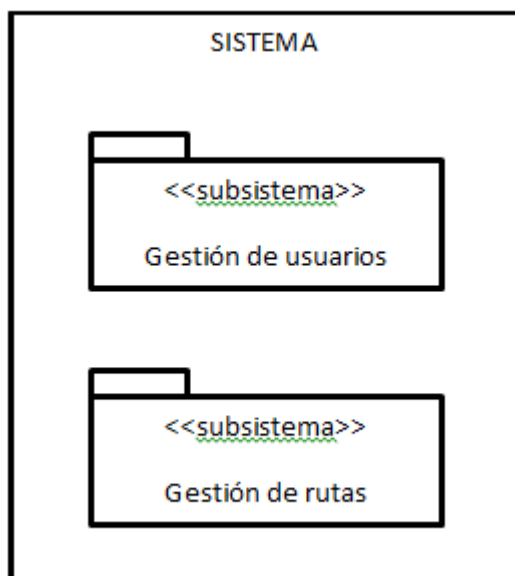
Versión	1.0 (25/07/2014)
Autores	Miguel Centeno Muñoz
Fuentes	Sergio Segura Rueda
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente

Importancia	restricción: <i>un usuario puede subir un máximo de diez rutas por día-</i>
Urgencia	vital
Estado	inmediatamente
Estabilidad	en construcción
Comentarios	alta
	Ninguno

CRQ-0004 Unicidad sobre los nombres de Ruta

Versión	1.0 (25/07/2014)
Autores	Miguel Centeno Muñoz
Fuentes	Sergio Segura Rueda
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>El nombre dado por un usuario a una ruta debe ser único.</i>
Importancia	vital
Urgencia	inmediatamente
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.

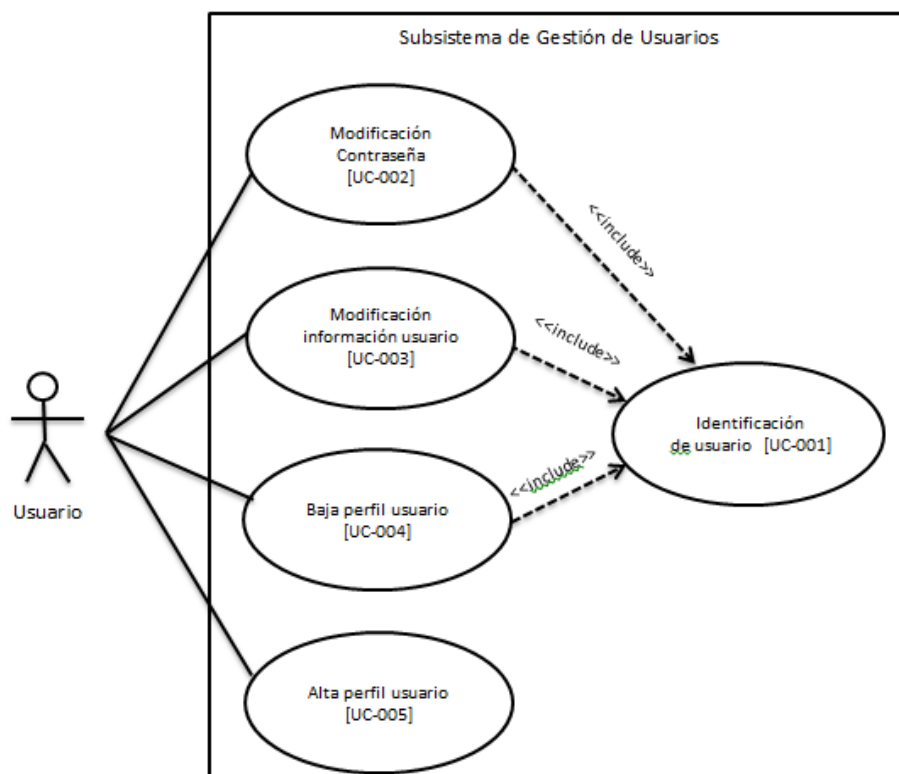


Imagen X: Subsistema de gestión de usuarios

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.

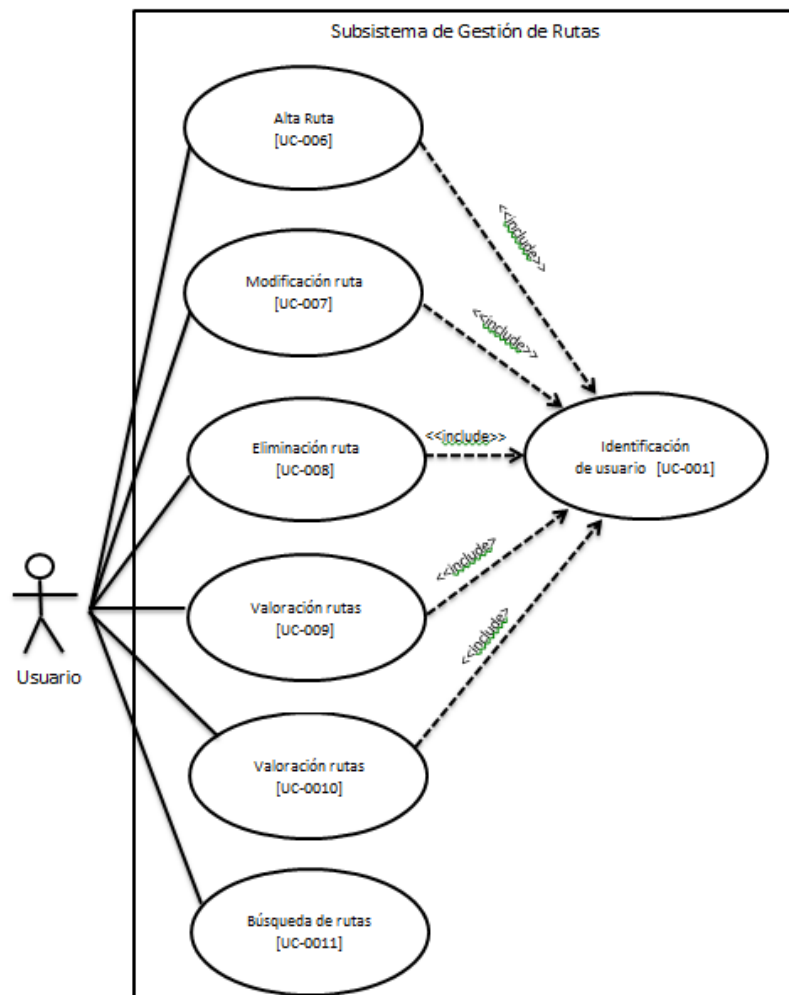


Imagen X: Subsistema de gestión de rutas

6.3.2.- Definición de actores

ACT-0001 Usuario

Versión	1.0 (21/06/2014)
Autores	Miguel Centeno Muñoz
Fuentes	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	Miguel Centeno Muñoz
Fuentes	Sergio Segura Rueda

Dependencias	<ul style="list-style-type: none"> • [OBJ-0001] Gestión Usuarios Aplicación • [IRQ-0001] Información sobre Usuario • [CRQ-0001] Unicidad de los nombres de 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] Baja usuario, [UC-0006] Alta ruta, [UC-0007] Modificación ruta, [UC-0008] Eliminación ruta. [UC-009] Valoración Ruta y [UC-010] Comentario Ruta								
Precondición	Ninguna								
Secuencia normal	<table> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>1</td><td>El sistema <i>solicita los datos de identificación del usuario</i></td></tr> <tr> <td>2</td><td>El actor Usuario (ACT-0001) <i>introduce los datos de identificación</i></td></tr> <tr> <td>3</td><td>El sistema <i>valida los datos introducidos por el usuario mostrando un mensaje de confirmación</i></td></tr> </table>	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>
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	<table> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>3</td><td><i>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 erróneamente., a continuación este caso de uso continúa</i></td></tr> </table>	Paso	Acción	3	<i>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 erróneamente., a continuación este caso de uso continúa</i>				
Paso	Acción								
3	<i>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 erróneamente., a continuación este caso de uso continúa</i>								
Importancia	vital								
Urgencia	inmediatamente								
Estado	en construcción								
Estabilidad	alta								

UC-0002 Modificación contraseña

Versión	1.0 (21/06/2014)
Autores	Miguel Centeno Muñoz
Fuentes	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>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	[IRQ-0001] Información sobre Usuario [OBJ-001] Gestión de Usuarios												
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												
UC-0004	Baja de un usuario												
Versión	1.0 (27/07/2014)												
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 [CRQ-0001] Unicidad de los nombres de usuario 												
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>un usuario solicite la baja de su perfil</i>												
Precondición	Usuario debe estar logado												
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>solicita la baja de su perfil</i></td></tr> <tr> <td>3</td><td>Si <i>Usuario contiene rutas creadas por él</i>, el sistema <i>elimina las rutas que pertenecen al usuario a dar de baja</i></td></tr> <tr> <td>4</td><td>El sistema <i>Elimina el perfil del usuario</i></td></tr> <tr> <td>5</td><td>El sistema <i>Cierra la sesión del usuario</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>solicita la baja de su perfil</i>	3	Si <i>Usuario contiene rutas creadas por él</i> , el sistema <i>elimina las rutas que pertenecen al usuario a dar de baja</i>	4	El sistema <i>Elimina el perfil del usuario</i>	5	El sistema <i>Cierra la sesión del usuario</i>
Paso	Acción												
1	Se realiza el caso de uso Identificación de usuario (UC-0001)												
2	El actor Usuario (ACT-0001) <i>solicita la baja de su perfil</i>												
3	Si <i>Usuario contiene rutas creadas por él</i> , el sistema <i>elimina las rutas que pertenecen al usuario a dar de baja</i>												
4	El sistema <i>Elimina el perfil del usuario</i>												
5	El sistema <i>Cierra la sesión del usuario</i>												
Postcondición	El usuario queda dado de baja del sistema y sus rutas eliminadas												
Importancia	vital												
Urgencia	inmediatamente												
Estado	en construcción												
Estabilidad	alta												
Comentarios	Ninguno												

UC-0005 Alta de nuevo usuario

Versión 1.0 (27/07/2014)

Autores	<ul style="list-style-type: none"> Miguel Centeno Muñoz
Fuentes	<ul style="list-style-type: none"> Sergio Segura Rueda
UC-0007	Modificación ruta
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>un usuario de la aplicación decide registrarse en la misma mediante los mecanismos ofrecidos</i>
Secuencia normal	<p>Paso Acción</p> <ol style="list-style-type: none"> 1 El actor Usuario (ACT-0001) <i>solicita registrarse en la aplicación</i> 2 El sistema <i>solicita la información necesaria</i> 3 El actor Usuario (ACT-0001) <i>provee la información necesaria</i> 4 El sistema <i>registra el usuario en la aplicación</i>
Postcondición	Usuario queda registrado en el sistema
Importancia	vital
Urgencia	inmediatamente
Estado	pendiente de verificación
Estabilidad	alta
Comentarios	Ninguno

6.3.3.2.- Subsistema de gestión de rutas

UC-0006	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"> [IRQ-0002] Información sobre Rutas [IRQ-0003] Información Categoría Ruta [OBJ-0002] 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 registrar una nueva ruta</i>										
Precondición	Ninguna										
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <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> </tbody> </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										

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 [IRQ-0003] Información Categoría Ruta [OBJ-0002] 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>el sistema 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>el sistema 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>el sistema 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-0008	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-0002] 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 clic 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 clic 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 clic 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																

UC-0009	Valoración 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 [IRQ-0004] Información sobre Valoración de Rutas [OBJ-0002] Gestión de Rutas [OBJ-0003] Gestión Valoración de Rutas 								
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario valora una ruta</i> .								
Precondición	Ninguna								
Secuencia normal	<table> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <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 opción de valorar una ruta haciendo clic en el elemento de interfaz destinado a tal función</i></td></tr> <tr> <td>3</td><td>El sistema <i>realiza las operaciones necesarias para la valoración de la ruta</i></td></tr> </tbody> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El actor Usuario (ACT-0001) <i>selecciona la opción de valorar una ruta haciendo clic en el elemento de interfaz destinado a tal función</i>	3	El sistema <i>realiza las operaciones necesarias para la valoración 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 opción de valorar una ruta haciendo clic en el elemento de interfaz destinado a tal función</i>								
3	El sistema <i>realiza las operaciones necesarias para la valoración de la ruta</i>								
Postcondición	Se actualiza y almacena la valoración de la ruta								
Importancia	vital								
Urgencia	inmediatamente								
Estado	en construcción								
Estabilidad	alta								

UC-0010 Comentario 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 [IRQ-0005] Información sobre Comentarios de Rutas [OBJ-0002] Gestión de Rutas [OBJ-0004] Gestión de Comentarios de Rutas 												
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario comenta una ruta</i> .												
Precondición	Ninguna												
Secuencia normal	<table> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <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 opción de comentar una ruta haciendo clic en el elemento de interfaz destinado a tal función</i></td></tr> <tr> <td>3</td><td>El sistema <i>solicita al usuario la introducción del comentario</i></td></tr> <tr> <td>4</td><td>El actor Usuario (ACT-0001) <i>escribe el comentario de la ruta</i></td></tr> <tr> <td>5</td><td>El sistema <i>registra el comentario de la ruta</i></td></tr> </tbody> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El actor Usuario (ACT-0001) <i>selecciona la opción de comentar una ruta haciendo clic en el elemento de interfaz destinado a tal función</i>	3	El sistema <i>solicita al usuario la introducción del comentario</i>	4	El actor Usuario (ACT-0001) <i>escribe el comentario de la ruta</i>	5	El sistema <i>registra el comentario 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 opción de comentar una ruta haciendo clic en el elemento de interfaz destinado a tal función</i>												
3	El sistema <i>solicita al usuario la introducción del comentario</i>												
4	El actor Usuario (ACT-0001) <i>escribe el comentario de la ruta</i>												
5	El sistema <i>registra el comentario de la ruta</i>												
Postcondición	Se actualiza y almacena el comentario de la ruta												
Importancia	vital												
Urgencia	inmediatamente												
Estado	en construcción												
Estabilidad	alta												

UC-0011	Búsqueda de Rutas						
Versión	1.0 (25/07/2014)						
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-0002] Gestión de Rutas [OBJ-0005] Gestión de Búsquedas de Rutas 						
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>un usuario realiza una búsqueda de rutas</i>						
Precondición	El usuario está logado o en la pantalla de inicio						
Secuencia normal	<table> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>1</td><td>El actor Usuario (ACT-0001) <i>rellena el formulario de búsqueda y pulsa el botón "Buscar"</i></td></tr> <tr> <td>2</td><td>El sistema <i>realiza la búsqueda en función a los parámetros introducidos por el usuario</i></td></tr> </table>	Paso	Acción	1	El actor Usuario (ACT-0001) <i>rellena el formulario de búsqueda y pulsa el botón "Buscar"</i>	2	El sistema <i>realiza la búsqueda en función a los parámetros introducidos por el usuario</i>
Paso	Acción						
1	El actor Usuario (ACT-0001) <i>rellena el formulario de búsqueda y pulsa el botón "Buscar"</i>						
2	El sistema <i>realiza la búsqueda en función a los parámetros introducidos por el usuario</i>						
Postcondición							
Importancia	vital						
Urgencia	inmediatamente						
Estado	en construcción						
Estabilidad	baja						
Comentarios	Ninguno						

6.3.- Requisitos no funcionales

NFR-0001	Seguridad
Versión	1.0 (27/07/2014)
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
Descripción	El sistema deberá <i>proveer de mecanismos de seguridad para un acceso seguro y proteger la información de los usuarios</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

NFR-0002	Portabilidad
Versión	1.0 (27/07/2014)
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 <i>podrá ser accedido por cualquiera de los navegadores más populares en la actualidad. Google Chrome, Mozilla Firefox, Internet Explorer, Opera Browser</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

NFR-0003	Escalabilidad
Versión	1.0 (27/07/2014)
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 <i>poseer mecanismos para garantizar su rendimiento en el caso de un acceso de, al menos, 100 usuarios registrados simultáneamente en la aplicación Web</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

6.4.- Matriz de rastreabilidad requisitos-objetivos

TRM-0001	IRQ-0001	IRQ-0002	IRQ-0003	IRQ-0004	IRQ-0005
OBJ-0001	↗	-	-	-	-
OBJ-0002	-	↗	↗	-	-
OBJ-0003	-	-	-	↗	-
OBJ-0004	-	-	-	-	↗
OBJ-0005	-	↗	-	-	-

6.5.- Matriz de rastreabilidad casos de uso-objetivos

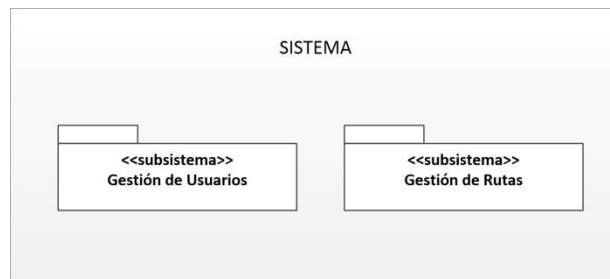
TRM-0002	OBJ-0001	OBJ-0002	OBJ-0003	OBJ-0004	OBJ-0005
UC-0001	↗	-	-	-	-
UC-0002	↗	-	-	-	-
UC-0003	↗	-	-	-	-
UC-0004	↗	-	-	-	-
UC-0005	↗	-	-	-	-
UC-0006	-	↗	-	-	-
UC-0007	-	↗	-	-	-
UC-0008	-	↗	-	-	-
UC-0009	-	-	↗	-	-
UC-0010	-	-	-	↗	-
UC-0011	-	-	-	-	↗

7.- Análisis de requisitos

El análisis de requisitos consiste en la elaboración de un modelo conceptual partiendo del proceso anteriormente realizado, la Elicitación de requisitos. Es uno de los procesos más importantes en el desarrollo de un proyecto software, pues es la manera de ir modelando tecnológicamente algo de lo cual hemos recabado información en el mundo físico. Este modelo es una primera aproximación que será mejorada en sucesivas etapas del desarrollo del proyecto.

7.1.- Modelo estático del sistema

En este apartado se define el modelo estático del sistema. La función del modelo estático del sistema es definir un conjunto de tipos abstractos de datos y asociaciones entre estos que satisfagan los requisitos de información que se detectaron en la fase de Elicitación. Se divide el conjunto de tipos de datos y asociaciones en dos subsistemas cuyos diagramas se muestran a continuación.

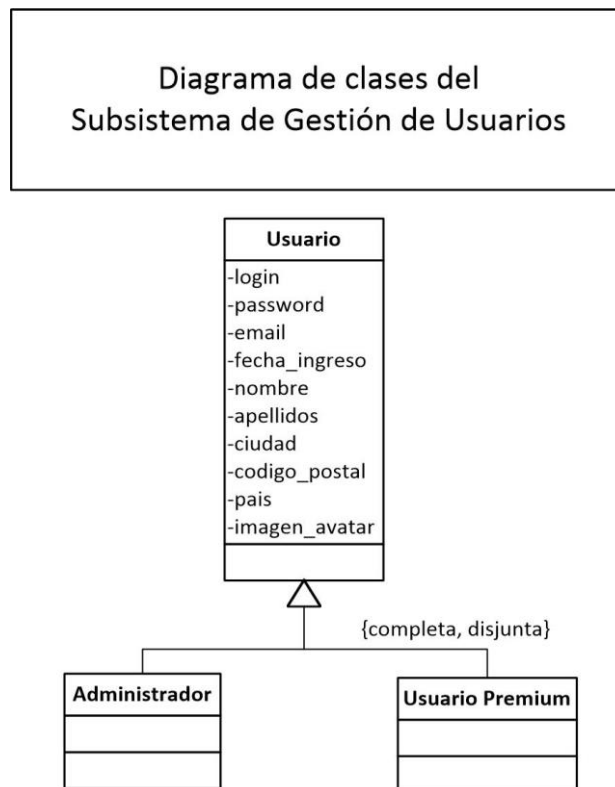


- **Subsistema de Gestión de usuarios:** tipos y asociaciones relacionados con la relación de usuarios, sus atributos y características.
- **Subsistema de Gestión de rutas:** tipos y asociaciones relacionados con la relación de rutas, categorías de estas, su gestión y características.

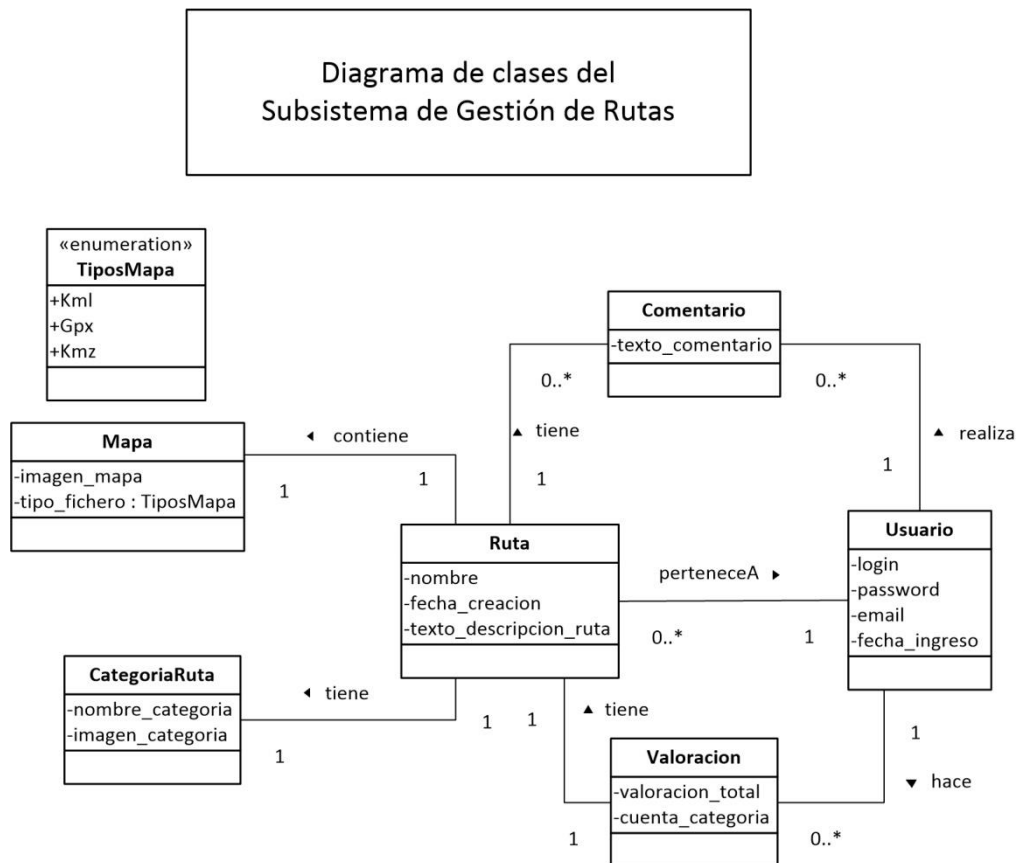
7.1.1.- Diagramas de tipos

Aquí se muestran cada uno de los diagramas de tipos separados por subsistemas.

7.1.1.1.- Subsistema de Gestión de Usuarios



7.1.1.2.- Subsistema de Gestión de Rutas



TYP-0001

Usuario

Versión	1.0(27/07/2014)	
Autores	Miguel Centeno Muñoz	
Dependencias	[OBJ-0001] Gestión de Usuarios	
Descripción	Este tipo de objetos representa <i>a los usuarios registrados en el sistema.</i>	
Componentes	Nombre	Tipo
	Valoracion	Multiplicidad
	rutas	Set<Ruta> 0..n

Atributo variable

Usuario:: login

Descripción	Este atributo representa <i>el nombre con el que el usuario accederá a la aplicación</i>
Tipo	String

Atributo variable **Usuario:: password**

Descripción	Este atributo representa <i>la contraseña del usuario para acceder al sistema</i>
Tipo	String

Atributo Variable **Usuario:: email**

Descripción	Este atributo representa <i>la dirección de correo electrónico del usuario</i>
Tipo	String

Atributo variable **Usuario:: fecha_ingreso**

Descripción	Este atributo representa <i>la fecha en la que el usuario fue registrado en el sistema</i>
Tipo	Date

Atributo variable **Usuario:: nombre**

Descripción	Este atributo representa <i>el nombre real del usuario</i>
Tipo	String

Atributo variable **Usuario:: apellidos**

Descripción	Este atributo representa <i>los apellidos del usuario</i>
Tipo	String

Atributo variable **Usuario:: ciudad**

Descripción	Este atributo representa <i>la ciudad de origen del usuario</i>
Tipo	String

Atributo variable **Usuario:: cod_postal**

Descripción	Este atributo representa <i>el código postal del usuario</i>
Tipo	String

Atributo variable **Usuario:: país**

Descripción	Este atributo representa <i>el país de origen del usuario</i>
Tipo	String

Atributo variable **Usuario:: imagen_avatar**

Descripción	Este atributo representa <i>el nombre del fichero de la imagen del cliente</i>
Tipo	String

Expresión de invariante **Usuario:: Unicidad de los nombres de usuario**

Descripción	No pueden existir dos usuarios con el mismo login
Comentarios	El nombre de usuario es el atributo login

7.1.2.2.- Subsistema de Gestión de Rutas

TYP-0002	Ruta		
Versión	1.0(27/07/2014)		
Autores	Miguel Centeno Muñoz		
Dependencias	[OBJ-0002] Gestión de Rutas		
	[OBJ-0005] Gestión Búsqueda Rutas		
Descripción	Este tipo de objetos representa <i>las rutas registradas en el sistema</i>		
Componentes	Nombre	Tipo	Multiplicidad

mapa	Mapa	1
categoria_ruta	CategoriaRuta	1
comentarios	Set<Comentario>	0..n
valoración	Valoracion	1

Atributo variable Ruta:: nombre_ruta

Descripción	Este atributo representa <i>el nombre con el que la ruta fue registrada</i>
Tipo	String

Atributo variable Ruta:: texto_descripcion

Descripción	Este atributo representa <i>la descripción que el usuario hace de la ruta</i>
Tipo	String

Atributo variable Ruta:: fecha_creacion

Descripción	Este atributo representa <i>la fecha en la que la ruta fue creada</i>
Tipo	Date

Expresión de invariante Usuario:: Unicidad de los nombres de usuario

Descripción	No pueden existir dos rutas con el mismo nombre
-------------	---

TYP-0003 Mapa

Versión	1.0(27/07/2014)		
Autores	Miguel Centeno Muñoz		
Dependencias	[OBJ-0002] Gestión de Rutas		
Descripción	Este tipo de objetos representa <i>el mapa de la ruta</i>		
Componentes	Nombre	Tipo	Multiplicidad
	tipo_fichero	TiposMapa	1

Atributo variable Mapa:: imagen_mapa

Descripción	Este atributo representa <i>el nombre el fichero del mapa</i>
Tipo	String

TYP-0004 CategoriaRuta	
Versión	1.0(27/07/2014)
Autores	Miguel Centeno Muñoz
Dependencias	[OBJ-0002] Gestión de Rutas [OBJ-0005] Gestión búsqueda rutas
Descripción	Este tipo de objetos representa <i>la categoría a la que pertenece la ruta</i>

Atributo variable CategoriaRuta:: nombre_categoria	
Descripción	Este atributo representa <i>el nombre de la categoría de la ruta</i>
Tipo	String

Atributo variable CategoriaRuta:: imagen_categoria	
Descripción	Este atributo representa <i>la imagen representativa de la categoría</i>
Tipo	String

TYP-0005 Valoracion	
Versión	1.0(27/07/2014)
Autores	Miguel Centeno Muñoz
Dependencias	[OBJ-0003] Gestión Valoración Rutas
Descripción	Este tipo de objetos representa <i>la valoración que tiene la ruta</i>

Atributo variable Valoracion:: valoración_total	
Descripción	Este atributo representa <i>la valoración total o suma de las valoraciones que tiene hasta el momento</i>
Tipo	String

Atributo variable Valoracion:: cuenta_valoraciones	
Descripción	Este atributo representa <i>el número de veces que la ruta ha sido valorada</i>
Tipo	String

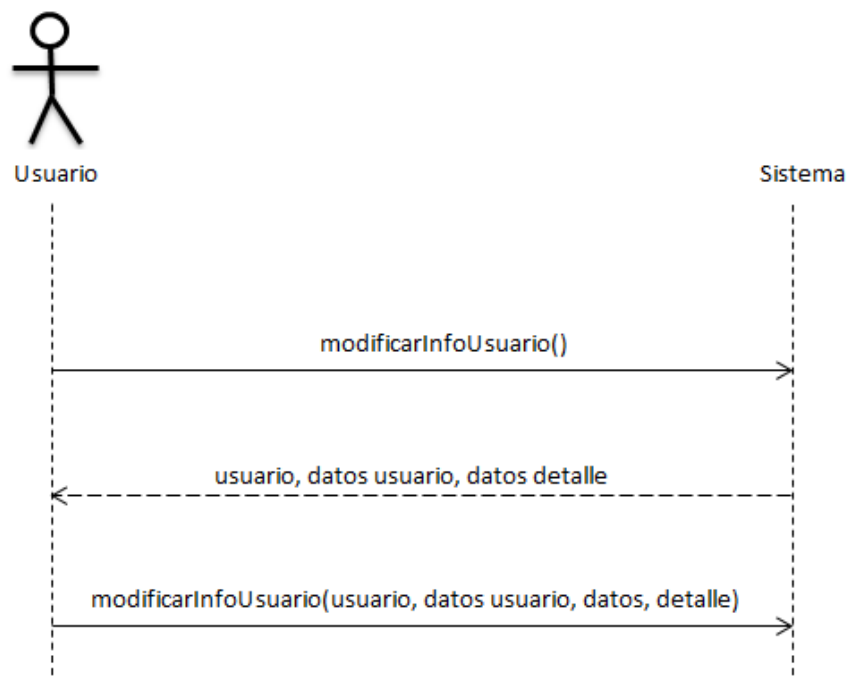
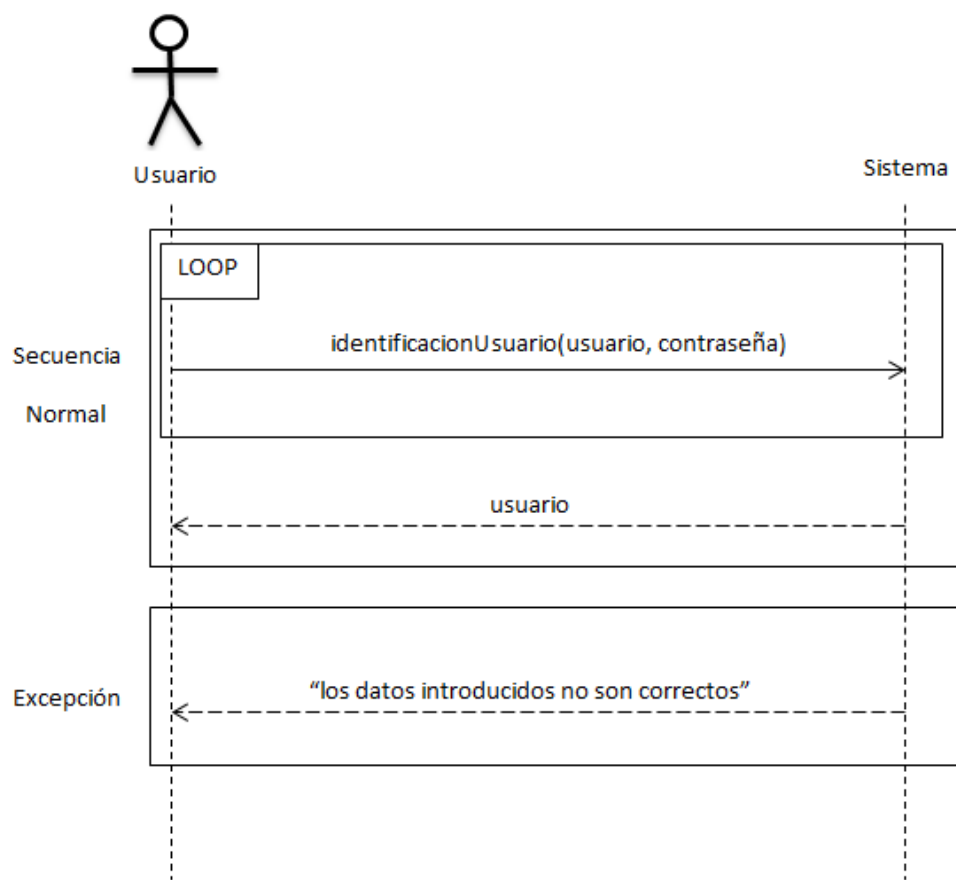
TYP-0006 Comentario	
Versión	1.0(27/07/2014)
Autores	Miguel Centeno Muñoz
Dependencias	[OBJ-0004] Gestión Comentarios Rutas
Descripción	Este tipo de objetos representa <i>los comentarios recibidos por las rutas</i>

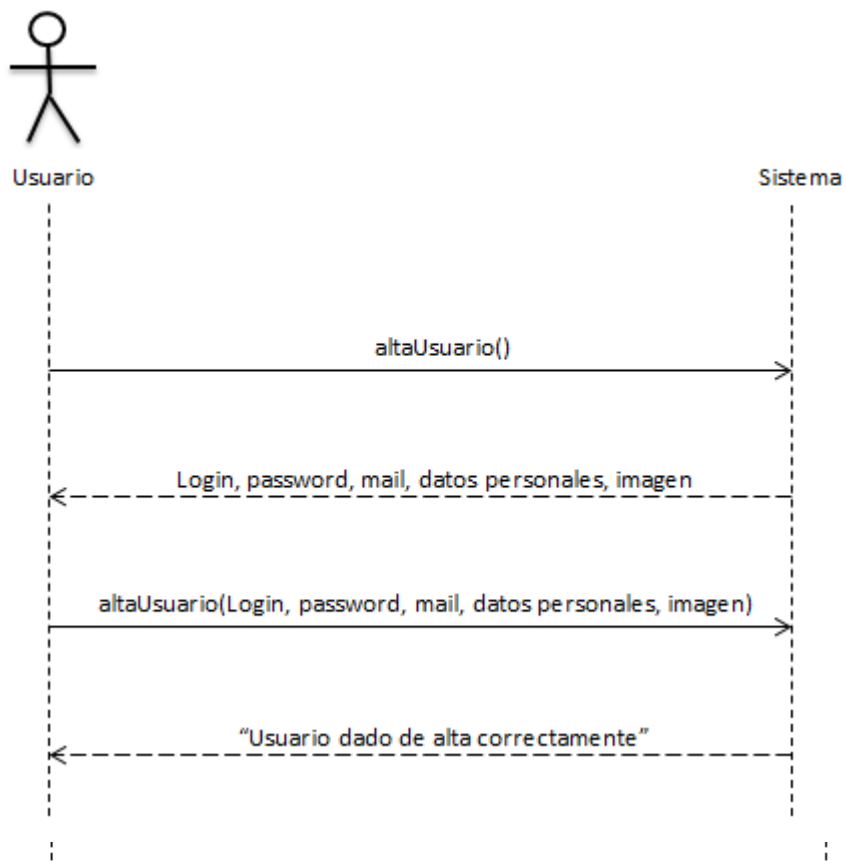
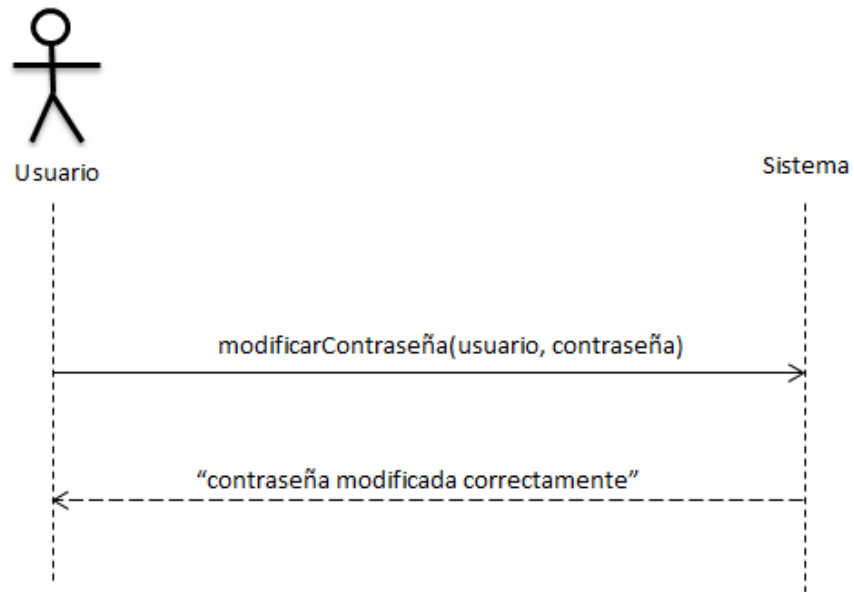
Atributo variable	Comentario:: texto_comentario
Descripción	Este atributo representa <i>el texto del comentario que un usuario hace sobre una ruta</i>
Tipo	String

TYP-0007	TiposMapa
Versión	1.0(27/07/2014)
Autores	Miguel Centeno Muñoz
Dependencias	[OBJ-0002] Gestión de Rutas
Descripción	Este tipo de objetos representa <i>el tipo de fichero del mapa</i>
Comentarios	Enumerado de los tipos de mapa

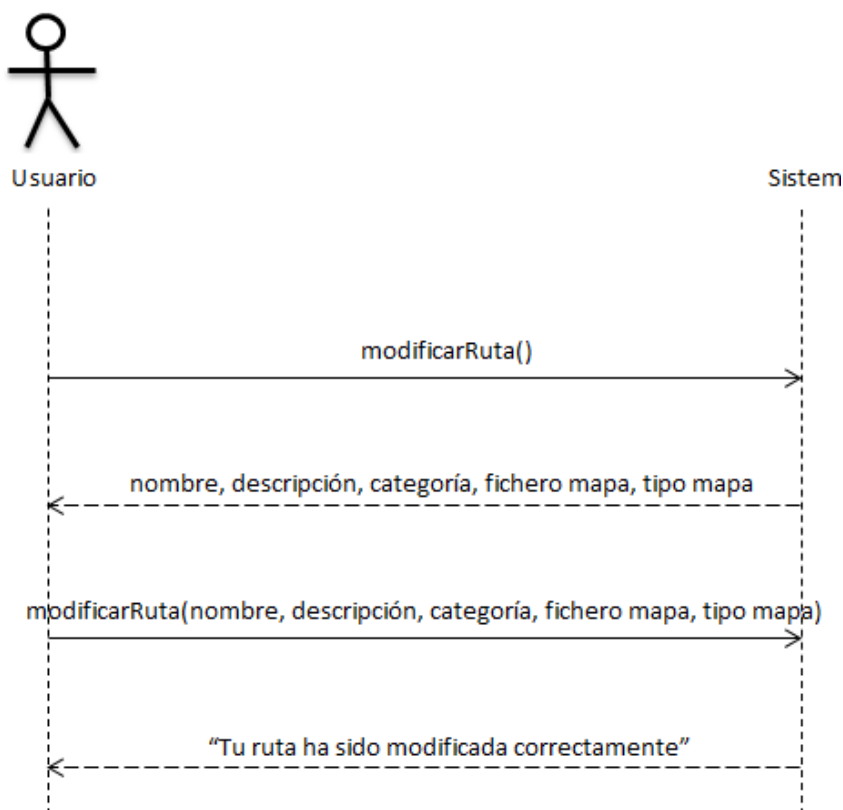
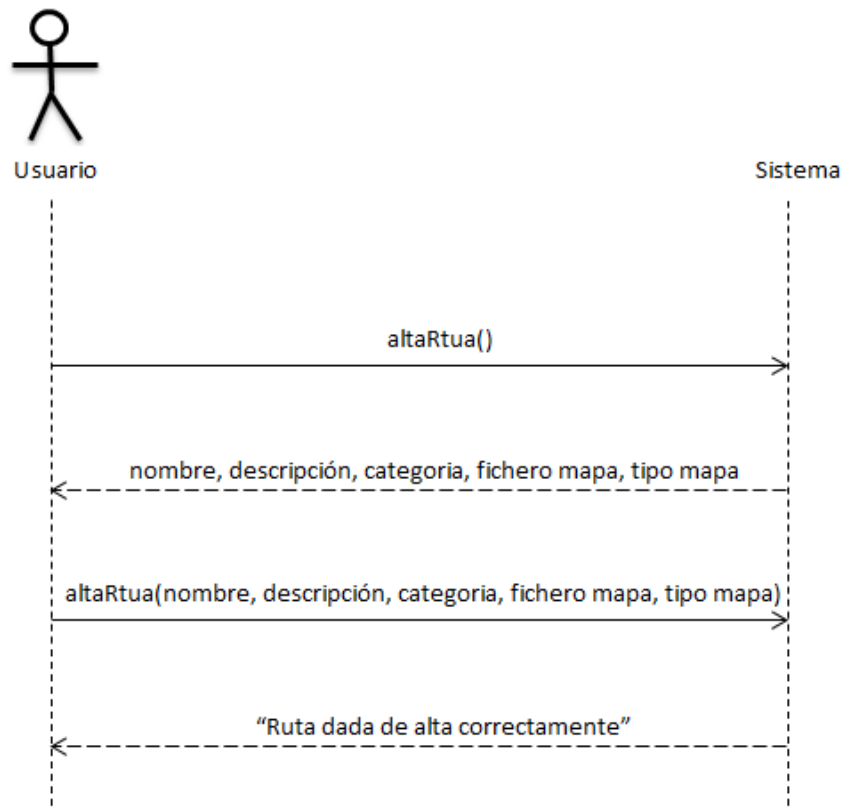
7.2.- Modelo dinámico del sistema

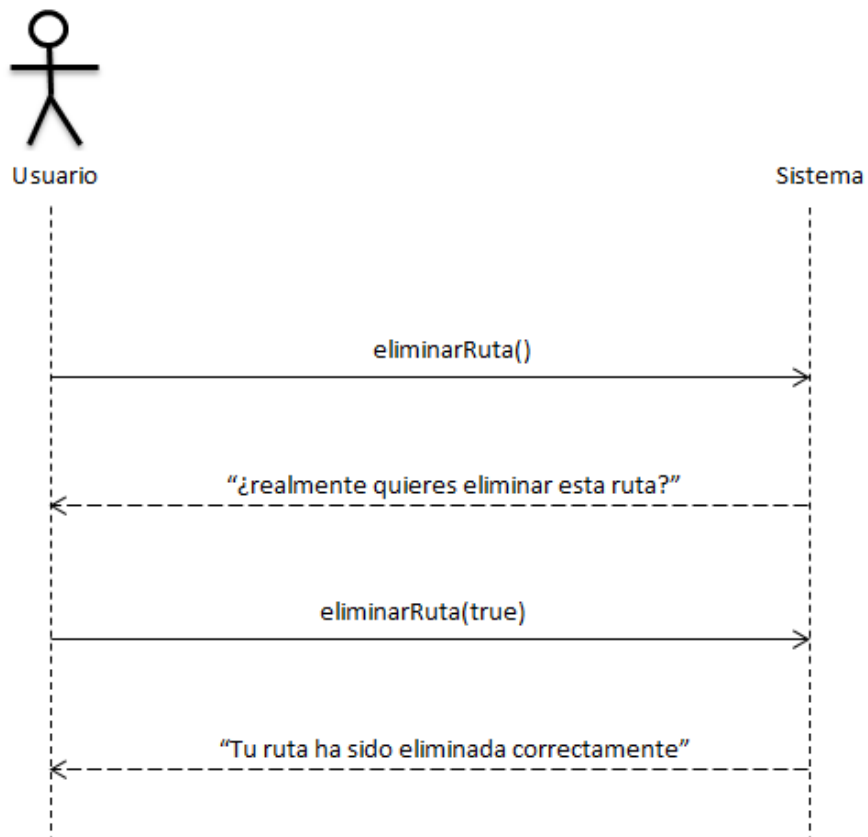
7.2.1.- Subsistema de gestión de usuarios





7.2.2.- Subsistema de gestión de rutas

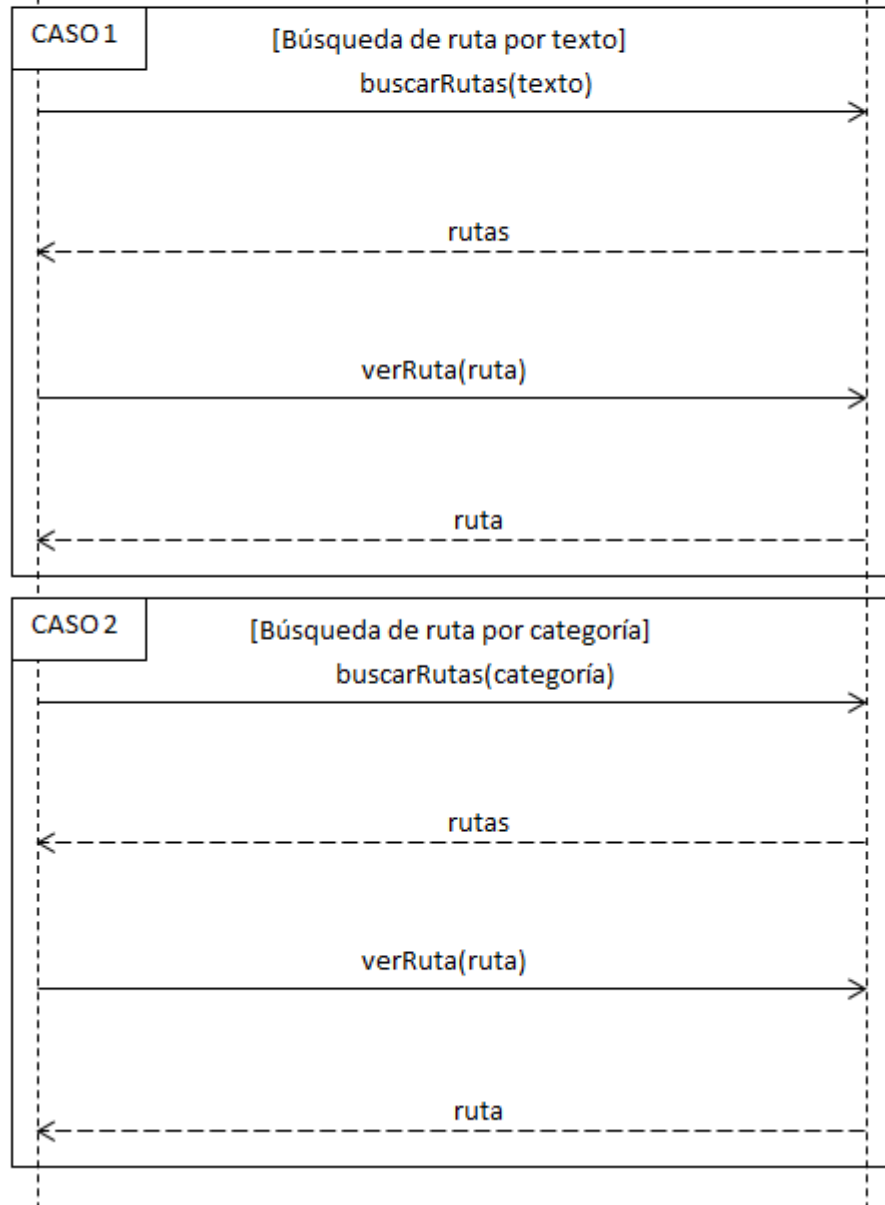


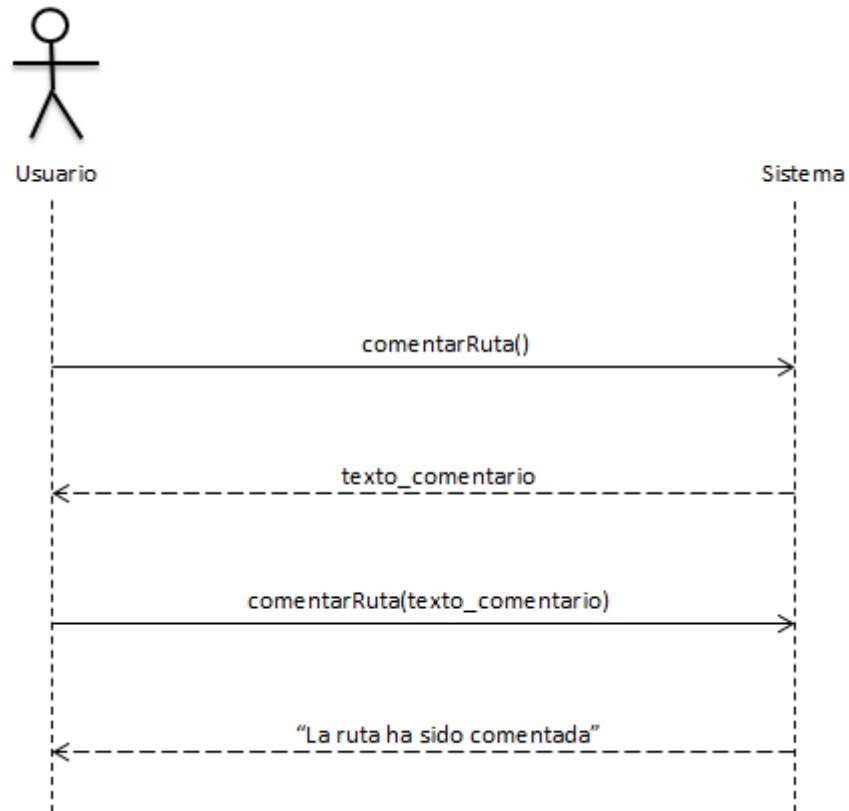




Usuario

Sistema





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 a partir de los requisitos obtenidos. Normalmente, los prototipos de interfaz de usuario, se usan como una primera aproximación de lo que acabará siendo la aplicación y para comunicarnos con un cliente final. Mediante estos prototipos y su presentación a los clientes, estos pueden evaluar que se van cumpliendo los requisitos que desean que la aplicación realice.

7.3.1.- Pantalla de inicio y búsqueda

La primera pantalla con la que nos encontramos tiene un formulario para buscar rutas en función a varios parámetros y, además, un top-menú de acceso rápido al login y el registro de usuarios.

The screenshot shows a web browser window with the address bar displaying 'http://pathgenerator.com'. The page features a large 'logotipo' text in the center. Below it is a search bar with the placeholder text 'Busca una ruta', a dropdown menu labeled 'Tipo de ruta', and a 'Buscar!' button. In the top right corner, there are links for 'Login' and 'Registro'. At the bottom left, there is a banner that reads 'Created with Moqups (Free Version)' and 'Upgrade your account to remove this banner'.

7.3.2.- Pantalla de login

Pantalla a través de la cual, un usuario puede acceder a la aplicación

The screenshot shows a web browser window with the address bar displaying 'http://pathgenerator.com/login'. The page features a large 'logotipo' text in the center. Below it, there are input fields for 'User' and 'Password'. A link for '¿Olvidaste tu usuario o contraseña? Click aquí' is located below the password field. In the top right corner, there are links for 'Inicio' and 'Registro'. At the bottom left, there is a banner that reads 'Created with Moqups (Free Version)' and 'Upgrade your account to remove this banner'.

7.3.3.- Pantalla de registro de usuario

Pantalla a través de la cual, un usuario puede registrarse en la aplicación

Mozilla

← → ↻ 🏠 <http://pathgenerator.com/register>

Inicio | Login

logotipo

Datos de Usuario

Username (*)

Password(*)

Repetir Password (*)

Datos Personales

Email (*)

Nombre

Apellidos

Pais

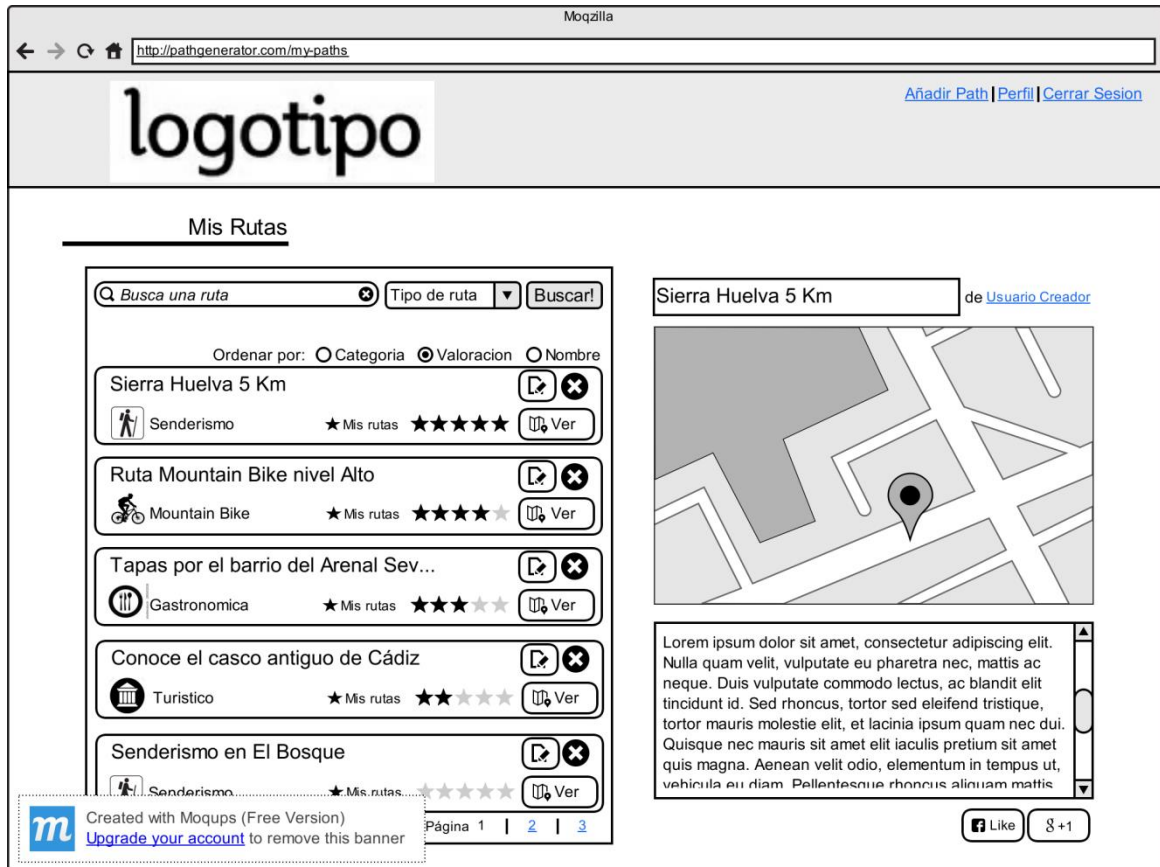
Foto

☒ Acepto los [términos de privacidad](#)

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

7.3.4.- Pantalla Mis Rutas

Pantalla en la que desembarca un usuario una vez realizado el login en la aplicación en ella podemos observar que hay un listado con las rutas que son creadas por el usuario, y un detalle de la ruta seleccionada. Además, se podrá compartir en redes sociales la ruta.



7.3.5.- Pantalla Añadir Ruta

Pantalla para añadir una ruta a la aplicación.

Mozilla

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

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

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

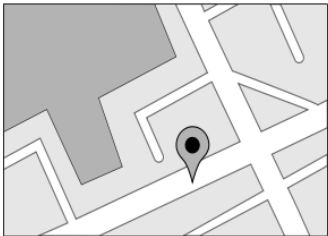
Añadir Path

Título (*)

Categoría(*)

Descripción (*)

Fichero (*)



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

7.3.6.- Pantalla Editar Perfil

Pantalla a través de la cual un usuario puede modificar o actualizar su información de perfil.

Moqzila

← → ↻ 🏠 <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)

Pantalla de búsqueda de rutas. Contiene el formulario de búsquedas, cambiando el top-menú al estar el usuario logado.



7.3.8.- Pantalla Resultados Búsqueda (Usuario logado)

Resultados de la búsqueda para un usuario logado en la aplicación.



7.3.9.- Pantalla Resultados Búsqueda (Usuario NO logado)

Pantalla con los resultados de una búsqueda para un usuario no logado.



8.- Diseño del sistema

En este capítulo se realizará el diseño del sistema a desarrollar obtenido a partir de la elicitación de requisitos y del análisis de estos.

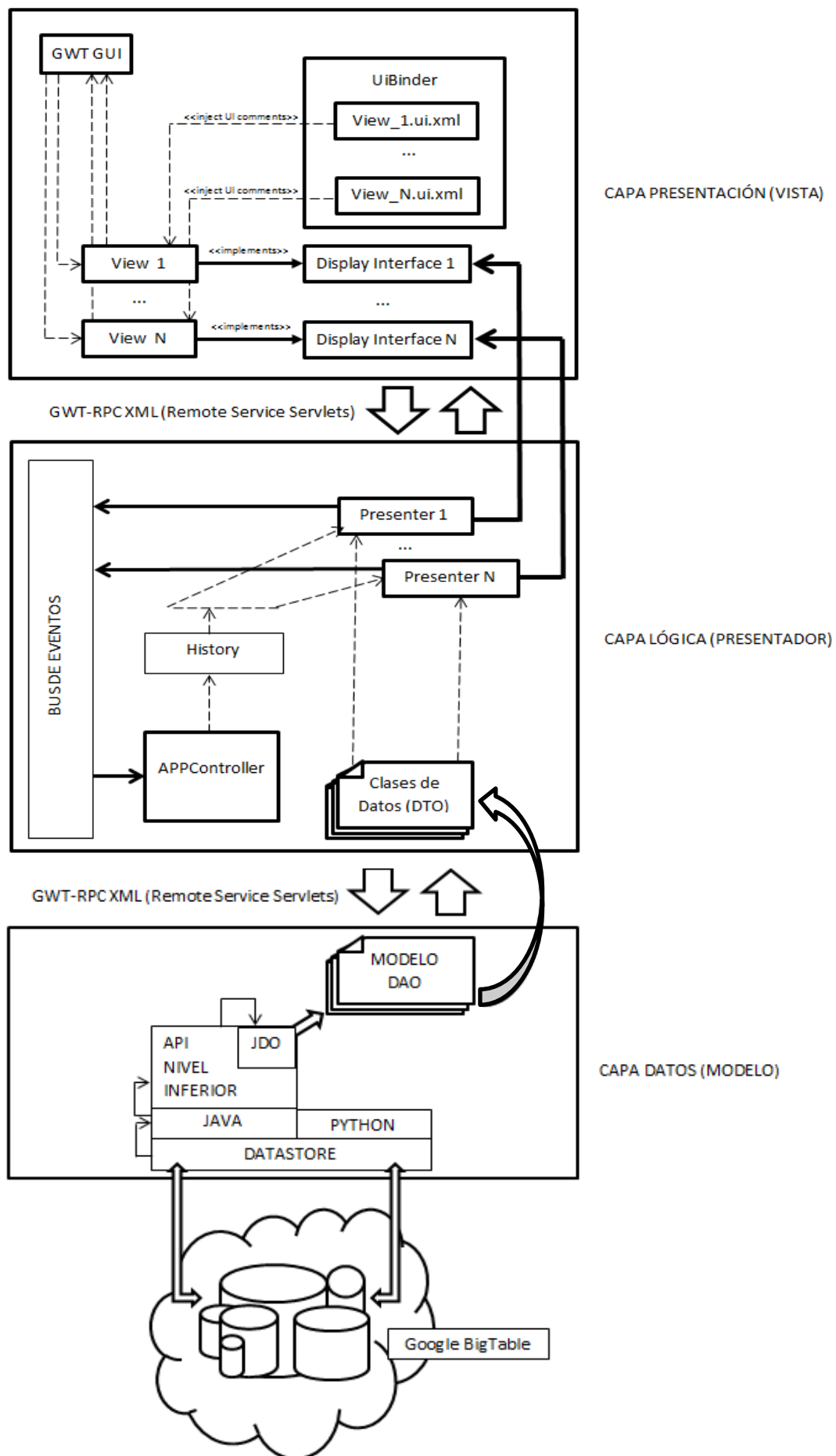
Se comenzará viendo la arquitectura elegida para el sistema, continuando con un grafo de navegabilidad, y adentrándonos posteriormente en el diseño detallado del sistema, en el cual se detallarán más profundamente los componentes del sistema. Finalmente se hará una recopilación de los puntos de variación así como los posibles puntos de evolución del mismo.

8.1.- Arquitectura del sistema

Como puede observarse en la imagen que se muestra a continuación, se ha optado por una arquitectura separada por capas para nuestro sistema. El modelo sigue el patrón arquitectónico MVP (Modelo Vista Presentador), el cual es una variación del clásico MVC (Modelo Vista Controlador).

En esta arquitectura, la capa de presentación o *Vista* es la encargada de mostrar la información que requiera al usuario, la capa de lógica de negocio o *Presentador* recibe la información de la Vista concreta, reclamando a la capa de datos o *Modelo* los datos con persistencia necesarios, para mostrarlos al usuario de la aplicación. Así mismo, si el usuario lo que quiere es almacenar o añadir cualquier tipo de información en el sistema, la *Vista* se lo comunica al *Presentador* que, a su vez, realiza la consulta para persistir la información en el *Modelo*.

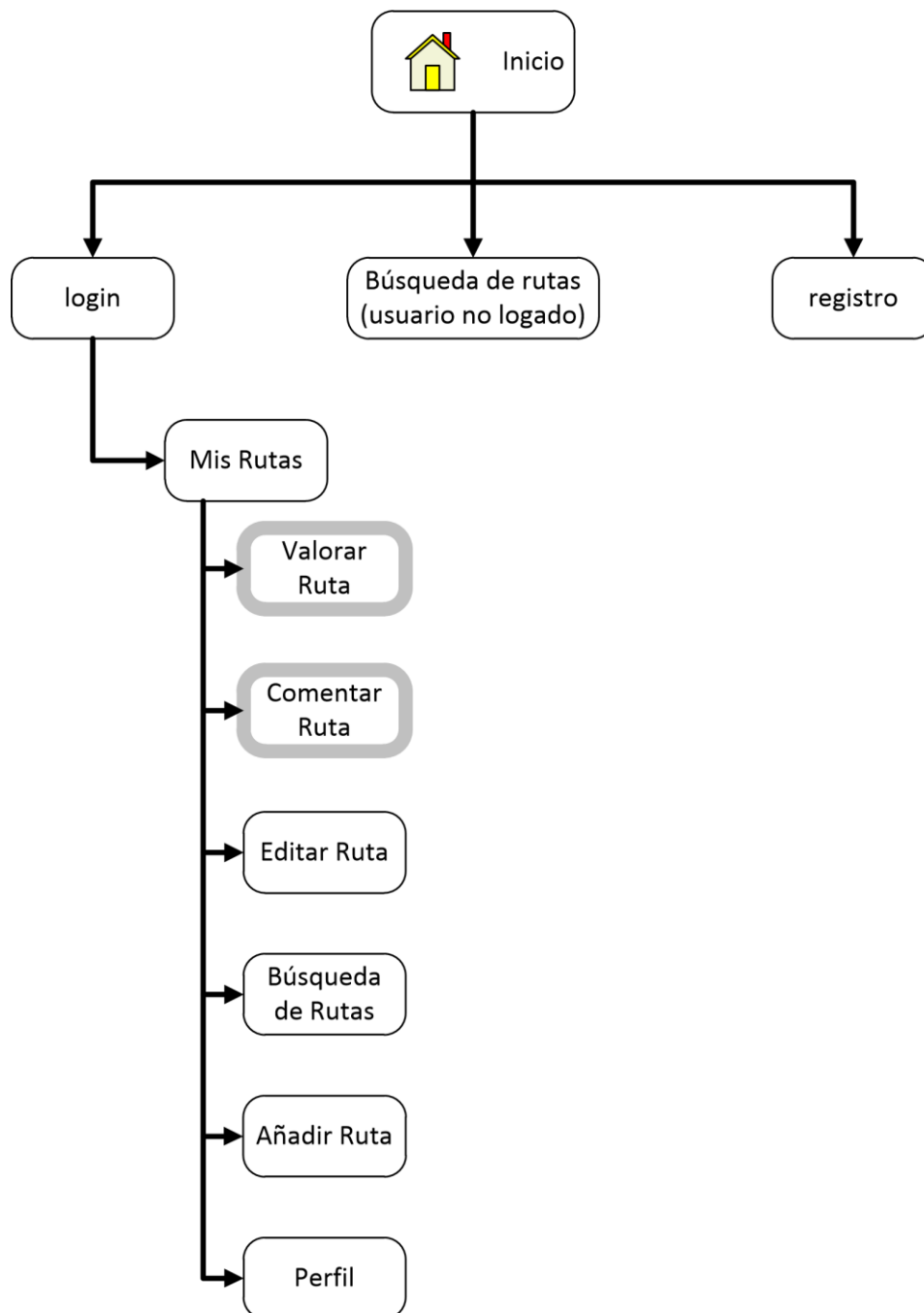
A grandes rasgos, así se realizan las transferencias en el sistema que nos ocupa. A continuación entraremos en detalles sobre las mismas.



8.2.- Grafo de Navegabilidad

El grafo de navegabilidad es una representación gráfica en la que mostramos de forma esquemática los principales marcos que podemos encontrarnos en el espacio de la información de nuestra aplicación y la interrelación entre ellos.

Empezando por el login, el usuario puede recorrer totalmente el subsistema de gestión de rutas y de usuarios a través de las diferentes pantallas (vistas) del mismo.



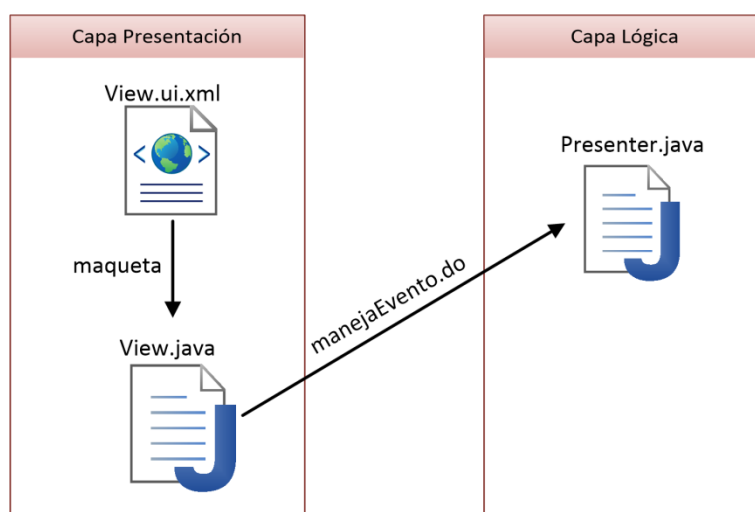
8.3.- Diseño detallado

A continuación procedemos a realizar un diseño detallado de cada una de las capas que componen la aplicación

8.3.1.- Capa de presentación

La capa de presentación es la encargada de mostrar la información a los usuarios de una aplicación. En esta capa se consigue un aislamiento completo del aspecto visual de la aplicación con la lógica de negocio de la misma. Con esto se logra un desacoplamiento muy útil en el caso de que en el futuro se quiera cambiar por completo la interfaz gráfica y el diseño de nuestro sistema.

Esta capa la componen, principalmente, cada una de las vistas de la aplicación. En el siguiente diagrama podemos ver el comportamiento de una vista (View) individual en su interacción con el Presenter correspondiente. Más adelante se detallarán estas relaciones.



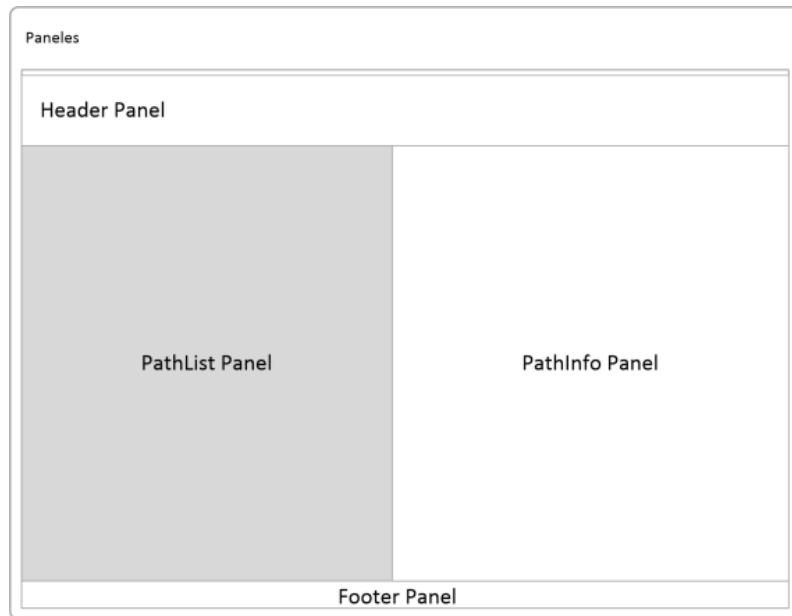
Las acciones que un usuario realiza sobre la aplicación, generan un evento, que será recogido por el Presenter correspondiente, y procederá a llevar a cabo la acción solicitada.

En MyPaths, el corazón de la interfaz de usuario es GWT. Como ya hemos comentado en otras ocasiones, GWT contiene una colección extensible de Widgets y Paneles que no son más que elementos de interfaz de usuario. GWT también provee una estructura o patrón propio llamado UiBinder, a través del cual, mediante anotaciones en xml, podemos crear un esqueleto de lo que será la pantalla por un lado (fichero .ui.xml) y por otro el contenido de la interfaz del panel en concreto. El concepto de “página” queda mediante esta metodología fuera de nuestro sistema.

GWT compila javascript para generar XHTML y lo hace ocultando y mostrando paneles en función al flujo de la aplicación.

Así, usando estos conceptos, la capa de Presentación de MyPaths es un conjunto de pares fichero.ui.xml y fichero.java para cada elemento de interfaz de usuario a presentar en un momento concreto de la aplicación, existiendo además una definición previa de los paneles donde dichos componentes se mostrará o no en un momento dado.

A continuación, vemos cómo podemos configurar los paneles necesarios para nuestra pantalla “Resultados de Búsqueda” vista en los prototipos visuales del capítulo anterior:



Este formato es usado en la mayoría de las pantallas que muestran listas de rutas (concretamente lo hacen en el panel PathList Panel). Así, tendríamos en PathList Panel una tabla compuesta por Rutas, cada celda de esta tabla tendría varios Widgets de GWT: Un Label para el título, Image para el logo del tipo de ruta, Button para “Añadir a mis rutas” y “Ver Ruta”, etc...

En el HeaderPanel y FooterPanel, se crearán los elementos necesarios para que sean plantilla en todas las páginas de la aplicación.

En el momento que el usuario interactúa con alguno de dichos elementos, se dispara un evento que el Presenter correspondiente es encargado de tratar para realizar la acción o acciones que el usuario requiere.

Otra característica propia de la capa de presentación es la gestión de los recursos del sistema. GWT provee un mecanismo para la integración de estilos en una clase java mediante la creación de ClientBundle's. Un ejemplo de ello es EstilosGlobales.java. Este elemento convierte cada clase css creada en Global.css en un método que puede ser invocado en los xml de UiBinder. Este patrón hace que se unifiquen los estilos así como facilitar la inyección de estilos en nuestro sistema para su posterior renderización.

Algo parecido (otro ClientBundle) realiza exactamente la misma estrategia para la gestión de ficheros estáticos de interfaz de usuario, como las imágenes. En la clase RecursosGlobales.java se crea un método (se autogenera a partir del nombre del fichero) para las inserciones de los que sean necesarios en los xml de UiBinder.

Esta manera que tiene de funcionar, a nivel de interfaz de usuario, GWT es muy eficiente, y además, como se comentó anteriormente, el uso de AJAX y comunicación asíncrona hace que, a nivel de Capa de Presentacion, las interfaces que el usuario percibe sean muy amigables.

8.3.2.- Capa de Lógica

La capa de lógica también se conoce como nivel de aplicación. Esta capa es la encargada de recibir las peticiones de los usuarios desde la capa de presentación, llamar a las funciones de acceso a datos si es necesario, hacer los cálculos pertinentes con la información y devolverla a la capa de presentación para que sea visualizada por el usuario.

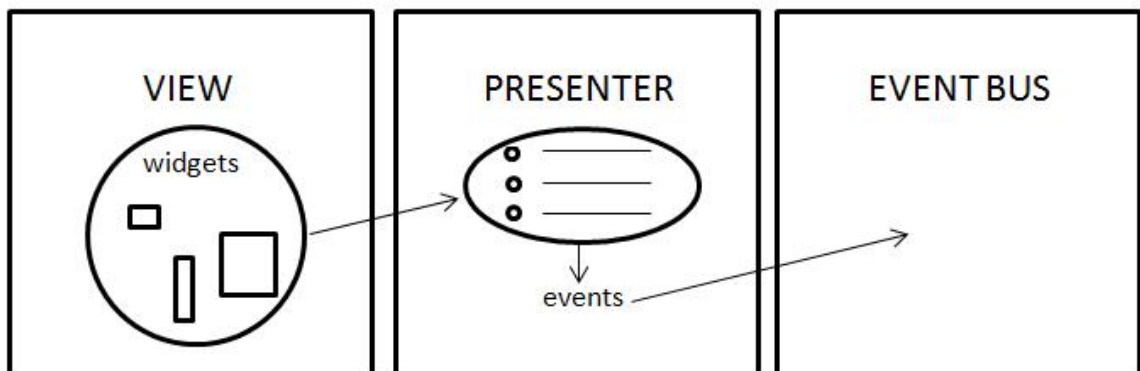
En esta capa se encuentra una clase presentador (Presenter) por cada View de la capa de presentación. Todas estas clases Presenter (del tipo PathListPresenter, PathDetailsPresenter...) implementan la interfaz Presenter.

(DIAGRAMA CLASES PRESENTERS E INTERFAZ PRESENTER)

En aplicaciones complejas, o de peso, realizadas con GWT el modelo MVP funciona a un nivel superior de complejidad mediante la presencia del ApplicationController, encargado de la gestión de todos los elementos de la capa de Lógica. Muchos ven esto como un modelo extendido (MVPa -> Model, View, Presenter, ApplicationController). El flujo de negocio de aplicaciones como PathGenerator es el siguiente:

- 1) Existe un proceso de arranque de GWT llamando al método `onModuleLoad()`
- 2) `onModuleLoad()` crea tres elementos (instancias): un servicio RPC (visto en 3.3.), un Bus de Eventos y el `AppController`.
- 3) El `AppController` recibe instancia del `RootPanel` (Panel principal de la capa de vista que envuelve todos los Paneles vistos en la capa de Presentación).
- 4) A partir de aquí, `AppController` controla los `Presenters`, creándolos y surtiéndoles las Views necesarias para manejar.

Para unir los `Presenters` con las Views, cada `Presenter` contiene una interfaz `Display` definida dentro de los `Presenters`.



Cuando el `Presenter` recibe la petición del usuario, este gestiona la lógica necesaria para gestionarlo, pero, si requiere una gestión de mayor nivel, el evento se pasa al `EventBus`. El Bus de Eventos es una clase de tipo `HandlerManager` y es un mecanismo para gestionar eventos.

En GWT para clase `XXXEvent`, implementa una interfaz `XXXEventHandler`.

(clases EVENTS Y SUS EVENTS HANDLERS EN IMAGEN)

Para cada Evento lanzado, otra función del Bus de Eventos es, además, gestionar qué `Handler` debe encargarse de él, pues puede haber varios a la espera simultáneamente.

8.3.2.1.- Gestión del histórico de navegación y las transiciones entre vistas.

Otras características de las aplicaciones GWT gestionadas en su completitud en la capa de lógica son el Historial de la aplicación y las transiciones entre vistas.. Como se explica en 3.2., GWT usa AJAX, por lo que las páginas (o parte de ellas) se actualizan sin el uso del historial del navegador. GWT añade funcionalidad para la gestión de este problema. La clase ApplicationController implementa ValueChangeHandler y el EventBus le envía HistoryEvents. Existe un estado inicial, y a partir de un sistema de etiquetas a través de tokens para poder llevar al usuario a la vista adecuada. Para manejar esta lógica está el método go().

Manejo del histórico imagen

8.3.2.2.- DTO – Data Transfer Object

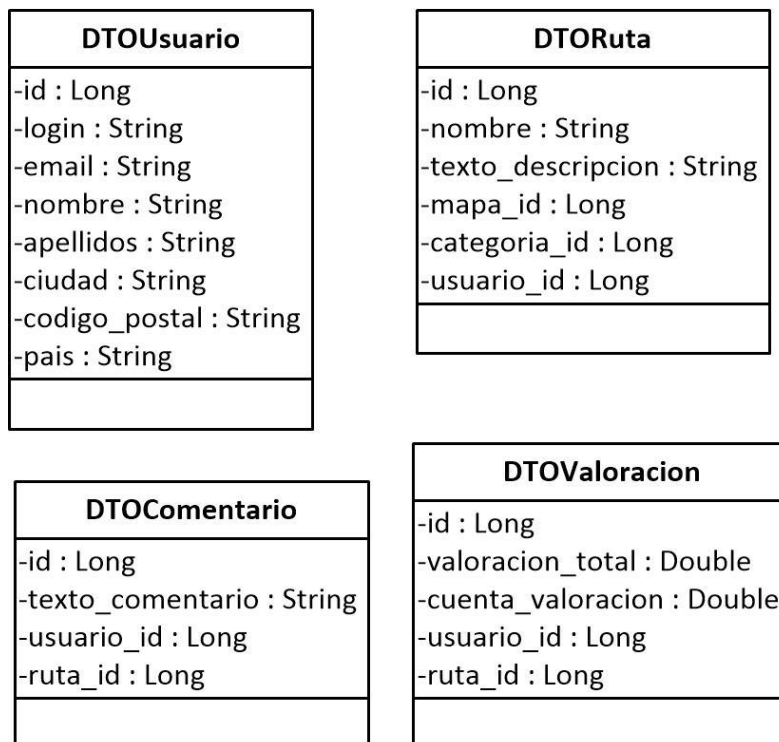
Las clases que conforman el modelo de datos, albergan gran cantidad de información. Muy a menudo, una de estas clases puede contener elementos que a su vez contienen atributos de otras colecciones.

Estos objetos se reciben de la capa de Datos para ser tratados en la capa de Presentación y, a su vez, se transfieren a la capa de Presentación para poder dar al usuario la información que solicita. Los datos acaban siendo de demasiado peso, que consumen demasiada memoria debido a su transferencia entre las 3 capas.

Para solucionar estos inconvenientes se recurre al patrón DTO (Objetos de Tránsito de Datos). Mediante este patrón se encapsulan objetos en objetos de menor tamaño que contienen sólo la información necesaria para mostrar al usuario, despreciando la información no requerida por este en un momento concreto.

Otro uso de este patrón es que puede ser usado para ocultar información que no quiere ser transferida pero que si se desea persistir.

A continuación mostramos un diagrama con los DTO creados para la MyPaths.



8.3.3.- Capa de Datos

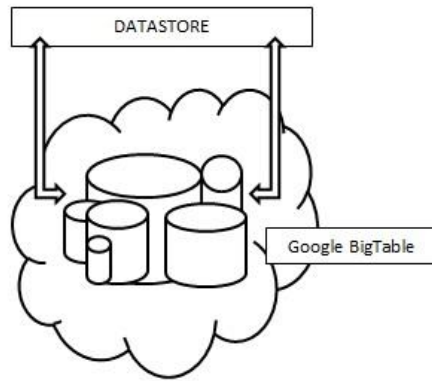
La capa de datos es la capa encargada de proporcionar los accesos a los datos de nuestra aplicación. En MyPaths haremos uso de los servicios que Google ofrece para almacenar la información en la nube a través de Google DataStore, parte fundamental de Google AppEngine (GAE) que ya vimos en 3.5.

En este punto nos encontramos con el concepto de Nube (Cloud). Procederemos a hacerlo comprensible a través de la implementación que Google ofrece con DataStore.

DataStore no es un sistema de gestión de Bases de Datos convencional o, al menos, de uso mayoritario hasta ahora. No es SQL (es NOSQL). Almacena y recupera la información mediante sentencias MySQL (Oracle), pero trabajando de manera diferente.

Google usa GFS (Google File System) que es un sistema de archivos distribuido en la nube. Para ello usa BigTable o Sistema de Gestión de Bases de Datos distribuido y altamente eficiente desarrollado en GFS.

DataStore es una capa por encima de BigTable pensada sobre el concepto de escalabilidad, así conceptos habituales de bases de datos relacionales como normalización o claves foráneas no tienen sentido. Las relaciones entre datos se gestionan almacenando los dos datos unidos, como si fueran uno solo. Esto escala y simplifica las búsquedas (ahorro de la 2ª búsqueda propia de una FK).

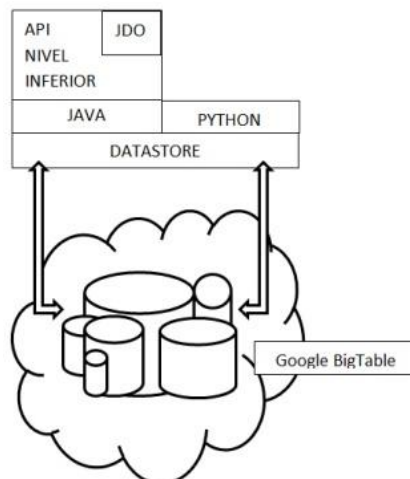


DataStore tiene dos implementaciones de Google: Java y Python. Las operaciones sobre el DataStore requieren el uso de uno de estos dos lenguajes de programación.

A su vez, Java contiene tres maneras de acceder a la información: JDO (Java Data Object), JPA (Java Persistence API) y una API de nivel inferior. Tanto JDO como JPA usan la API de bajo nivel y tienen mayor abstracción. En MyPaths se usa la implementación JDO porque permite a los desarrolladores centrarse más en qué almacenar que en cómo hacerlo y porque tiene una alta abstracción. Además, no se centra en BBDD's relacionales (por esto encaja mejor que JPA).

Características de la API de Bajo Nivel:

- Es la capa más cercana al DataStore.
- Datos y operaciones realmente primitivas.
- Hay que indicar qué y cómo almacenar
- Uso de entidades diferentes a JDO o JPA
- Mapas -> atributos clave-valor. no se centra en BBDD's relacionales (por esto encaja mejor que JPA).

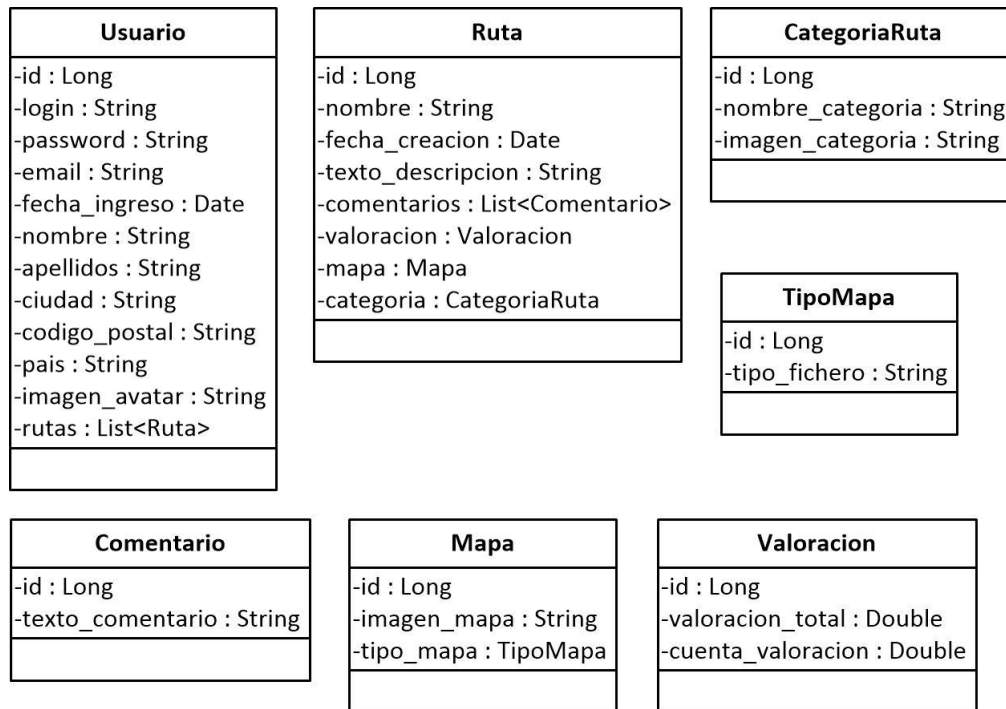


La implementación que GWT usa de JDO se llama DataNucleus. Hay funcionalidades en JDO que no pueden usarse en GWT. Estas son:

- No pueden definirse relaciones sin propiedad.
- No pueden definirse relaciones de propiedad varios a varios.
- No pueden hacerse consultas JOIN
- No pueden hacerse consultas polimórficas.

En JDO, la persistencia de la información se realiza a través de clases Java sencillas llamadas POJO (Plain Old Java Object). Estos tipos de objeto irán acompañados de unas anotaciones que son la manera de inyectar comprensión de JDO para la persistencia y manejo de los datos.

A continuación podemos ver un diagrama con todos los POJOS o entidades de la aplicación.



8.3.4.- Patrones de diseño utilizados

A continuación, se explican los distintos patrones de diseño usados para el diseño del sistema, y dónde se han aplicado.

Patrón MVP – Model View Presenter

Los motivos para la utilización de MVP en el diseño de nuestra aplicación son, primeramente, para poder servirnos de la clara separación en capas y, de este modo, conseguir desacople en el código, pudiendo así en el futuro cambiar los modos de implementar cualquiera de las capas sin tener que modificar las otras. Cada capa, además, tiene su funcionalidad por separado, lo que hace que podamos hacer pruebas unitarias.

También el uso de este patrón hace que, si nuestro modelo, arquitectura, lógica o pantallas crecen, podemos implementarlo sin necesidad de grandes cambios.

Patrón DTO – Data Transfer Object

Los objetos de transferencia de datos se usan para encapsular la información que transferimos entre subsistemas de la aplicación y al nivel de presentación, permitiéndonos así sólo usar los datos del modelo que necesitamos en cada momento, haciendo que las transferencias entre capas sean más eficientes al transferir un tamaño de datos más pequeño. **(ampliar con gráficos)**

8.4.- Condicionantes

8.4.1.- Puntos de variación

En este apartado se listan los posibles cambios que puede sufrir la aplicación, así como su incidencia en nuestro sistema y cómo nuestro diseño se adaptaría a ellos.

Soporte multilinguaje

Sería interesante plantearnos una evolución en este aspecto. GWT soporta perfectamente la gestión de uso de Bundles de cadenas de texto y su selección en función a que el usuario seleccione un idioma diferente ofrecido e, incluso, de cambiar en función a desde dónde se esté accediendo a la aplicación.

El trabajo que requiere es únicamente la traducción de los términos usados.

Futuros cambios en el diseño o ampliación de GWT GUI

En caso de que en un futuro se quiera ampliar las capacidades de GWT, sus Widgets y Paneles, por ejemplo, añadiendo cualquiera de las librerías existentes a tal efecto, es tan sencillo como importarlas en el proyecto.

A su vez, si quisiésemos o necesitáramos cambiar el framework por cualquiera

8.4.2.- Puntos de evolución

Aplicación móvil

Geolocalización de las rutas cercanas a partir de gps

Creación de grupos de usuarios

Gestión de rutas privadas entre grupos

Soporte para más tipos de rutas

8.5.- Modelo de Datos

Incidir en que nuestra aplicación usa la nube.

Ofrecer ventajas de usar la nube.

Explicar funcionamiento de Google Big Data y el concepto de sistema de gestión de bases de datos distribuido.