



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA

INGENIERÍA TÉCNICA INFORMÁTICA DE GESTIÓN

PATHGENERATOR

GESTIÓN DE RUTAS CATEGORIZADAS EN LA NUBE
USANDO GWT, SPRING, DATASTORE Y APPENGINE

Realizado por
Miguel Centeno Muñoz
28633254W

Dirigido por
Sergio Segura Rueda

Departamento
Lenguajes y Sistemas Informáticos

Sevilla, de 2014

Resumen

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

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

Posteriormente a esta primera interpretación del problema, surge otro factor determinante en la funcionalidad de *PathGenerator*: 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.

PathGenerator 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 *PathGenerator* 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 *PathGenerator* 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 GWT (Google Web Toolkit), Google AppEngine (GAE), DataStore, API Google Maps para GWT, Projetsii, SVN, Git, SONAR, Junit.

Tabla de contenidos

Resumen.....	3
Parte I.- Introducción y planificación.....	7
1.- Introducción.....	9
1.1.- Motivación.....	9
1.2.- Objetivos del proyecto.....	10
1.2.1.- Objetivos docentes.....	10
1.2.2.- Objetivos técnicos.....	10
1.3.- Desarrollo del proyecto.....	10
1.3.1.- Projesii.....	11
1.4 Estructura del documento.....	11
2.- Planificación.....	12
2.1.- Planificación temporal.....	12
2.1.1. Estimación de tiempos.....	13
2.1.2.- Desviación respecto estimación de tiempos.....	14
2.2.- Planificación de costes.....	14
2.2.1.- Estimación de costes.....	14
2.2.2- Desviación respecto estimación de costes.....	14
Parte II – Materias relacionadas.....	15
3.- Materias relacionadas.....	17
3.1.- Google Web Toolkit (GWT).....	17

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

En base a aficiones compartidas con conocidos, familiares y amigos, surge la idea de desarrollar NOMBREAPP con el fin de facilitar la organización y planificación del tiempo libre en base a las aficiones de cada uno.

Tras tener la idea del desarrollo de NOMBREAPP como una mera afición, conversaciones entabladas con los potenciales usuarios de mi entorno (clientes) hacen que sea aún más motivante pues entiendo que cubre las necesidades de un amplio espectro de posibles usuarios de diferentes edades y gustos muy distintos.

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

NOMBREAPP 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.

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, como es el caso del framework de desarrollo Spring, Hibernate, AJAX, y GWT y hacer uso de ellas para el desarrollo del proyecto.
- Hacer simulación profesionalizada del ciclo de vida del software y funciones de consultoría del software, análisis de costes y de tiempos.
- Ser capaz de adaptar el proceso de desarrollo de un proyecto software a necesidades reales de clientes.

1.3.- Desarrollo del proyecto

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

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

1.3.1.- 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.

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. Siendo esta parte tercera 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.

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 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:

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

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.

<i>TAREA</i>	<i>DÍAS</i>
<i>T1.- Estudio del problema y prácticas previas realizadas</i>	10
<i>T2.- Familiarización con el entorno de desarrollo y herramientas</i>	7
<i>T3.- Desarrollo de la aplicación</i>	
<i>T3.1. Elicitación de requisitos</i>	14
<i>T3.2. Análisis de requisitos</i>	14
<i>T3.3. Diseño del sistema</i>	5
<i>T3.4. Implementación de la aplicación</i>	28
<i>T3.5. Pruebas Unitarias</i>	3
<i>T3.6. Implantación del sistema</i>	2
<i>T4.- Documentación del proyecto</i>	14
<i>T5.- Manual de usuario</i>	3
TOTAL	100 días

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

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

Diagrama de gantt: TOMSPANNER

2.1.2.- Desviación respecto estimación de tiempos

Pendiente

2.2.- Planificación de costes

Pendiente

2.2.1.- Estimación de costes

Pendiente

2.2.2.- Desviación respecto estimación de costes

Pendiente

Parte II – Materias relacionadas

3.- Materias relacionadas

En el capítulo tercero, se repasan las tecnologías y herramientas de las que se ha hecho uso durante el desarrollo de *PathGenerator*. En ellas presentaremos brevemente sus ventajas y motivos para elegir las para el 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 o desarrolladores.

Otra de las principales ventajas que ofrece GWT es la posibilidad de hacer debug 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.

Escritura de código → Debug → Optimización → Despliegue

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

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

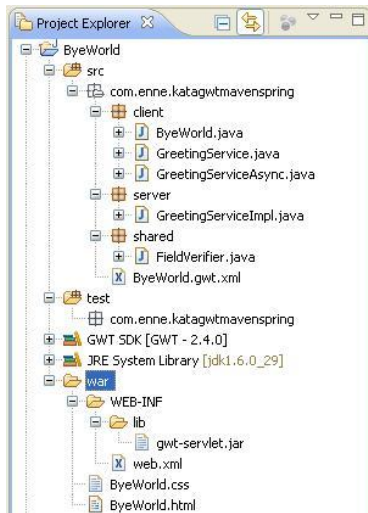


Imagen 1: Estructura de un proyecto GWT en Eclipse.

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

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

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

- Módulo XML (cuyo nombre será: nombreapp.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 (nombreapp.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.

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).

MVP aplicado a Gwt:

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.

Y 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 cuál 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
- ApplicationController

b) Interactuación de los componentes:

- Enlazar Presenter – Views
- Eventos y Bus de Eventos
- Histórico y transición entre las vistas.
- Pruebas (testing)

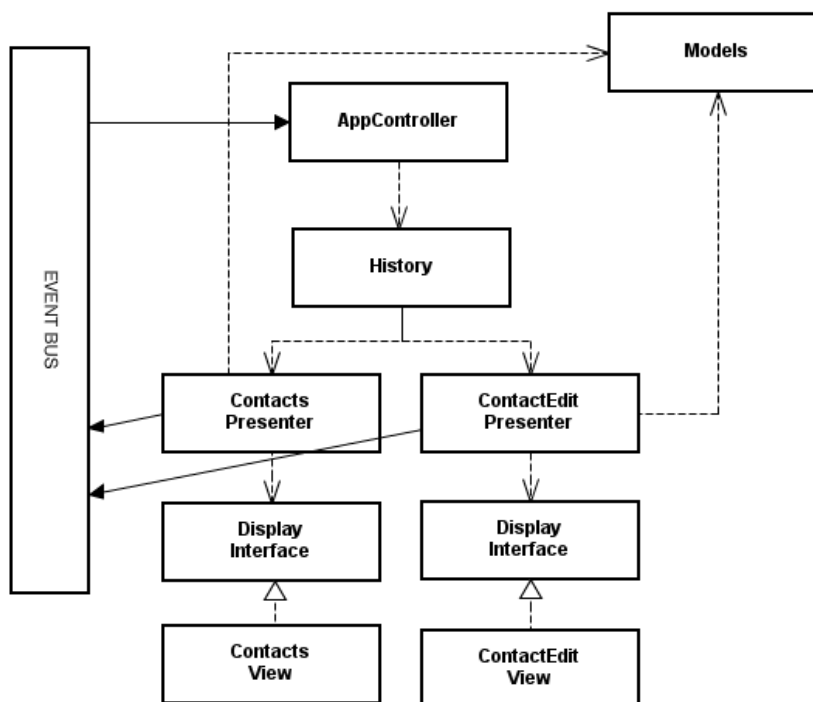


Imagen 2: Diagrama patrón MVP en Gwt en App de prueba de gestión de contactos.

