



CheckTenerife.com
Desafío de OpenData en el Turismo de Tenerife

Challenge of OpenData in Tenerife Tourism

David Rodríguez-Pastrana Parareda

Ingeniería del Software

Escuela Técnica Superior de Ingeniería Informática

Trabajo de Fin de Grado



La Laguna, 9 de julio de 2014

D. **José Luis Roda García**, con N.I.F. 43.356.123-L profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y Sistemas de la Universidad de La Laguna

C E R T I F I C A

Que la presente memoria titulada:

“CheckTenerife.com”
Desafío de Open Data en el Turismo de Tenerife

ha sido realizada bajo su dirección por D. David Rodríguez-Pastrana Parareda, con N.I.F. 40.360.393-Q.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firma la presente en La Laguna a 9 de julio de 2014.

Agradecimientos

Quisiera agradecer al director del proyecto, al señor José Luis Roda García por la ayuda prestada y el hacer posible la realización de este proyecto.

Agradecer al Cabildo de Tenerife por la plataforma web ofrecida sobre la publicación de los datos abiertos desde “<http://www.opendatacanarias.es>”.

También, quiero agradecer al CTO de la empresa FL3XX, al señor Artsiom Anisimau, por todo lo aprendido durante las prácticas en el desarrollo Java y el uso de los frameworks que han sido aplicados en este proyecto.

Por último, agradecer a mi familia por el apoyo recibido durante las etapas difíciles y por ayudarme económicamente en los estudios.

Resumen

El proyecto realizado ofrece un portal web semántico y adaptativo RWD (Responsive Web Design) sobre el turismo de Tenerife gracias a los datos abiertos obtenidos desde internet. El objetivo ha sido dar valor a los datos mostrados desde “OpenData Canarias” y mejorar el SEO (Search Engine Optimization) en la información mostrada. El portal web permite a cualquier turista acceder a toda la información sobre el turismo en Tenerife, pudiendo conocer la dirección de cómo llegar a un destino, geolocalizar los sitios donde comer, los alojamientos, museos, oficinas de turismo, recursos volcánicos, monumentos, playas y conocer los últimos eventos turísticos y los últimos eventos programados en los diferentes museos.

Se trata de un servicio público para que el turista o ciudadano encuentre fácilmente cualquier lugar e información detallada de nuevos sitios que visitar, teléfonos de contacto, horarios de apertura, accesibilidad, categoría, valoración de otros usuarios, ver el modo StreetView, etc.

A través del uso de los Microdatos toda la información expuesta en la web es interpretada por cualquier agente buscador permitiendo ofrecer los resultados buscados con un formato más legible mediante el uso de RichSnippet de Google.

Actualmente, la web se encuentra en línea y se puede acceder desde el enlace “www.checktenerife.com” para consultar los resultados ofrecidos en este proyecto.

Palabras clave

OpenData, SEO, Web Semántica, Microdatos, Geoposicionamiento, Accesibilidad, Turismo.

Abstract

The project done offers a semantic and responsive web portal RWD (Responsive Web Design) about the tourism in Tenerife thanks to the open data obtained from Internet. The aim has been to add value to the public data displayed and obtained from "OpenData Canarias" and improve the SEO (Search Engine Optimization) in the information shown. The web portal allows access to any tourist who searches for tourist information in the island, being able to know the address and directions to a destination, geolocate the places where to eat, lodgings, museums, tourist offices, volcanic resources, monuments, beaches and learn about the new tourist events and the latest programs going on in museums.

This is a public service for the tourist or citizen where he will be able to find easily any place and detailed information about new places to visit, such as telephone numbers, opening hours, type of access, type of category, rating from other users, check the StreetView mode, etc.

Through the usage of Microdata, all information posted on the web is interpreted by any browser and allows delivering the results through a more readable format using RichSnippet from Google.

Currently, the website is online and can be accessed from the link "www.checktenerife.com" to view the results offered in this project.

Keywords

OpenData, SEO, Semantic Web, Microdata, Geolocation, Accessibility, Tourism.

Índice General

1. Introducción	1
1.1. Descripción del contexto.....	1
1.1.1. El marcado semántico y la web semántica.....	1
1.1.2. Apertura de datos públicos OpenData.....	2
1.2. Objetivos.....	3
1.3. Contenido de la memoria.....	4
2. Estudio de viabilidad	6
2.1. Situación actual del mercado laboral.....	6
2.2. Perfil del usuario.....	6
2.3. Objetivos.....	7
2.3.1. Objetivos conseguidos.....	7
2.3.2. Futuros objetivos.....	8
2.4. Sistema realizado.....	9
2.4.1. Descripción.....	9
2.4.2. Método de desarrollo.....	9
2.4.3. Recursos.....	11
2.4.4. Análisis de coste.....	14
2.4.5. Evaluación de riesgos.....	17
2.4.6. Alternativas.....	17
2.5. Planificación.....	19
3. Implementación	22
3.1. Lenguajes de programación y tecnologías utilizadas.....	22
3.2. Configuración del proyecto con Maven y Apache Tomcat.....	22
3.3. Metodologías propias del desarrollo.....	27
3.3.1. Diagrama de Clases.....	28
3.3.2. Diagrama de Objetos.....	29
3.3.3. Clases utilizadas.....	30
3.3.4. Funciones utilizadas.....	30
3.3.5. Nomenclaturas de los campos de la base de datos.....	30
3.3.6. Las clases “LocationServiceImpl” y “LocationServiceDAOImpl”.....	32
3.3.7. Uso de los microdatos para la validación RichSnippet.....	32
3.3.8. Análisis del código del proyecto con SonarQube.....	34
3.3.9. Tiempos ahorrados en la codificación con jRebel.....	36
3.3.10. Problemas en el desarrollo del proyecto.....	37
3.3.10.1. Problema en la configuración del proyecto.....	37
3.3.10.2. Problemas con la obtención de Coordenadas.....	37
3.3.10.3. Problemas con la obtención de imágenes.....	38
3.3.10.4. Problemas con los microdatos.....	39
3.3.10.5. Problemas con la actualización de los Dataset.....	39
3.3.10.6. Problemas con las librerías de jQuery y la API de Google Maps.....	40
4. Resultados	45
4.1. Página de inicio.....	45
4.2. Página de cómo llegar.....	47
4.3. Página de eventos turísticos.....	48
4.4. Página de museos.....	49
4.6. Página de oficinas de turismo.....	53
4.7. Página de alojamientos.....	56

4.8. Página sobre la gastronomía.....	59
4.9. Página de las playas.....	62
4.9. Página de recursos volcánicos.....	65
4.10. Página de monumentos naturales.....	68
4.11. Resultado del diseño web adaptativo RWD.....	71
3. Conclusiones y Trabajos Futuros.....	72
4. Conclusions.....	74
Bibliografía.....	76
Bibliografía Física.....	76
Bibliografía On-line.....	76

Índice de figuras

Figura 2.1.: Coste de instancia m1 de Amazon	13
Figura 3.2.: Diagrama de objetos.....	29
Figura 3.3.: Herramienta SonarQube para analizar código Java	35
Figura 3.4.: Plugin de Eclipse jRebel en ahorro de compilación.....	36
Figura 4.1.: Captura página de inicio.....	45
Figura 4.2.: Captura validación página de inicio	46
Figura 4.3.: Captura página de destino	47
Figura 4.4.: Captura página de eventos turísticos	48
Figura 4.5.: Captura página de museos.....	49
Figura 4.6.: Captura validación página de museos	51
Figura 4.7.: Captura página de eventos en museos.....	52
Figura 4.8.: Captura página de oficinas de turismo	53
Figura 4.9.: Captura validación página de oficinas de turismo	55
Figura 4.10.: Captura página de alojamientos	56
Figura 4.11.: Captura validación página de alojamientos.....	58
Figura 4.12.: Captura página de gastronomía	59
Figura 4.13.: Captura validación página de gastronomía	61
Figura 4.14.: Captura página de playas.....	62
Figura 4.15.: Captura validación página de playas.....	64
Figura 4.16.: Captura página de recursos volcánicos	65
Figura 4.17.: Captura validación página de recursos volcánicos	67
Figura 4.18.: Captura página de monumentos naturales.....	68
Figura 4.19.: Captura validación página de monumentos naturales	70
Figura 4.20.: Captura del diseño web adaptativo RWD	71

Índice de tablas

Tabla 2.1.: Tabla de costes de hardware	14
Tabla 2.2.: Tabla de costes de recursos humanos	14
Tabla 2.3.: Tabla del coste total del proyecto	15
Tabla 2.4.: Tabla de futuros costes de hardware por año.....	15
Tabla 2.5.: Tabla de futuros costes de recursos humanos por año.....	16
Tabla 2.6.: Tabla de futuros costes totales por año.....	16
Tabla 2.7.: Planificación del tiempo empleado en el proyecto	20
Tabla 3.1.: Nomenclatura de los lugares en la base de datos.....	31
Tabla 3.2.: Nomenclatura de las votaciones en la base de datos.....	32

1. Introducción

1. 1. Descripción del contexto

1.1.1. El mercado semántico y la web semántica

La definición de web semántica parte de Mr. Tim Berners-Lee, allá por el año 91 cuando creó la web. Sir Tim Berners-Lee concibió la web como una web separada en dos partes, la web de los documentos y la web de los datos.

La primera resolvía, mediante HTML, el problema básico por el que se creó la web, encontrar una forma de compatibilizar los documentos de las personas independientemente del sistema operativo, o suite de ofimática que utilizaran, y la segunda resolvía, mediante RDF y OWL, el problema de localizar los datos en un formato común que permitiese encontrar relaciones claras entre las publicaciones en internet.

La web de los documentos, Web 2.0, comprende aquellos sitios web que permiten compartir documentos y enlaces de hipertexto y está diseñada para la lectura humana y no para procesar la información que contiene.

Por otro lado, la web de los datos, Web 3.0, es la evolución de la 2.0 para ofrecer semántica en los datos publicados con un significado y puedan ser interpretados por los buscadores de internet.

La posibilidad de localizar y relacionar los datos en la web, permite que los buscadores ofrezcan resultados de forma precisa, evitando información confusa e irrelevante.

Con la web 3.0, se consigue que los datos sean tan accesibles y enlazables como los enlaces de hipertexto. Un ejemplo sería buscar un alojamiento accesible para personas con movilidad reducida. En la Web 2.0, los resultados se tendrán que buscar dentro de la página web y en la Web 3.0 sin acceder al contenido de la web, el buscador es capaz de ofrecer los resultados e indicar el tipo de habitación, acceso al lugar, imagen de la habitación, servicios ofertados, votos de otros usuarios, etc.

Por tanto, se trata del nacimiento de los datos en la web conocido por “Data Web”, donde se consigue publicar la información con significado y que cualquiera otra

persona en internet sea capaz de reutilizar la información fomentando el uso de internet y el acceso a la información mostrada.

Para dotar a la información de significado y crear una Web Semántica han aparecido diferentes tipos de ontologías y anotaciones para marcar los contenidos de la web.

La evolución ha sido lenta en comparación con la Web 2.0 y en un principio se usaron los microformatos sin ser éstos muy útiles debido a su dificultad de marcado y a la complejidad de estructurar los datos.

Una evolución de los microformatos ha sido el lenguaje RDFa a través de los modelos RDF (Resource Description Framework), que sirven para representar los datos en esquemas XML. Este lenguaje ofrece una mayor flexibilidad en cuanto al marcado de contenido, aunque como desventaja los marcadores dependen del tipo de documento teniendo que ser válido únicamente para XHTML, lo cual ofrece una mayor rigidez de cambios y de control en la estructura del código.

Sin embargo, en junio de 2011 nace una herramienta apoyada y desarrollada por los grandes buscadores, **Schema.org** para marcar el contenido como parte de HTML5 a través de los microdatos. El hecho de que esté creada, apoyada y desarrollada por los buscadores, los sitios web van a poder subir su ranking SEO (Search Engine Optimization) al ofrecer la información con semántica. Desde su sitio web, se encuentran publicados una colección de vocabularios de marcado junto con las entidades a las que pertenecen de uso más común como persona, evento, organización, producto, revisión, oferta y un largo etcétera.

1.1.2. Apertura de datos públicos OpenData

El 30 de septiembre de 2010, es una fecha importante para la historia de los datos abiertos cuando el Archivo Nacional del Reino Unido libera una licencia gubernamental de reutilización de los datos generados por esa nación.

La apertura de datos públicos, conocido también como OpenData consiste en poner la información que posee el sector público al alcance de todo el mundo en formatos digitales, estandarizados y abiertos. Al mismo tiempo se facilita el acceso a esta información con el fin de fomentar la reutilización de los datos.

De esta forma, la sociedad – ciudadanos, empresas u otras instituciones - pueden acceder fácilmente a los datos públicos para informarse o para crear nuevos servicios aumentando el valor social de los datos y también poder dar un uso comercial.

Así pues, facilitar la información pública en formatos abiertos para que todo el mundo pueda utilizarlos es ir más allá del simple proceso de permitir la reutilización de la información, es devolver a la sociedad sus datos públicos y fomentar que los utilice todo aquel que lo que desee.

Cabe destacar, que para la publicación de los datos abiertos existen varios tipos de formato como pueden ser RDF, CSV, JSON, XML, RSS o a través de una API pública.

El 3 de agosto de 2013 nace el portal web OpenData Canarias para ofrecer acceso a los datos del sector público con la finalidad de promover y dinamizar la generación de aplicaciones y servicios con valor añadido.

Para ello, se creó un primer concurso el 11 de julio de 2013, donde se premiaron a las personas que fueron capaces de ofrecer una aplicación con la mayor creatividad y utilidad social a través de los datos abiertos expuestos en OpenData Canarias.

Actualmente, hay un segundo concurso disponible desde el 1 de julio de 2014 tras la actualización de la web y subir nuevos ficheros referidos al turismo de Tenerife.

Este proyecto **CheckTenerife.com** va a ser presentado al segundo concurso con el objetivo de mejorar el sector turístico en la isla de Tenerife.

1.2. Objetivos

El objetivo de este proyecto **CheckTenerife.com** es reutilizar los datos abiertos que nos ofrece la web “OpenDataCanarias.es” para ofrecer un servicio centralizado y con valor añadido referido al sector turístico de Tenerife.

Para ello, como objetivos marcados para mejorar los resultados de los lugares turísticos se ha necesitado:

- Crear un portal web disponible para cualquier público.
- Crear una base de datos donde almacenar la información.
- Enriquecer la información mostrada a través del geoposicionamiento y añadir varias opciones como ver el StreetView, votar, filtrar las búsquedas, obtener la ruta de destino y compartir la información en las redes sociales.
- Añadir semántica en los datos para mejorar los resultados en los buscadores mediante los microdatos y presentarlos en formato RichSnippet y subir el posicionamiento de la información, ranking SEO.

1.3. Contenido de la memoria

En el capítulo **2. Estudio de viabilidad** se realiza la planificación realizada para el desarrollo del proyecto donde se comprueba si resulta viable su realización, explicando la situación actual del mercado laboral, los objetivos cumplidos, los costes actuales y futuros, los riesgos, otras alternativas que se tomaron en un principio y la planificación.

En el capítulo **3. Implementación** se presentan los lenguajes de programación utilizados y las metodologías usadas para el desarrollo del proyecto explicando las configuraciones que han sido necesarias, las herramientas utilizadas, el código implementado y los problemas encontrados.

En el capítulos **4. Resultados** se describen y se muestran a través de capturas de pantalla los resultados ofrecidos en cada página del portal web. También se añaden capturas de pantalla sobre el código validado a través de la herramienta de pruebas.

En el capítulo **5. Conclusiones** se muestran las conclusiones obtenidas de la realización de este proyecto y las ampliaciones que se pueden realizar para mejorar el resultado del mismo.

2. Estudio de viabilidad

En este capítulo se analiza la viabilidad del proyecto, realizando un estudio de los medios necesarios para el desarrollo tanto económico como teórico.

2.1. Situación actual del mercado laboral

Actualmente, el turismo de Tenerife no es accesible de forma fácil desde un mismo portal web y tampoco hay herramientas útiles para la búsqueda de información. Poder encontrar cómodamente los sitios donde comer, alojarse, playas, monumentos, museos, recursos volcánicos y eventos son difíciles de buscar en internet al encontrarse publicados en diferentes fuentes de información.

Ofrecer un servicio público que pueda dar valor a los datos sobre el turismo, siendo el mayor recurso económico que tiene la isla, y poner disposición del público un portal web que permita visitar y acceder al mayor número de lugares interesantes de la isla, sería una cuestión de reutilizar los datos basados en el turismo para poder sacarles funcionalidad y provecho en una misma aplicación.

Sabiendo que muchos ciudadanos usan el término “estar aplatanado” por falta de actividad e interés de visitar nuevos sitios en la isla, un portal web que ofrezca utilidad en conocer nuevos lugares permitiría un mayor movimiento tanto de los turistas como de los ciudadanos alrededor de la isla.

En cuanto a cifras sobre lo importante que resulta el turismo en Tenerife, en el año 2013 se recibieron 3,93 millones de turistas extranjeros con un crecimiento del 4,48% respecto al año anterior y superando el récord del año 2012.

En el siguiente enlace se puede comprobar mejor sobre la noticia:

http://www.eldiario.es/canariasahora/economia/Record-Gran-Canaria-Tenerife-Lanzarote_0_220478873.html

2.2. Perfil del usuario

La aplicación web desarrollada está pensada para cualquier turista o ciudadano pueda tener acceso a la información turística en un mismo portal web y accesible desde los dispositivos móviles.

Se espera en un futuro poder crear usuarios administradores para poder tener un control de la información mostrada y puedan modificar la información desactualizada.

2.3. Objetivos

A continuación se presentan los objetivos conseguidos y los que se podrían plantear de cara al futuro en caso de ponerse en marcha y haya un mínimo de visitas diarias en el portal web. También, sería interesante que los organismos públicos estuvieran interesados en formar un equipo de desarrollo para mejorar la aplicación.

2.3.1. Objetivos conseguidos

- Crear una aplicación Web en Java y Wicket.
- Realizar un diseño adaptativo en la web para que se pueda acceder desde cualquier dispositivo móvil, Tablet u ordenador.
- Recoger los datos referidos al sector turístico desde Open Data Canarias (<http://opendatacanarias.es>).
Datasets: alojamientos, restauración, oficinas de turismo y museos.
- Obtener las coordenadas de los establecimientos que carezcan.
- Crear una Base de Datos con las tablas necesarias para guardar la información de cada establecimiento.
- Mostrar los datos almacenados en la Base de Datos mediante el geoposicionamiento y ofrecer un listado que permita ver los detalles de cada establecimiento.
- Añadir nuevas funciones a los datos recogidos, como ver el modo StreetView de la calle, puntuar, obtener la ruta de destino y configurar un filtrado de búsquedas.
- Permitir que los datos publicados puedan ser tratados e interpretados por cualquier computador como los buscadores de internet.
- Participar en el Proyecto Ideas 2014 de la Universidad.
- Publicar el proyecto en Radio Club Tenerife, Cadena SER.

2.3.2. Futuros objetivos

- Corregir y actualizar la información de los ficheros más vulnerables debido al constante cambio de información.
Para ello, OpenData Canarias tendría que tener tanto el ficheros de alojamientos como el de restauración sincronizados con los del Cabildo de Tenerife para que la información sea objetiva. En caso de ponerse en práctica, la aplicación desarrollada sería capaz de actualizar la información mensualmente.
- Mejorar el ranking SEO de la web a través de las herramientas de verificación, que ofrecen recomendaciones y posibles mejoras.
A continuación detallo la puntuación ofrecida en el portal:

Valoración SEO: Un 72% desde <http://seositecheckup.com>

Un 60% desde <http://www.woorank.com>

También, se ha obtenido un buen ranking a través del análisis de posicionamiento global ofrecido por AlexaRank, herramienta que sirve para conocer el número de visitas esperado mundialmente y local. La popularidad del portal tomada por AlexaRank ha sido:

Ranking global: 1.027.000 popular.

Ranking en España: 13.302 popular.

Media de visitas esperadas por día: 15 visitas.

- Seguir añadiendo más información sobre el turismo a medida que se vayan incorporando nuevos ficheros en OpenData Canarias.
- Mejorar la usabilidad del portal y añadir mayor funcionalidad a los datos. Una posible mejoría, sería marcar en el mapa tanto los sitios que han sido visitados como los que faltan por visitar.
- Realizar una Aplicación gratuita para los móviles. Dentro de la aplicación, se añadiría el modelo de cobro dentro la aplicación “Freemium”, de pagar por usar la aplicación sin internet. Los turistas que no tengan tarifa de datos en el móvil y dependan del Roaming con un coste muy elevado, podrían consultar cualquier información turística desde el móvil.

2.4. Sistema realizado

2.4.1. Descripción

Se ha realizado una aplicación web multiplataforma que permite a cualquier turista consultar información de los lugares de mayor interés.

Para la gestión de los datos recogidos de “OpenDataCanarias.es” se ha creado una base de datos en PostgreSQL con 14 tablas, de las cuales 7 son para almacenar la información de los ficheros de alojamientos, restauración, oficinas de turismo, museos, recursos volcánicos, playas y monumentos naturales. Las otras 7 tablas sirven para almacenar las votaciones realizadas de las tablas mencionadas y por tanto son hijas de las mismas.

Para la publicación en internet de un entorno programado con el lenguaje Java, se ha instalado el servidor web Apache Tomcat versión 7.0 en el servidor de Amazon EC2 (Elastic Compute Cloud).

2.4.2. Método de desarrollo

Se han puesto en práctica diferentes frameworks para el desarrollo de este proyecto que son Wicket [1], Spring [2] y Hibernate [3].

A través del framework de Spring, se consigue tener un desarrollo estructurado y bien definido en los directorios, ganando tiempo en el desarrollo y reduciendo posibles errores. Spring, nos permite un mayor control de los objetos creados y es una alternativa al uso común de JavaBeans. Gestiona el almacenamiento de cada objeto y el ciclo de vida a través de sus propios métodos. También, facilita estructurar mejor el código mediante sus declaraciones en las interfaces.

Para estructurar bien el código en Spring, se ha seguido la estructura del patrón de arquitectura MVC (Modelo Vista Controlador).

En la capa de modelo, es donde se encuentra la lógica del programa, la información de cada objeto encapsulada y la mayor parte del código necesario para procesar las peticiones desde la capa vista.

En la capa vista es donde el usuario interactúa con la aplicación, en ella se encuentran los ficheros html que tienen el mismo nombre que los ficheros java para que Wicket actúe como controlador Servlet entre la capa vista y la capa modelo mediante el mismo id especificado en ambos ficheros.

El framework de Wicket, permite una mayor integración del lenguaje Java para el desarrollo de aplicaciones web dinámicas. Las peticiones del usuario son mapeadas en componentes, ofreciendo un control de cada uno para ser procesado nuevamente. Esta framework tiene grandes ventajas al permitirnos actualizar cualquier vista de la página cuando se llama al componente creado.

Como dificultad requiere de un mayor control para cada componente creado y poderlo mostrar en la capa vista debido al uso de métodos propios de la librería y de los id necesarios que van a enlazar la capa vista con la capa modelo.

En cuanto a los accesos de la base de datos (BD), se ha usado el framework Hibernate JPA, que permite utilizar objetos persistentes en los datos en lugar de manipular los datos directamente de la BD. Se han usado clases persistentes en las tablas de la BD relacional creada y que tiene el nombre “ULL”.

Se consigue tener una mayor transparencia de la API de JDBC y el lenguaje SQL a través de la API de Persistencia de Java (JPA).

Como lenguaje de consultas, Hibernate nos facilita los accesos a la BD mediante HQL (Hibernate Query Language), que es un lenguaje de consultas que permite realizar búsquedas, borrados/actualizaciones en masa usando semántica orientada a objetos.

Hibernate, utiliza una sintaxis parecida a SQL, aunque también permite lanzar consultas SQL a través de la API de Programación para QBC (Query By Criteria).

Para crear la BD “ULL” se ha usado PostgreSQL al tener una licencia abierta, estar optimizado para el desarrollo web, tener una buena herramienta de control llamada pgAdmin, usar una sintaxis SQL sencilla, estar disponible para cualquier plataforma, diseñado para grandes volúmenes de datos usando la estrategia MVCC (Multiversion Concurrency Control) y por ofrecer la mayor estabilidad y rendimiento sin uso comercial.

[1] <http://wicket.apache.org>

[2] <http://projects.spring.io/spring-framework>

[3] <http://hibernate.org/orm>

2.4.3. Recursos

Recursos de hardware

Para el desarrollo del proyecto y la validación del correcto funcionamiento, se ha utilizado un ordenador portátil MacBook Pro Retina de 15 pulgadas con las siguientes características:

Procesador: Intel Core i7 CPU, 2.3 GHz

Memoria RAM: 16GB 1600MHz DDR3

Disco duro: 500GB SSD

Gráficos: NVIDIA GeForce GT 750M 2048MB

Sistema operativo: OS X Mavericks

Para el alojamiento del proyecto se ha utilizado el servidor de Amazon EU, que se encuentra en Irlanda y el control del mismo es a través de la interfaz AWS (Amazon Web Services). Para ello, se ha creado una microinstancia (T1 Micro) en el servidor de Amazon Elastic Compute Cloud (Amazon EC2), que ofrece un servicio gratuito durante el primer año. Las características técnicas del servidor para la instancia T1 Micro son:

Procesador: Intel Xeon E5-2670 2.5GHz

Memoria RAM: 0.613GB

Disco duro: 8GB EBS (Amazon Elastic Block Store) con rendimiento constante y baja latencia.

Sistema operativo: Amazon Linux AMI PV EBS 64-bit.

En él se han instalado los dos servicios necesarios para ejecutar el portal web, que son el servicio de Tomcat v.7.0 y el de la base de datos de PostgreSQL v.9.3.4.

En Tomcat, el directorio raíz “/webapps/ROOT” es donde se ha desplegado el fichero WAR del código.

Recursos de software

Para el diseño, análisis y desarrollo del proyecto se han utilizado los siguientes programas:

Servidor de Base de datos: PostgreSQL v. 9.3.4

Editor Java: Eclipse Java EE IDE for Web Developers v. Kepler

Servidor de la aplicación Java: Apache Tomcat v. 7.0

Editor de Base de datos: PgAdmin3 de PostgreSQL

Sistema de control de versiones y seguimiento de proyectos: Github

Análisis del código Java: SonarQube v. 4.3.2
Editor de diagramas UML: Dia v.0.97.2
Editores de texto: Xcode v.5.1.1 y TextMate v.2.0 alpha
Editor de gráficos: Pixelmator v.3.0 FX
Navegadores de páginas web: Firefox v.29.0.1, Safari v.7.0.4 y Chrome v.35.
Plugins de Firefox que ayudan a analizar la codificación y funcionamiento de las páginas web: Firebug v.1.12, Clear Cache v1.4, Web Developer v.1.2.5 y Total Validator v.7.4.0.
Transferencia de ficheros (FTP): Transmit v.4.1.7

Para tener un control y seguimiento del código durante el desarrollo, se ha usado Github descargando el plugin de Eclipse desde el siguiente enlace:

<http://download.eclipse.org/egit/updates>

La dirección de Github donde poder consultar el código del proyecto es:

<https://github.com/alu0100508031/checktenerife>

Para el alojamiento del proyecto en Amazon, se han instalado los siguientes programas a través del terminal accediendo por SSH:

Servidor de Base de datos: PostgreSQL v.9.3.4.
Servidor de la aplicación Java: Apache Tomcat v.7.0

La suscripción gratuita de Amazon es válida sólo para un año y nos permite acceder a la instancia creada ofreciéndonos la siguiente información:

Acceso SSH con clave: `ssh -i ~/checktenerife.pem ec2-user@54.72.199.250`
Transferencia de ficheros (SFTP): `sftp://ec2-user@ec2-54-72-199-250.eu-west-1.compute.amazonaws.com`
DNS público: `ec2-54-72-199-250.eu-west-1.compute.amazonaws.com`
IP fija usada para el dominio: 54.72.199.250

Amazon, también ofrece las instancias reservadas que permiten tener descuentos interesantes en caso de realizar un uso intensivo y continuo del servidor.

Las dos opciones que ofrece son:

- Uso en demanda, donde podremos arrancar y parar el servidor cuando queramos y únicamente se pagan las horas de uso.
- Reserva por 1 o 3 años, se realiza un pago inicial y a cambio se obtiene un descuento sobre las horas de uso.

En el caso de las instancias reservadas existen tres variantes en función del uso que se le de a la instancia (basadas en el tiempo de utilización) y esto va a influir en la facturación. Existe uso ligero, uso medio y uso intenso, relacionados con la utilización que se vaya a realizar del servidor.

Para este portal web es interesante una instancia del tipo m1 (mediana), donde se realizaría el pago por adelantado y dese el momento en que se inicie la instancia, se le asigna el precio correspondiente a la instancia reservada.

Los precios han sido consultados desde el siguiente enlace:

<http://aws.amazon.com/es/ec2/purchasing-options/reserved-instances/>

La reserva para 3 años es la más conveniente ya que se facturan sólo las horas utilizadas y resulta ser más rentable como se observa en el gráfico (Figura 2.1.).

Instancia de Amazon m1 (4GB de RAM y 2 ECU) para 3 años:

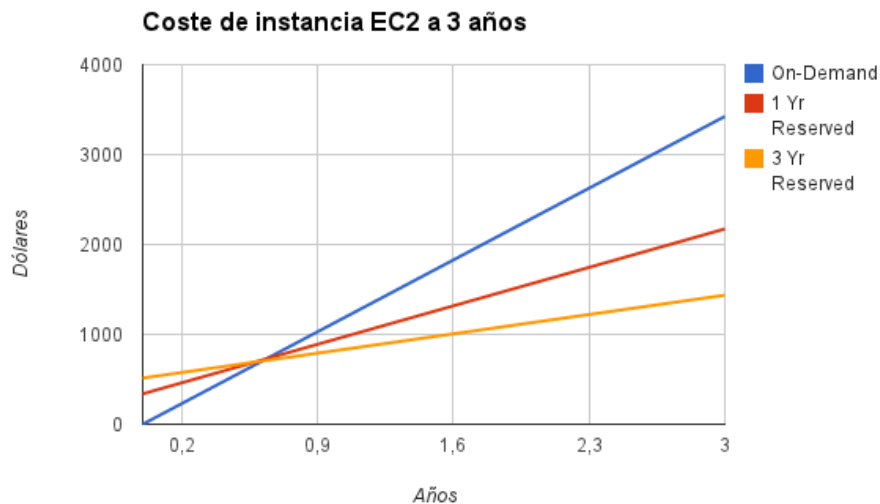


Figura 2.1.: Coste de instancia m1 de Amazon

El coste rondaría los 800€ (267€ / año) en caso de reservar para 3 años de uso aunque podría ser menor dependiendo de la cantidad de horas de uso.

2.4.4. Análisis de coste

Costes presentados en el proyecto

Costes de Hardware

Recurso	Coste
Ordenador portátil utilizado para el desarrollo de la aplicación	2.629 €
Servidor en Amazon EC2 (suscripción válida solo para un año)	0 € / 1 año
Compra de los dominios " www.checktenerife.com " y " www.checktenerife.es "	14€ / 1 año
Total	2.643 €

Tabla 2.1.: Tabla de costes de hardware

Costes de Software

No se calcula un coste adicional ya que el software utilizado es gratuito o ya venía instalado en la compra del ordenador utilizado.

Costes de Recursos Humanos

Recurso	Iniciales	Trabajo	Coste	Coste Total
Director del proyecto	DP	100h	0€ / h	0 €
Programador	P	600h	20€ / h	12.000 €
Total		400h	-	12.000 €

Tabla 2.2.: Tabla de costes de recursos humanos

Coste total del proyecto

Recurso	Coste
Costes de Hardware	2.643 €
Costes de Software	0 €
Costes de Recursos Humanos	12.000 €
Total	14.643 €

Tabla 2.3.: Tabla del coste total del proyecto

Costes esperados de cara al futuro

Para continuar el desarrollo del proyecto, se podrían estimar unos costes anuales de cara a mejorar y mantener el servicio en línea.

Costes de Hardware anuales

Recurso	Coste
Servidor de Amazon (suscripción de cobro)	267 € / año
Compra de los dominios “ www.checktenerife.com ” y “ www.checktenerife.es ”	14€ / año
Total	281 € / año

Tabla 2.4.: Tabla de futuros costes de hardware por año

Costes de Software anuales

No se calcula ningún coste adicional ya que todo el software utilizado ha sido de código abierto.

Costes de Recursos Humanos anuales

Recurso	Iniciales	Trabajo	Coste	Coste Total
Director ejecutivo (CEO)	CEO	800h / año	0€ / h	0 €
Director de tecnología (CTO)	CTO	960h / año (media jornada)	20€ / h	19.200 € / año
Programador Java	P	960h / año (media jornada)	15€ / h	14.400 € / año
Total		2.720 h / año	-	33.600 € / año

Tabla 2.5.: Tabla de futuros costes de recursos humanos por año

Coste total esperado anualmente

Recurso	Coste
Costes de Hardware	281 € / año
Costes de Software	0 € / año
Costes de Recursos Humanos	33.600 € / año
Total	33.881 €

Tabla 2.6.: Tabla de futuros costes totales por año

2.4.5. Evaluación de riesgos

Evaluación de riesgos relacionados con la formación de nuevas tecnologías

- No poder alcanzar los conocimientos necesarios sobre los frameworks de Wicket, Spring y Hibernate, lo que provocaría necesitar una programación más al alcance de la mano, como por ejemplo programar la aplicación directamente en PHP y usar un CMS que facilite las tareas.

Evaluación de riesgos relacionados con el análisis y desarrollo de la aplicación

- No conseguir definir un entorno de usuario adecuado que sea lo suficientemente práctico e intuitivo sin perjudicar la usabilidad de la herramienta.
Falta de una implementación con medidas de seguridad que pueda alterar y/o dar un mal uso de la información.
- No conseguir una estructura de programación modular que facilite la incorporación de nuevas tecnologías en un futuro.

Evaluación de riesgos relacionados con la implementación

- No realizar correctamente la fase de pruebas que permita localizar errores y mejoras en el acabado final de la aplicación.
- No conseguir un funcionamiento óptimo en la carga del servidor de Amazon para el correcto funcionamiento del servicio.
- Abandono del tipo de tecnología usada al tratarse de una programación más avanzada y por requerir mayor dedicación de aprendizaje.

2.4.6. Alternativas

CMS's Drupal [4], Wordpress [5], Joomla [6], Magnolia [7]

La opción de utilizar un CMS fue descartada ya que además de ser difícil de encontrar uno desarrollado en Java y que pueda integrarse con la aplicación desarrollada, ofrece las siguientes dificultades que hacen que se descartara:

- Un CMS resulta muy práctico si las funcionalidades coinciden con los requisitos, siendo estos simples y con un diseño simple. Como ejemplos, podrían ser el registro y control de usuarios, crear una tienda online, publicitar un comercio, etc.
- Ofrecen un exceso de utilidades no necesarias que provoca una sobrecarga de los recursos de la memoria del hardware empeorando los tiempos de ejecución de la aplicación.
- Ofrecen una base de datos con demasiada información sin usar, lo que conlleva mayor carga y un bajo nivel de integración.

- En algunos casos, aunque sean productos gratuitos, cuando se requiere una documentación más elaborada o una actualización superior del CMS hay que suscribirse a un uso de pago.

[4] <https://www.drupal.org>

[5] <http://wordpress.com>

[6] <http://www.joomla.org>

[7] <http://www.magnolia-cms.com>

Lenguaje PHP [8]

A pesar de ser un lenguaje muy versátil, sencillo y documentado, se trata de un lenguaje que se interpreta en ejecución y por tanto no necesita de una separación de capas, lo que hace que sea más vulnerable a agujeros de seguridad. No permite tener un diseño escalable ni orientado a objetos (OOP), donde el código HTML, SQL y PHP puede encontrarse mezclado ofreciendo una mayor dificultad de mantener el código y añadir nuevos módulos. También, es un lenguaje orientado únicamente al desarrollo web sin posibilidad de integrarlo en plataformas móviles.

[8] <http://www.w3schools.com/PHP>

Lenguaje JSP [9]

Como inconveniente presenta un bajo nivel de abstracción para poder reutilizar los componentes a nivel de servidor, teniendo que duplicar código ya implementado y manteniendo la capa de lógica de negocio mezclada con el código HTML.

[9] <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>

Frameworks Java para la capa de presentación: JSF [10], Struts [11], Stripes [12], Tapestry [13]

La metodología de estos frameworks es parecida a la de Spring, pero con la diferencia que muchos de éstos no son compatibles y recomendables para el uso de Wicket por estar orientado a la reutilización de componentes Java y no ofrecen la misma compatibilidad.

[10] <http://www.eclipse.org/webtools/jsf/main.php>

[11] <http://struts.apache.org>

[12] www.stripesframework.org

[13] <http://tapestry.apache.org>

2.5. Planificación

A continuación (Tabla 2.7) se detalla la planificación tomada para la realización del proyecto en 9 fases principales. Se calcula el trabajo realizado entre los periodos 01/08/2013 y 09/07/2013.

Id	Nombre de la tarea	Duración	Comienzo	Fin	Predecesora
1	Inicio del proyecto: asignación y conformidad del proyecto.	1 hora	01/08/13	01/08/13	
2	Planificación.	5 días	01/08/13	05/08/13	
3	Estudio de viabilidad	8 horas	01/06/13	05/08/13	1
4	Investigación de los fundamentos teóricos.	30 días	05/08/13	04/09/13	
5	Investigación de entornos de programación web CMS, PHP, Java	10 días	05/08/13	15/08/13	3
6	Investigación de Wicket, Spring, Hibernate	14 días	15/08/13	29/08/2013	5
7	Investigación de entornos de servidor Linux y Apache Tomcat	5 días	29/08/2013	03/09/2013	6
8	Redacción de la documentación de fundamentos teóricos	1 día	03/09/2013	04/09/2013	7
9	Análisis de la aplicación.	27 días	04/09/2013	30/09/13	
10	Análisis de requisitos (diagrama de casos de uso)	7 días	04/10/13	11/10/13	7
11	Análisis de datos (diagrama de clases)	7 días	11/10/13	18/10/13	10
12	Análisis de seguridad	7 días	18/10/13	25/10/13	11
13	Documentación del análisis	4 días	25/10/13	29/10/13	12
14	Aprobación del análisis	2 días	29/10/13	31/10/13	13
15	Diseño de la aplicación.	62 días	31/10/13	31/12/2013	
16	Diseño de la base de datos	10 días	31/10/13	09/11/13	14
17	Diseño funcionalidad	7 días	09/11/13	15/11/13	15
18	Diseño de la interfaz	30 días	15/11/13	15/12/13	16
19	Diseño de las pruebas	8 días	15/12/13	23/12/13	17

Id	Nombre de la tarea	Duración	Comienzo	Fin	Predecesora
20	Documentación del diseño	5 días	23/12/13	27/12/13	18
21	Aprobación del diseño	5 días	27/12/13	31/12/13	19
22	Desarrollo de la aplicación.	151 días	31/12/13	25/05/14	
23	Preparación del entorno de desarrollo	10 días	31/12/13	09/01/14	21
24	Configuración de la base de datos	5 días	09/01/14	14/01/14	23
25	Programación de la aplicación	100 días	14/01/14	20/04/14	24
26	Desarrollo de la interfaz de usuario	36 días	20/04/14	25/05/14	25
27	Test y pruebas.	13 días	25/05/14	07/06/14	
28	Realización de las pruebas	10 días	25/05/14	04/06/14	26
29	Documentación del desarrollo y test	1 día	04/06/14	05/06/14	28
30	Aprobación del desarrollo y pruebas	2 días	05/06/14	07/06/14	29
31	Implementación.	30 días	08/06/14	08/07/14	
32	Utilización de la aplicación para consultar información turística	7 días	08/06/14	15/06/14	30
33	Pruebas sobre los Microdatos	7 días	15/06/14	22/06/14	32
34	Pruebas sobre el SEO web	16 días	22/06/14	08/07/14	33
35	Cierre del proyecto.		09/07/14	09/07/14	
36	Defensa del proyecto.	1 hora	20/07/14	20/07/14	

Tabla 2.7.: Planificación del tiempo empleado en el proyecto

3. Implementación

En este capítulo se describe la implementación de la aplicación, mencionando los lenguajes de programación y las tecnologías utilizadas, las metodologías propias del desarrollo y los problemas que se han encontrado.

3.1. Lenguajes de programación y tecnologías utilizadas

Para la realización del portal web se han utilizado las siguientes tecnologías:

- **Lenguajes de programación:** Java, Wicket, Spring, Hibernate, jQuery, HTML y CSS.
- **Servidores que necesita la aplicación para funcionar:** PostgreSQL y Apache Tomcat.
- **Programas que han ayudado a editar la programación:** Eclipse, plugin Firebug de Firefox que ayuda a ver la ejecución del código y sus interacciones de CSS y JavaScript de una página web desde el navegador y el editor de bases de datos pgAdmin de PostgreSQL.

3.2. Configuración del proyecto con Maven y Apache Tomcat

Como herramienta de programación se ha utilizado “Eclipse for Java Developers” versión Kepler, al ser la más útil para crear un proyecto web en Java con el gestor de directorios y de actualizaciones Maven y añadir compatibilidad con la mayoría de plugins necesarios y que explico más adelante cuáles han sido usados.

Se ha usado el servidor de Apache Tomcat versión 7.0 dentro de la herramienta de Eclipse para desplegar el proyecto y así poder ver los resultados desde el navegador a través de la url “http://localhost:8080”.

Para la configuración y gestión de los directorios con Maven y el framework de Wicket se ha seguido el siguiente manual:

<https://wicket.apache.org/start/quickstart.html>

El código escrito en el terminal para crear los directorios iniciales del proyecto y luego importar a Eclipse ha sido:

```
mvn archetype:generate -DarchetypeGroupId=org.apache.wicket -  
DarchetypeArtifactId=wicket-archetype-quickstart -DarchetypeVersion=6.16.0 -  
DgroupId=com.checktenerife -DartifactId=ull -  
DarchetypeRepository=https://repository.apache.org/ -DinteractiveMode=false
```

Dentro del fichero de configuración de Maven, fichero “pom.xml” se han añadido otras librerías para usar los frameworks de Spring, Hibernate, PostgreSQL para conectar a la base de datos, WordUtils para dar formato a los campos, API de Google Places para obtener nuevas coordenadas desde Java.

Configuración añadida al fichero POM de Maven:

```
<!-- Spring -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-web</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-orm</artifactId>
  <version>${spring.version}</version>
</dependency>
<!-- Hibernate -->
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>${hibernate.version}</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-entitymanager</artifactId>
  <version>${hibernate.version}</version>
</dependency>
<!-- PostgreSQL -->
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <version>${postgresql.version}</version>
</dependency>
<dependency>
  <groupId>commons-dbcp</groupId>
```

```
        <artifactId>commons-dbc</artifactId>
        <version>${dbc.version}</version>
    </dependency>
    <!-- Geocoder -->
    <dependency>
        <groupId>com.google.code.geocoder-java</groupId>
        <artifactId>geocoder-java</artifactId>
        <version>${geocoder.version}</version>
    </dependency>
    <!-- WordUtils -->
    <dependency>
        <groupId>org.apache.commons</groupId>
        <artifactId>commons-lang3</artifactId>
        <version>${wordutils.version}</version>
    </dependency>
```

Para la configuración del framework Spring, se ha tenido que modificar el fichero “/WEB-INF/web.xml” para ejecutar el servicio.

Configuración añadida al fichero WEB de Maven:

```
<filter>
    <filter-name>encoding-filter</filter-name>
    <filter-class>
        org.springframework.web.filter.CharacterEncodingFilter
    </filter-class>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>UTF-8</param-value>
    </init-param>
    <init-param>
        <param-name>forceEncoding</param-name>
        <param-value>>true</param-value>
    </init-param>
</filter>
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener
</listener-class>
</listener>
<listener>
    <listener-class>
        org.springframework.web.context.request.RequestContextListener
    </listener-class>
</listener>
```

Se ha tenido que crear un nuevo fichero el funcionamiento de Spring y Hibernate JPA, y la base de datos (BD) de PostgreSQL. Para ello, se ha creado un nuevo fichero “applicationContext.xml” donde se añade la configuración de la BD.

Fichero creado para el uso de PostgreSQL, Spring y Hibernate:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

        xmlns:context="http://www.springframework.org/schema/context"
        xmlns:tx="http://www.springframework.org/schema/tx"
        xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-3.0.xsd">
        <context:component-scan base-package="com.checktenerife.tf" />
        <bean id="jpaVendorAdapter"
class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
        <property name="databasePlatform"
value="org.hibernate.dialect.PostgreSQL82Dialect" />
        <property name="generateDdl" value="true" />
        <property name="showSql" value="false" />
        </bean>
        <bean id="jpaDialect" class="org.springframework.orm.jpa.vendor.HibernateJpaDialect"
/>

        <bean id="entityManagerFactory"
class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
        <property name="dataSource" ref="dataSource" />
        <property name="jpaVendorAdapter" ref="jpaVendorAdapter" />
        <property name="persistenceXmlLocation" value="classpath*:META-
INF/persistence.xml" />
        <property name="persistenceUnitName" value="ULL" />
        <property name="jpaDialect" ref="jpaDialect" />
        <property name="jpaPropertyMap">
        <props>
        <prop
key="hibernate.cache.use_second_level_cache">false</prop>
        <prop key="hibernate.cache.use_query_cache">false</prop>
        </props>
        </property>
        </bean>
        <bean id="mainDataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-
method="close" lazy-init="false">
        <property name="driverClassName" value="org.postgresql.Driver" />
        <property name="url" value="jdbc:postgresql://127.0.0.1/ULL" />
        <property name="username" value="postgres" />
        <property name="password" value="postgres" />
        <property name="maxActive" value="10" />
        <property name="maxIdle" value="25" />
        <property name="minIdle" value="3" />
        <property name="minEvictableIdleTimeMillis" value="10000" />
        <property name="timeBetweenEvictionRunsMillis" value="20000" />
        <property name="validationQuery" value="SELECT 1"/>
        <property name="testOnBorrow" value="false" />
        <property name="testOnReturn" value="false" />
        <property name="testWhileIdle" value="true" />
        <property name="maxWait" value="8000" />
        <property name="poolPreparedStatements" value="true" />
        </bean>
        <bean id="dataSource"
class="org.springframework.jdbc.datasource.LazyConnectionDataSourceProxy">

```

```

        <property name="targetDataSource">
            <ref local="mainDataSource" />
        </property>
    </bean>
</context:annotation-config/>

    <bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">
        <property name="entityManagerFactory" ref="entityManagerFactory" />
    </bean>
    <bean id="persistenceExceptionTranslationPostProcessor"
class="org.springframework.dao.annotation.PersistenceExceptionTranslationPostProcessor" />
    <tx:annotation-driven />
</beans>

```

Para configurar la persistencia de los Beans en la BD y almacenar todos los campos directamente tras iniciar el servidor, se ha creado el fichero “persistence.xml” para iniciar la creación de todas las tablas junto con los atributos declarados:

Fichero creado para el uso de Hibernate:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Persistence deployment descriptor for dev profile -->
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd"
    version="2.0">
    <persistence-unit name="ULL" transaction-type="RESOURCE_LOCAL">
        <provider>org.hibernate.ejb.HibernatePersistence</provider>
        <properties>
            <property name="hibernate.dialect"
value="org.hibernate.dialect.PostgreSQLDialect" />
            <property name="hibernate.hbm2ddl.auto" value="validate" />
            <property name="hibernate.show_sql" value="false" />
            <property name="hibernate.format_sql" value="false" />
            <property name="hibernate.connection.driver_class"
value="org.postgresql.Driver" />
            <property name="hibernate.connection.url"
value="jdbc:postgresql:ULL" />
            <!--<property name="hibernate.connection.username" value="postgres"
/>
            <property name="hibernate.connection.password" value="postgres" /> --
        >
            <property name="hibernate.connection.pool_size" value="2" />
        </properties>
    </persistence-unit>
</persistence>

```

3.3. Metodologías propias del desarrollo

En este apartado se explica las metodologías definidas a la hora de programar el código de la aplicación, tales como: las clases utilizadas, nomenclaturas utilizadas en el análisis y desarrollo de la aplicación y los pasos que se podrán seguir en un futuro para incluir para mejorar la aplicación.

3.3.1. Diagrama de Clases

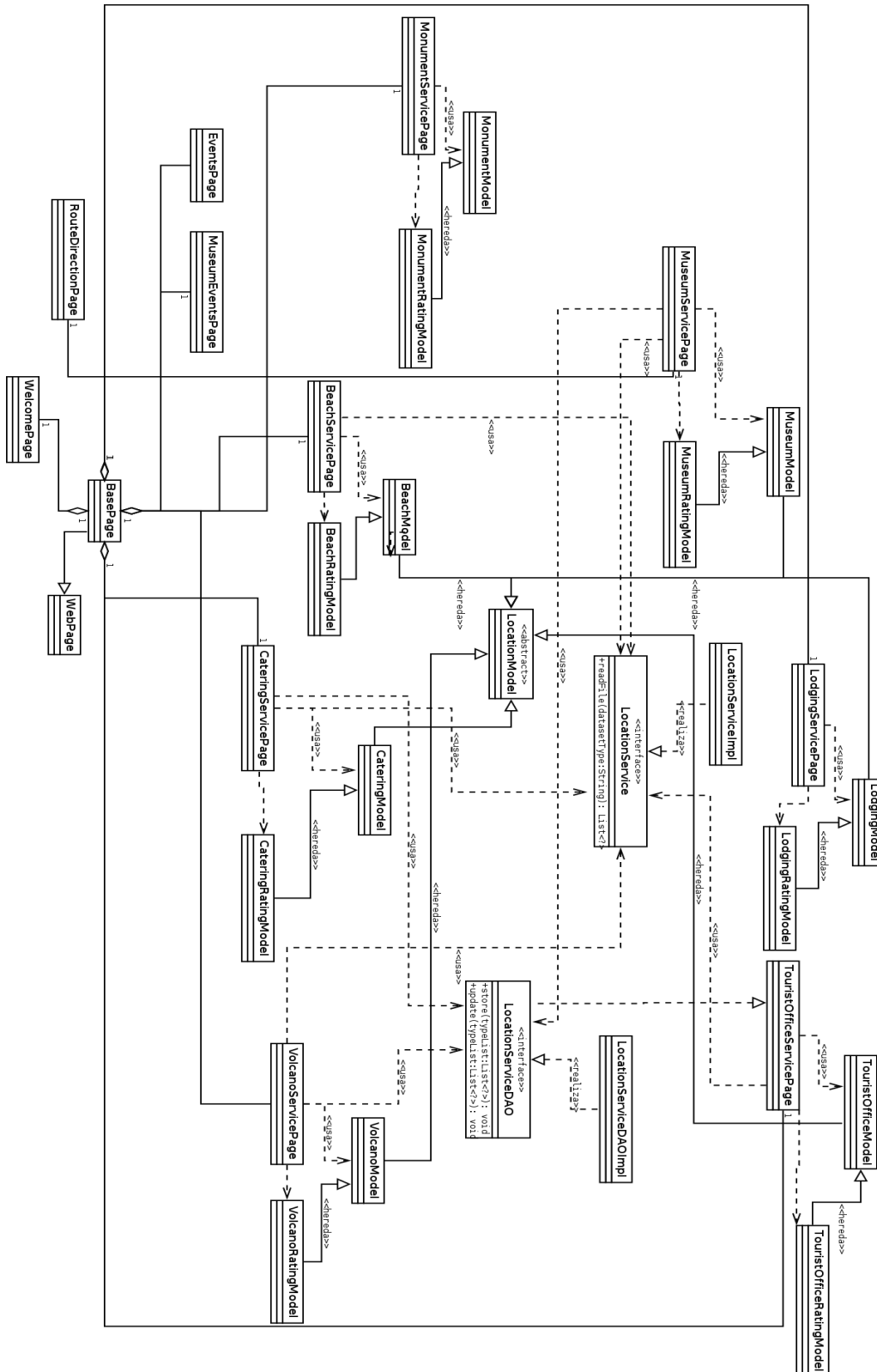


Figura 3.1.: Diagrama de clases

3.3.2. Diagrama de Objetos

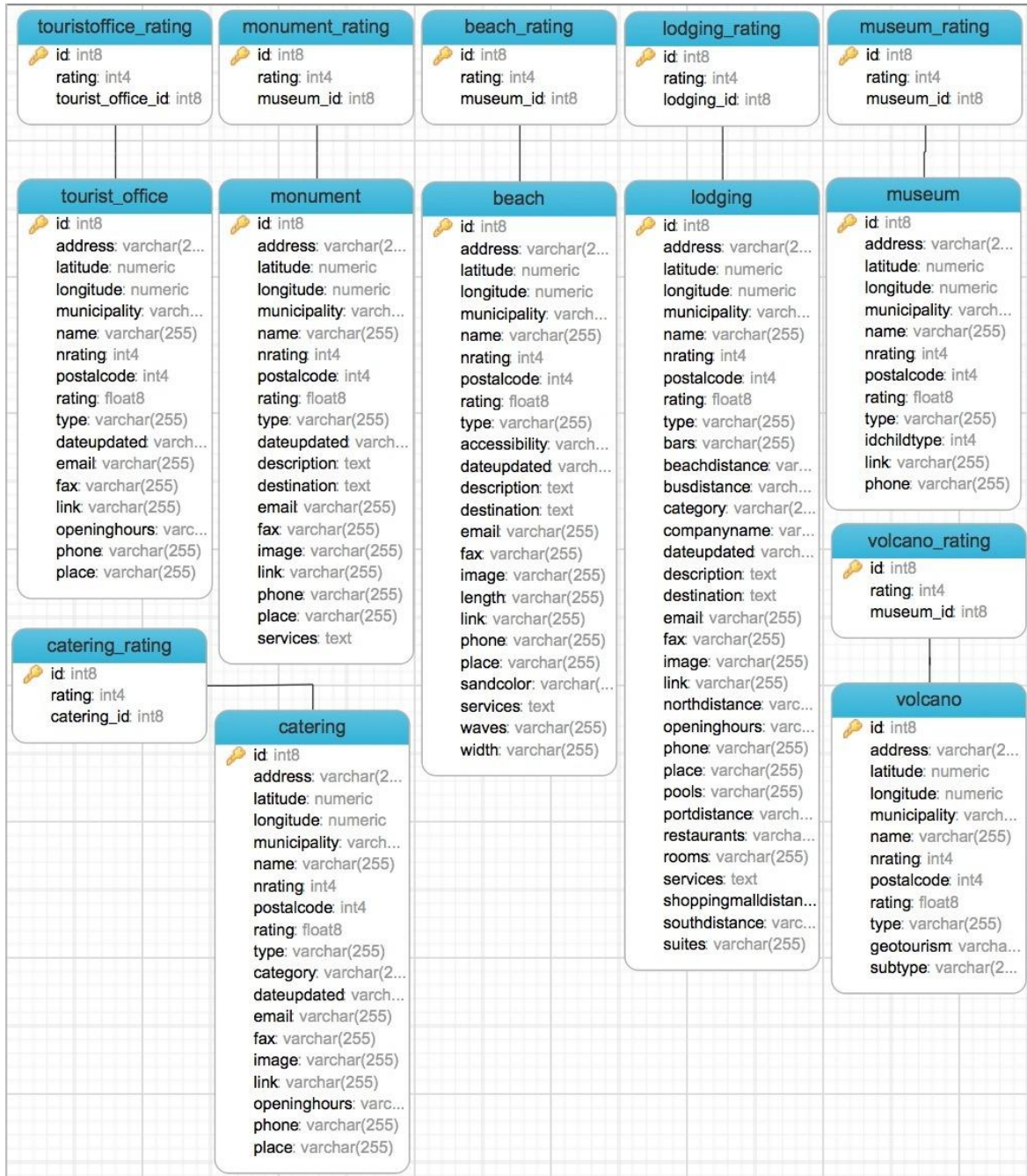


Figura 3.2.: Diagrama de objetos

3.3.3. Clases utilizadas

Con el objetivo de satisfacer la metodología MVC impuesta por el framework Spring, en la realización de la programación se ha dividido las clases definidas en los directorios con diferentes nomenclaturas:

Las clases con extensión “DAO” y “DAOImpl” de la capa Controlador, las clases con extensión “ServiceImpl” y “ServicePage” de la capa Modelo que son las encargadas de crear la información que se presentará en la web y las páginas HTML de la capa Vista. Para transportar los datos utilizados en la aplicación de una capa a otra se utilizan las clases Entity y las clases Bean con extensión “Model” que son clases tipo JavaBeans con la diferencia que las Entity están definidas por el framework Hibernate y se encargan de generar automáticamente las tablas en la base de datos, ahorrando al programador tener que realizar esta parte del código.

3.3.4. Funciones utilizadas

Dentro de las clases de la capa Modelo, que como se ha comentado es donde se centra la mayor parte de la programación del desarrollo, se suele seguir una nomenclatura y orden con los nombres de los métodos, consiguiendo así una mejor estructura de la programación y una mejor claridad a la hora de saber qué hace cada método:

- El método “readFile” se encarga de transferir los datos desde la capa Controlador a todas las clases Beans con extensión “Model” mediante el uso de otros métodos para almacenar toda la información de los ficheros CSV leídos desde la url de OpenData Canarias.
- Los métodos que comienzan por la nomenclatura “store” se encargan de almacenar toda la información de las clases Beans en la base de datos de PostgreSQL y con nombre “ULL”.
- Los métodos que comienzan con la nomenclatura “store” son las funciones que dan la orden de modificar el valor del contenido de la base de datos.
- Para el resto de métodos se utilizan otras nomenclaturas de menor relevancia.

3.3.5. Nomenclaturas de los campos de la base de datos

En la definición de los campos de las 14 tablas creadas también se sigue una nomenclatura que indica qué tipo de dato es cada campo. Se han programado los Beans para que generen las columnas dinámicamente utilizando estas nomenclaturas, así se consigue crear nuevas tablas en caso de eliminarlas en la base de datos “ULL”.

- **Nomenclaturas de cada tabla creada para cada página:**

Atributo	Descripción
id	Identificador auto-generado para cada elemento de la tabla.
name	Campo de 255 caracteres donde guardar el nombre.
address	Campo de 255 caracteres donde guardar la dirección.
municipality	Campo de 255 caracteres donde guardar el municipio.
postalcode	Campo de número entero donde guardar el código postal.
latitude	Campo numérico sobre la coordenada latitud que permite un máximo de 1000 decimales.
longitude	Campo numérico sobre la coordenada longitud que permite un máximo de 1000 decimales.
type	Campo de 255 caracteres donde guardar el tipo.
place	Campo de 255 caracteres donde guardar el sitio.
description	Campo de texto ilimitado donde guardar la descripción del sitio.
accessibility	Campo de 255 caracteres donde guardar la accesibilidad del sitio.
destination	Campo de texto ilimitado donde guardar la destinación del sitio.
dateupdated	Campo de 255 caracteres donde guardar la fecha de la última modificación realizada.
email	Campo de 255 caracteres donde guardar el correo electrónico.
phone	Campo de 255 caracteres donde guardar el número de teléfono.
fax	Campo de 255 caracteres donde guardar el número de fax.
image	Campo de 255 caracteres donde guardar la url de la imagen.
length	Campo de 255 caracteres donde guardar el tamaño longitud del sitio.
width	Campo de 255 caracteres donde guardar el tamaño anchura del sitio.
services	Campo de texto ilimitado donde guardar los servicios ofrecidos.
rating	Campo decimal donde guardar la puntuación total de cada elemento.

Tabla 3.1.: Nomenclatura de los lugares en la base de datos.

- **Nomenclaturas de cada tabla votación creada para cada página:**

Atributo	Descripción
id	Identificador auto-generado para cada elemento votado.
rating	Campo entero donde guardar el número de veces votado el elemento.
nombrepagina_id	Identificador obtenido del elemento votado.

Tabla 3.2.: Nomenclatura de las votaciones en la base de datos.

3.3.6. Las clases “LocationServiceImpl” y “LocationServiceDAOImpl”

Estas clases son las más usadas y se utilizan para desarrollar acciones muy comunes evitando así repetir en exceso código de programación. En el caso de la Clase “LocationServiceImpl” se encarga de realizar todos los pasos necesarios para leer cada fichero CSV desde la url “http://opendatacanarias.es” y actualizar la información en el portal web. La clase tiene sus propios métodos para corregir el formato de cada campo, obtener las coordenadas faltantes y de almacenar toda la información en la base de datos con nombre “ULL” de PostgreSQL. La clase “LocationServiceDAOImpl” entre otras cosas se encarga de interactuar con la base de datos, como almacenar y modificar la información como en el caso de realizar una votación.

3.3.7. Uso de los microdatos para la validación RichSnippet

Tras realizar un estudio sobre el tipo de ontologías y anotaciones a usar y explicado en el capítulo **1.1 Descripción del contexto**, se han usado los microdatos para etiquetar o marcar cada atributo a través del tipo de esquema al que pertenece. Esto permite una mejor presentación de los resultados en los buscadores como el formato RichSnippet de Google, que muestra los detalles de la publicación.

A continuación se muestra un ejemplo de un resultado en RichSnippet:



Cabe mencionar, que los microdatos es lo que recomiendan los grandes buscadores por estar desarrollado y respaldado por los mismos y ofrecer toda la documentación desde la web **Schema.org**. Los microdatos, en comparación con otros tipo de metadatos, permiten una mayor comprensión del contenido etiquetado y tener una estructura de los datos más flexible. Éstos forman parte del estándar de HTML5, con lo cual, también se evita tener que definir el tipo de documento XHTML como sucede con el lenguaje RDFa.

En este portal web, cada página pertenece a un tipo de esquema definido como **itemscope** por ser el padre de los subitems definidos como **itemprop**.

En el **itemscope** es obligatorio definir el tipo de esquema a través de **itemtype**. En caso de necesitar otro esquema de atributos dentro del itemscope padre y no perder la herencia, al hijo itemscope hay que añadirle la propiedad **itemprop** ya que sino queda definido como un nuevo esquema.

A continuación, muestro un ejemplo de herencia:

Para el caso de la página de alojamientos se a definido el esquema **itemtype** *LodgingBusiness* y dentro del mismo, se encuentra otro esquema hijo **itemtype** *PostalAddress* con el atributo **itemprop** *address* por ser un atributo perteneciente al esquema *LodgingBusiness*, se ha podido definir un esquema perteneciente a otro.

El código del ejemplo es el siguiente:

```
<div itemscope itemtype="http://schema.org/LodgingBusiness">
  <div itemprop="address" itemscope itemtype="http://schema.org/PostalAddress">
    <span itemprop="streetAddress" wicket:id="address" /><br />
    <span itemprop="addressLocality" wicket:id="municipality" />
    <span wicket:id="postalcodeTag" />
    <span itemprop="postalCode" wicket:id="postalCode" />
    <meta itemprop="addressRegion" content="Spain" /><br />
  </div>
</div>
```

Las páginas html del portal web, han sido definidas con estos esquemas:

<http://schema.org/LodgingBusiness>
<http://schema.org/FoodEstablishment>
<http://schema.org/TouristInformationCenter>
<http://schema.org/TouristAttraction>
<http://schema.org/Museum>
<http://schema.org/Beach>
<http://schema.org/Volcano>

3.3.8. Análisis del código del proyecto con SonarQube

Como herramienta de análisis de código Java, se ha utilizado el plugin de Eclipse “SonarQube Java” instalado desde el enlace:

<http://dist.sonar-ide.codehaus.org/eclipse/>

Para poder sincronizar el proyecto con el servidor de SonarQube, se ha descargado la versión 4.3.2 y su ejecutable “SonarQube Runner” con versión 2.4.

Luego, dentro del directorio del proyecto se ha creado el fichero “sonar-project.properties” con la siguiente información para analizar el código del proyecto en el directorio “/src/main/java”:

```
# Required metadata
sonar.projectKey=ull
sonar.projectName=CheckTenerife project analyzed with the SonarQube Runner
sonar.projectVersion=1.0
# Comma-separated paths to directories with sources (required)
sonar.sources=src/main/java
# Language
sonar.language=java
# Encoding of the source files
sonar.sourceEncoding=UTF-8
```

Desde el terminal se ha ejecutado el servicio servidor mediante:

```
Dauids-MacBook-Pro:ull david$ /Users/david/Documents/workspace_ull/ull/sonarqube-4.3.2/bin/macosx-universal-64/sonar.sh start
Starting SonarQube...
Started SonarQube.
```

Luego, tras comprobar que desde el enlace “http://localhost:9000” SonarQube funciona (Figura 3.3) se ha ejecutado dentro del directorio del proyecto otro servicio, el de ejecución, para realizar las comprobaciones del código:

```
Dauids-MacBook-Pro:ull david$ cd /Users/david/Documents/workspace_ull/ull
Dauids-MacBook-Pro:ull david$ /Users/david/Documents/workspace_ull/ull/sonar-runner-2.4/bin/sonar-runner
```

A continuación se muestra una captura de los informes realizados del proyecto por SonarQube.

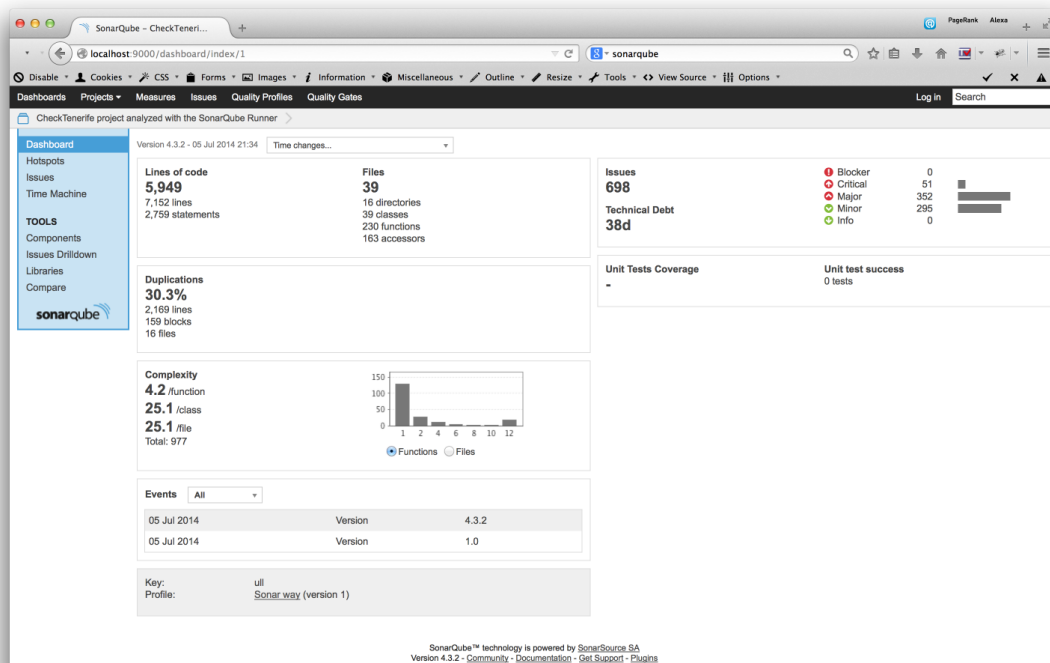


Figura 3.3.: Herramienta SonarQube para analizar código Java

Como resultado, podemos ver que se ha escrito un total de 5.949 líneas de código en las clases analizadas dentro del directorio “/src/main/java” del proyecto.

Hay un total de 16 directorios, 39 clases, 230 funciones y 163 declaraciones. En cuanto a la métrica del código, se llega a observar una complejidad ciclomática media en las funciones de 4.2, por lo tanto no existe ningún riesgo al estar por debajo del rango de riesgo comprendido entre 11 y 50.

En cambio, dentro de las clases hay una complejidad ciclomática media de 25.1 y esto significa un riesgo alto debido a que la clase “LocationServiceImpl” incorpora los métodos más costosos como el “readFile()” que se encarga de leer toda la información desde las url de OpenData Canarias, y el método “getGoogleCoordinates()” que realiza las llamadas para obtener las coordenadas guardándolas en ficheros de texto.

3.3.9. Tiempos ahorrados en la codificación con jRebel

Para no tener que reiniciar continuamente el servidor, después de aplicar los cambios sobre el código Java se ha usado el plugin “jRebel” con licencia de uso no comercial para poder compilar los cambios en ejecución en el servidor de Tomcat.

A continuación muestro una imagen de mi cuenta sobre el tiempo ahorrado:

En los últimos 30 días: 126 horas y 51 minutos.

Total: 28.013.545 despliegues ahorrados al servidor de Apache Tomcat.

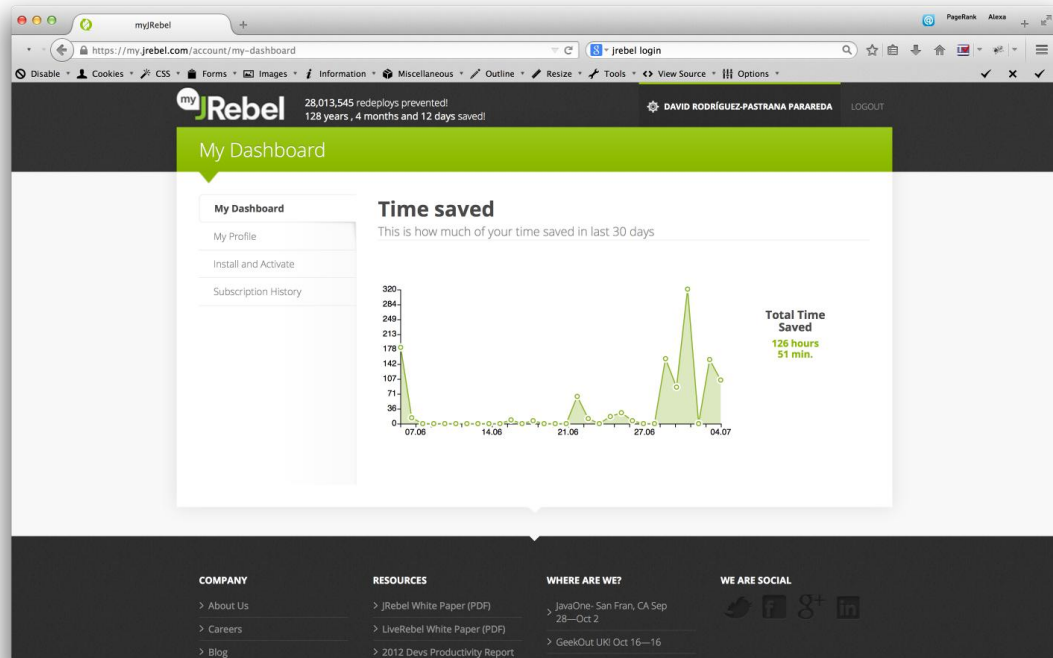


Figura 3.4.: Plugin de Eclipse jRebel en ahorro de compilación.

3.3.10. Problemas en el desarrollo del proyecto

En este apartado se detallan los diferentes problemas encontrados durante el desarrollo y configuración del proyecto, al ser la primera vez que se pone en práctica la configuración real de un proyecto Java sobre una plataforma web a través de diferentes frameworks. Se ha tenido que dedicar un esfuerzo mayor en aprenderlos siendo el framework más costoso de aprender el de Wicket, recurriendo a diferentes manuales para conocer los métodos usados en su librería.

3.3.10.1. Problema en la configuración del proyecto

La configuración de las librerías en Maven y la creación de nuevos ficheros ha conllevado muchas complicaciones al no haber en internet suficiente información para ver otros ejemplos de configuración.

La configuración del gestor Maven y del framework de Wicket no fue tan costosa como se pudo comprobar en el capítulo 3.2, y la dificultad estuvo en configurar los frameworks de Spring y Hibernate, teniendo que emplear mucho tiempo en crear la configuración y cargar las nuevas librerías sobre la persistencia en la BD.

3.3.10.2. Problemas con la obtención de Coordenadas

En el fichero de restauración de OpenData Canarias aparecía en un principio 12.000 establecimientos, del cual sólo un 1% tenía coordenadas.

Para ello, se ha implementado un algoritmo que es capaz de guardar las coordenadas en ficheros de texto añadiendo tres columnas.

La primera columna para indicar la fila del fichero, la segunda para saber si se ha encontrado la coordenada y marcada como “true” o “false” para no tener que buscar de nuevo, y la tercera columna para añadir la coordenada de latitud y longitud en caso de haberse encontrado. El fichero de texto, es comprobado varias veces hasta alcanzar el 95% de coordenadas encontradas.

La mayor complicación estuvo en la limitación que ofrece Google Maps para el uso de la API por no tener contratado un servicio Premium. La respuesta que ofrece la API tras alcanzar las 2.000 solicitudes es “OVER_QUERY_LIMIT” cada 24 horas, cuando era necesario realizar unas 11.000 llamadas.

El problema se pudo resolver haciendo un pequeño *hack*. Se llamó a la función “getGoogleCoordinates()” cada vez que se realizaba un cambio de IP en el ordenador para así obtener las 11.000 coordenadas encontradas.

Un ejemplo de cómo queda el fichero de texto tras encontrar las coordenadas es el siguiente. Las coordenadas con mayor número de decimales son las encontradas y las demás las que se pudieron leer.

```
1, true, 28.48967742919922,-16.318952560424805
2, true, 28.211740493774414,-16.423879623413086
3, true, 28.1673135,-16.5031708
4, true, 28.09684944152832,-16.61952018737793
5, true, 28.11540412902832,-16.564077377319336
6, true, 28.4747371673584,-16.253124237060547
7, true, 28.125757217407227,-16.576845169067383
8, true, 28.454376220703125,-16.44013023376465
9, true, 28.1778793,-16.479779
10, true, 28.3079186,-16.4052541
11, true, 28.390085220336914,-16.643831253051758
12, true, 28.417173385620117,-16.54897689819336
13, true, 28.532930374145508,-16.3693904876709
```

3.3.10.3. Problemas con la obtención de imágenes

La url de las imágenes leídas desde los ficheros CSV tienen el siguiente formato:

```
../PublishingImages/hotel_rural_san_miguel-229509-1158727_3_b.jpg
```

Se ha buscado el origen del enlace y se ha podido comprobar que pertenece al siguiente servidor privado:

http://infoten.es/fichas/PublishingImages/hotel_rural_san_miguel-229509-1158727_3_b.jpg

Por tanto, se pide una identificación de usuario y contraseña para poder visualizar la imagen. Este problema se espera solucionar cuando OpenData Canarias pueda ofrecer alguna solución.

3.3.10.4. Problemas con los microdatos

Añadir microdatos en las votaciones no ha sido posible debido a que no se permite añadir múltiples esquemas en una misma página. Desde la herramienta de test de Google RichSnippet se detallaba el siguiente error:

Error: multiple items voted on the same page.

La solución a este problema consiste en añadir la votación en distintas páginas url para cada establecimiento, pero por el tipo de desarrollo implementado, se muestran 50 establecimientos por página y por tanto, no ha sido posible especificar las etiquetas a través del esquema: <http://schema.org/AggregateRating>

3.3.10.5. Problemas con la actualización de los Dataset

En OpenData Canarias, el día 18 de junio de 2013 se actualizaron los ficheros CSV de donde se obtiene toda la información y se publicaron nuevos ficheros referidos al turismo de Tenerife.

Como solución, hubo que actualizar el formato de lectura de los ficheros de alojamientos y restauración. También, hubo que añadir nuevos campos con información más detallada de cada establecimiento, cambiar el formato de lectura, añadir nuevas coordenadas y cambiar el enlace de donde se recoge la información.

Tras realizar los cambios, se obtuvo como ventaja una estructura y orden homogéneo para todos los ficheros leídos, siguiendo un mismo patrón que permitió añadir los nuevos ficheros publicados de forma fácil.

La nueva información recogida es sobre las playas, monumentos naturales y recursos volcánicos de la isla. Además, también se añadieron noticias RSS para mostrar los eventos turísticos y de los museos.

3.3.10.6. Problemas con las librerías de jQuery y la API de Google Maps

Se encontraron dificultades en desarrollar código jQuery usando librerías específicas para etiquetar los marcadores del mapa a través de los microdatos. Al no usar los métodos estándar de JavaScript, se pudo empezar con un ejemplo sencillo que ofrece IBM desde el siguiente enlace:

<http://www.ibm.com/developerworks/web/library/x-html5microdata1/>

En él se explica cómo usar la librería de jQuery con la API de Google Maps y los microdatos con la última versión de la API, la 3.0.

Para ello, se descargó el paquete “jquery-ui-map-3.0-rc.zip” desde el enlace:

<https://code.google.com/p/jquery-ui-map/downloads/list>

Tras añadir las librerías necesarias, la mayor complicación fue aprender el uso de cada una y que explico a continuación.

- **Librería “jquery-ui-map.js”:** Permite añadir los marcadores en el mapa mediante jQuery. Al no usar los métodos estándar de JavaScript para llamar a la API de Google Maps y no haber muchos ejemplos en internet, se tuvo que usar en varias ocasiones el depurador FireBug de Firefox. A continuación, muestro la configuración del mapa y el método más usado “markerProperties” para ver cómo se inicializan los marcadores en el mapa:

```
map = $('#map_canvas');
map.gmap({
  'panControl': false,
  'zoomControl': false,
  'scaleControl': false,
  'mapTypeControl': true,
  'mapTypeControlOptions': { position: google.maps.ControlPosition.TOP_CENTER },
  'streetViewControl': false,
  'overviewMapControl': false
});
$(window).resize(function() {
  map.gmap('refresh');
});
$(window).trigger('resize');
```

```
function markerProperties(icon,result,item,index) {
  var name = result.name;
  var lat = result.geo[0].latitude;
  var lng = result.geo[0].longitude;
  var latLng = new google.maps.LatLng(lat, lng);
  var markerOptions = {
    'bounds':true,
    'position': latLng,
```

```

        'icon': icon
    };
    var markerDetails = "";
    markerDetails += '<div class="markerInfo">';
    markerDetails += '<h2>'+name+'</h2>';
    markerDetails += '<a onclick="!hideDirectionTable ? displayDirectionTable() : false;';
    markerDetails += '</div>';
    map.gmap('addMarker', markerOptions, function(map, marker) {
        $(item).click(function(event) {
            if($(event.target).is('a') ){
                $(item + ' .findClickable').unbind("click");
            }
            $(marker).triggerEvent('click');
            map.panTo(marker.position);
            return false;
        });
        }).click( function() {
        if(last_checked != "") {
            $(last_checked).removeClass('markerClicked');
        }
        map.gmap('openInfoWindow', { 'content': markerDetails }, this);
        map.gmap('displayStreetView', 'streetview{0}'.replace('{0}', index), {
            'position': $(this).get(0).getPosition(), 'pov': { 'heading': 34, 'pitch': 10,
            'zoom': 1 }
        });
        destination = this.position;
        map.gmap('search', { 'location': destination }, function(results, status) {
            if (status === 'OK') {
                $('#to').val(results[0].formatted_address);
                $('#resetInputTo').show();
            }
        });
        $(item).addClass('markerClicked');
        checkIfInView($(item));
        last_checked = $(item);
    });
}

```

- **Librería “jquery.ui.map.microdata.js”:** Permite añadir microdatos a los marcadores del mapa a través de las etiquetas “itemprop” e “itemscope” definidas en el código html. Un ejemplo sencillo sería el siguiente:

```

map.gmap('microdata', 'http://schema.org/LodgingBusiness', function(result, item, index) {

    var type = result.alternateName;
    var icon = null;
    if(type == null) {
        icon = { 'url' : '/images/markers/lodging/lodging_0star.png' };
    }
    switch(type) {
        case 'Hotel':
            category = result.makesOffer[0].category;
            switch(category) {
                case '1 estrella': icon = { 'url' : '/images/markers/lodging/hotel_1star.png' };
            }
        }
    }
}

```

```
        break;
    case '2 estrellas': icon = {'url' : '/images/markers/lodging/hotel_2stars.png'};
        break;
    case '3 estrellas': icon = {'url' : '/images/markers/lodging/hotel_3stars.png'};
        break;
    case '4 estrellas': icon = {'url' : '/images/markers/lodging/hotel_4stars.png'};
        break;
    case '5 estrellas': icon = {'url' : '/images/markers/lodging/hotel_5stars.png'};
        break;
    case '5 estrellas Lujo': icon = {'url' : '/images/markers/lodging/hotel_5stars.png'};
        break;
    default: icon = {'url' : '/images/markers/lodging/hotel_0star.png'};
    }
});
```

- **Librería “jquery.ui.map.extensions.js”**: Con esta librería se consigue obtener la ubicación actual del usuario, la dirección de destino y ver el modo StreetView. El código es el siguiente:

Para ver el modo StreetView:

```
map.gmap('displayStreetView', 'streetview{0}'.replace('{0}', index), {
    'position': $(this).get(0).getPosition(), 'pov': {'heading': 34, 'pitch': 10, 'zoom': 1 }
});
```

Para obtener la ubicación actual:

```
function initGeolocation() {
    $('#submitFindPosition').click(function() {
        map.gmap('getCurrentPosition', function(position, status) {
            if (status === 'OK') {
                latlng = new google.maps.LatLng(position.coords.latitude,
                position.coords.longitude);
                map.gmap('get', 'map').panTo(latlng);
                map.gmap('search', { 'location': latlng }, function(results, status) {
                    if (status === 'OK') {
                        $('#from').val(results[0].formatted_address);
                        $('#currentLatitude').empty();
                        $('#currentLongitude').empty();
                        $('#currentLatitude').text(results[0].geometry.location.lat());
                        $('#currentLongitude').text(results[0].geometry.location.lng());
                        initMarkers();
                        $('#resetInputFrom').show();
                    }
                });
            } else {
                alert('Unable to get current position');
            }
        });
    });
}
```

Para obtener la dirección de destino:

```
function initDirection() {
$('#submitFindDirection').click(function() {
    var selectedMode = $("#mode").val();
    var displayDirectionsOptions = {
        'origin': $('#from').val(),
        'destination': $('#to').val(),
        'avoidHighways': false,
        'avoidTolls': false,
        'unitSystem': google.maps.UnitSystem.METRIC,
        'travelMode': google.maps.DirectionsTravelMode[selectedMode]
    };
    var panel = {
        'panel': document.getElementById('directions')
    };
    map.gmap('displayDirections', displayDirectionsOptions, panel,
function(success, response) {
    suppressMarkers: true;
    if (success) {
        if(response == 'ZERO_RESULTS') {
            alert('Error. Introduce una dirección más exacta. ');
            $('#directions').text(' ');
            findDirection = false;
        } else {
            findDirection = true;
        }
    } else {
        findDirection = false;
    }
    $('#directions').show();
});
return false;
});
}
```


4. Resultados

En este capítulo se detallan los resultados ofrecidos en el portal, mostrando una captura de pantalla de cada página y de las pruebas realizadas con éxito tras incorporar los microdatos. Las capturas del portal web se han realizado desde el enlace “<http://www.checktenerife.com>”, y las capturas realizadas con la herramienta RichSnippet de Google, han sido comprobadas desde: <http://www.google.com/webmasters/tools/richsnippets>

4.1. Página de inicio

En esta página se puede observar un slider animado con 4 imágenes de Tenerife y unos títulos dinámicos referidos al portal web. Las imágenes y los títulos se adaptan al tamaño del dispositivo para ofrecer una presentación atractiva de los resultados mostrados.

También, se han añadido varias funciones interesantes para subir el SEO de la web en los buscadores:

- Se ha detallado el tipo de proyecto realizado.
- Panel para seguir las novedades del portal web en las redes sociales.
- Panel para recomendar el portal en las redes sociales, enviar un correo e imprimir con formato cualquier página (esquina inferior derecha).
- Panel para contactar sobre algún problema desde cualquier página (esquina inferior izquierda).
- **Enlace:** <http://www.checktenerife.com/portada>

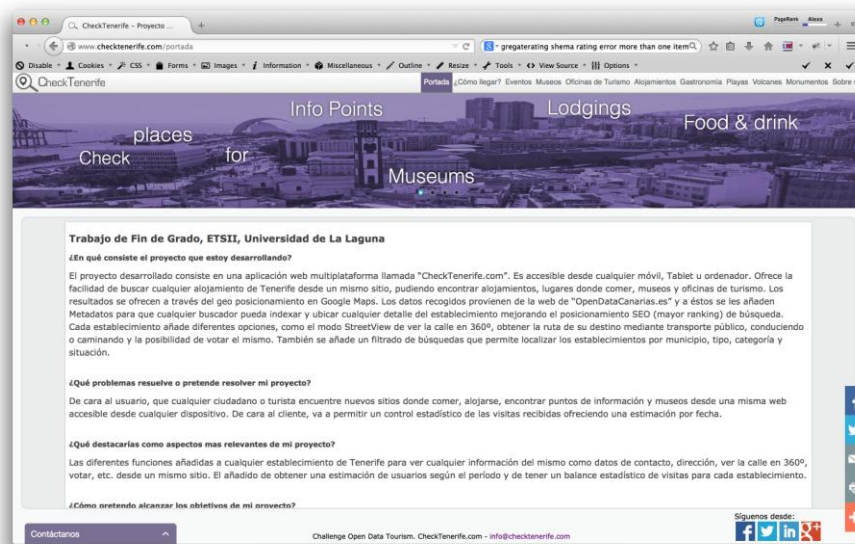


Figura 4.1.: Captura página de inicio

Validación página de inicio

Datos estructurados extraídos

Item	
type:	http://schema.org/webpage
property:	
image:	http://www.checktenerife.com/images/logo.png
significantlink:	Portada
significantlink:	¿Cómo llegar?
significantlink:	Eventos
significantlink:	Museos Eventos
significantlink:	Oficinas de Turismo
significantlink:	Alojamientos
significantlink:	Gastronomía
significantlink:	Playas
significantlink:	Volcanes
significantlink:	Monumentos
significantlink:	Sobre mí

Figura 4.2: Captura validación página de inicio

4.2. Página de cómo llegar

Aprovechando la funcionalidad de obtener la dirección de destino para cada establecimiento, se añadió esta página para poder escribir la dirección de inicio y de destino, escoger el medio de transporte y obtener la ruta disponible gracias a la API de Google Map places. En la dirección de inicio se puede también añadir nuestra ubicación actual y los marcadores mostrados en el mapa se pueden mover para auto generar la dirección de destino. En caso de seleccionar el medio de transporte público, se nos ofrecerá la hora del siguiente tranvía o bus más próximo.

- **Enlace:** <http://www.checktenerife.com/desino>

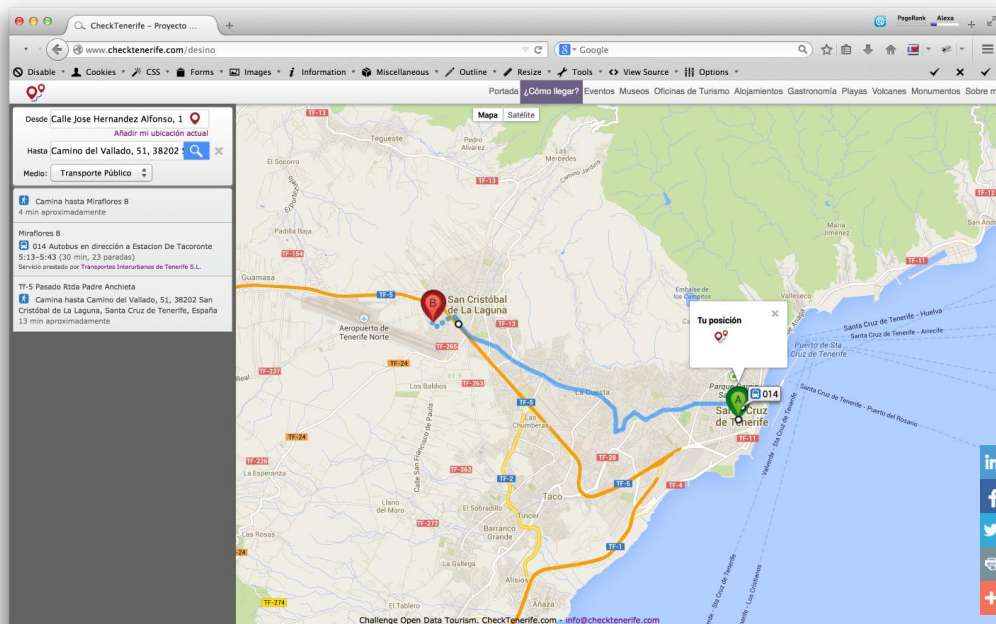


Figura 4.3.: Captura página de destino

4.3. Página de eventos turísticos

En la siguiente página se muestran los últimos 10 eventos turísticos actualizados desde la web “www.webtenerife.com”. Para ver mayor información sobre el evento hay que dar click en el título y entonces se abrirá una nueva pestaña del evento original al estar limitado a mostrar un máximo de dos líneas.

- **Enlace:** <http://www.checktenerife.com/eventos>

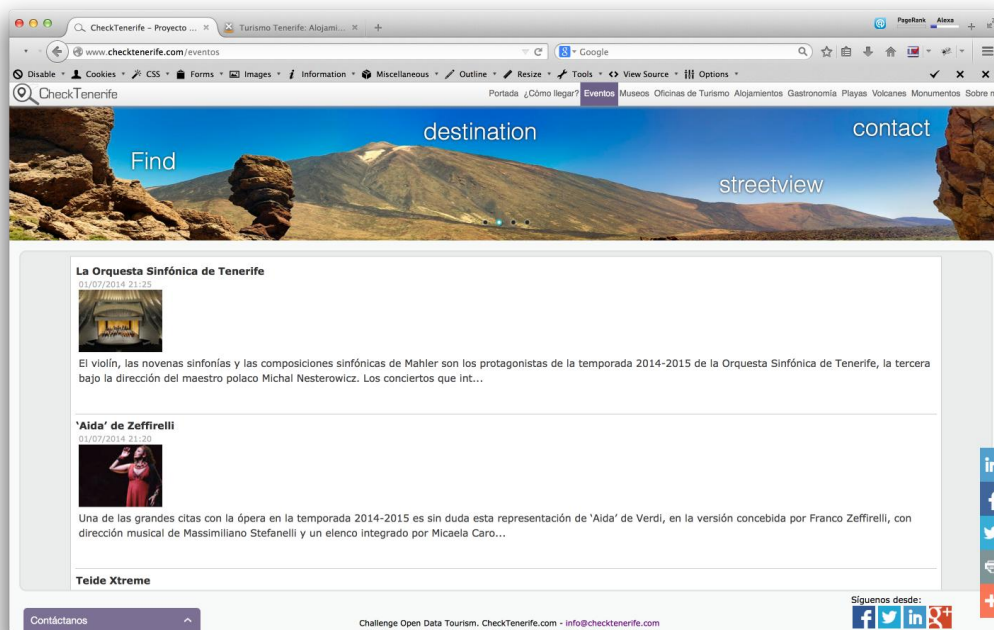


Figura 4.4.: Captura página de eventos turísticos

4.4. Página de museos

Esta página muestra el geoposicionamiento de los museos de Tenerife y la información de cada uno siendo la dirección, teléfono, web y buscar en Google. Se puede ver la valoración de otros usuarios, añadir el voto, obtener la dirección de destino, ver el modo StreetView y filtrar las búsquedas por municipio o buscar por el nombre del museo.

- Enlace: <http://www.checktenerife.com/museos>

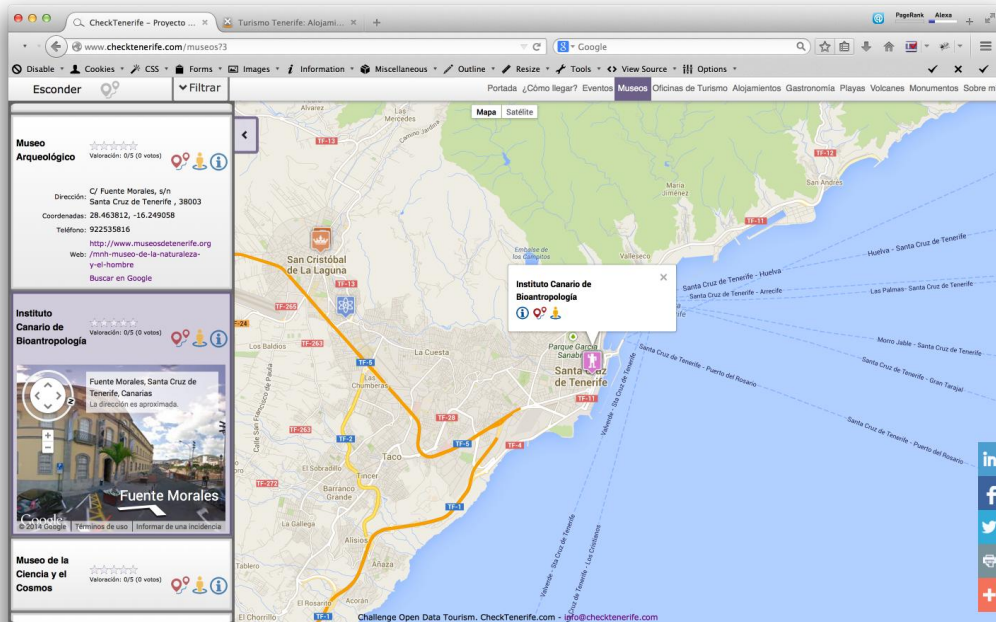




Figura 4.5.: Captura página de museos

Validación página de museos



+check ☰ 🔔 + 

Herramientas para webmasters de Google

Página principal

Herramienta de pruebas de datos estructurados

Asistente para el marcado de datos estructurados

Ayuda relacionada con:
[Solución de problemas](#)
[Centro de ayuda](#)

Herramienta de pruebas de datos estructurados

URL

HTML

VISTA PREVIA

Ejemplos ▾

Selecciona la pestaña HTML para ver el HTML recuperado y prueba a modificarlo.

Resultados de búsqueda de Google

Búsqueda personalizada de Google

Vista previa

CheckTenerife - Proyecto de Información turística de Tenerife

www.checktenerife.com/museos

El fragmento de la página aparecerá aquí. El motivo por el que no podemos mostrar texto de tu página web es que el texto mostrado depende de la consulta realizada por el usuario.

Resultado de las pruebas de autoría

La página no contiene ningún marcado de autoría. [Más información](#)

Verificación por correo electrónico de la autoría

Introduce un perfil de Google+ para comprobar si el autor ha **verificado correctamente una dirección de correo electrónico** en el dominio www.checktenerife.com para establecer la autoría de esta página web. [Más información](#)

Verificar autoría

Editor

La página no contiene marcado de editor. [Más información](#)

Datos estructurados extraídos

Item

type: http://schema.org/webpage

property:

- image: http://www.checktenerife.com/images/logo.png
- significantlink: Portada
- significantlink: ¿Cómo llegar?
- significantlink: Eventos
- significantlink: Museos Eventos
- significantlink: Oficinas de Turismo
- significantlink: Alojamientos
- significantlink: Gastronomía
- significantlink: Playas
- significantlink: Volcanes
- significantlink: Monumentos
- significantlink: Sobre mí

Item

type: http://schema.org/museum

property:

- name: Museos de Tenerife (Sede Administrativa)
- description: 8
- address: *Item 1*

geo:	Item 2
telephone:	922209320
url:	http://www.museosdetenerife.org
url:	Buscar en Google

Item 1	
type:	http://schema.org/postaladdress
property:	
streetaddress:	C/ Fuente Morales, s/n
addresslocality:	Santa Cruz de Tenerife
postalcode:	38003
addressregion:	Spain

Item 2	
type:	http://schema.org/geocoordinates
property:	
latitude:	28.463812
longitude:	-16.249058

Item	
type:	http://schema.org/museum
property:	
name:	Museo de la Naturaleza y el Hombre
description:	1
address:	Item 3
geo:	Item 4
telephone:	922535816
url:	http://www.museosdetenerife.org/mnh-museo-de-la-naturaleza-y-el-hombre
url:	Buscar en Google

Item 3	
type:	http://schema.org/postaladdress
property:	
streetaddress:	C/ Fuente Morales, s/n
addresslocality:	Santa Cruz de Tenerife
postalcode:	38003
addressregion:	Spain

Item 4	
type:	http://schema.org/geocoordinates
property:	
latitude:	28.463812
longitude:	-16.249058

Item	
type:	http://schema.org/museum
property:	
name:	Museo de Ciencias Naturales
description:	2
address:	Item 5
geo:	Item 6
telephone:	922535816
url:	http://www.museosdetenerife.org/mnh-museo-de-la-naturaleza-y-el-hombre
url:	Buscar en Google

Figura 4.6.: Captura validación página de museos

4.5. Página de eventos en museos

Al marcar museos, aparece un submenú debajo del mismo para poder ver los últimos 10 eventos en los museos de Tenerife.

Tiene como límite mostrar los dos primeros párrafos del evento original que proviene desde la web “www.webtenerife.com”.

- **Enlace:** www.checktenerife.com/noticias

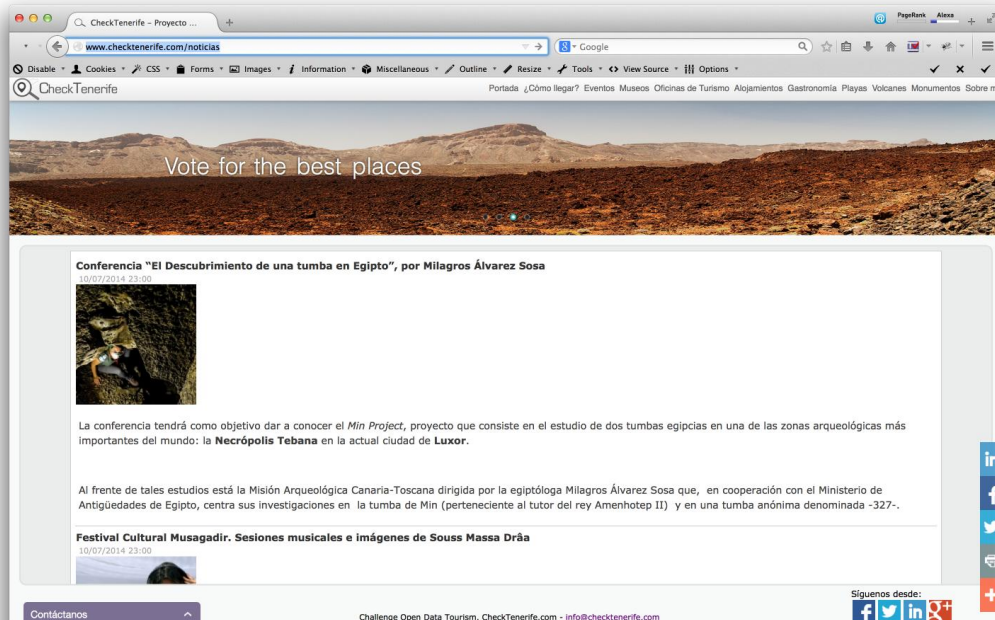


Figura 4.7.: Captura página de eventos en museos

4.6. Página de oficinas de turismo

Esta página muestra el geoposicionamiento de las oficinas de turismo de Tenerife y la información de cada oficina siendo la dirección, horario de apertura, teléfono, fax, email, web y buscar en Google. Se puede ver la valoración de otros usuarios, añadir el voto, puntuar, obtener la dirección de destino, ver el modo StreetView y filtrar las búsquedas por municipio o buscar por el nombre de la oficina.

- **Enlace:** <http://www.checktenerife.com/informacion>

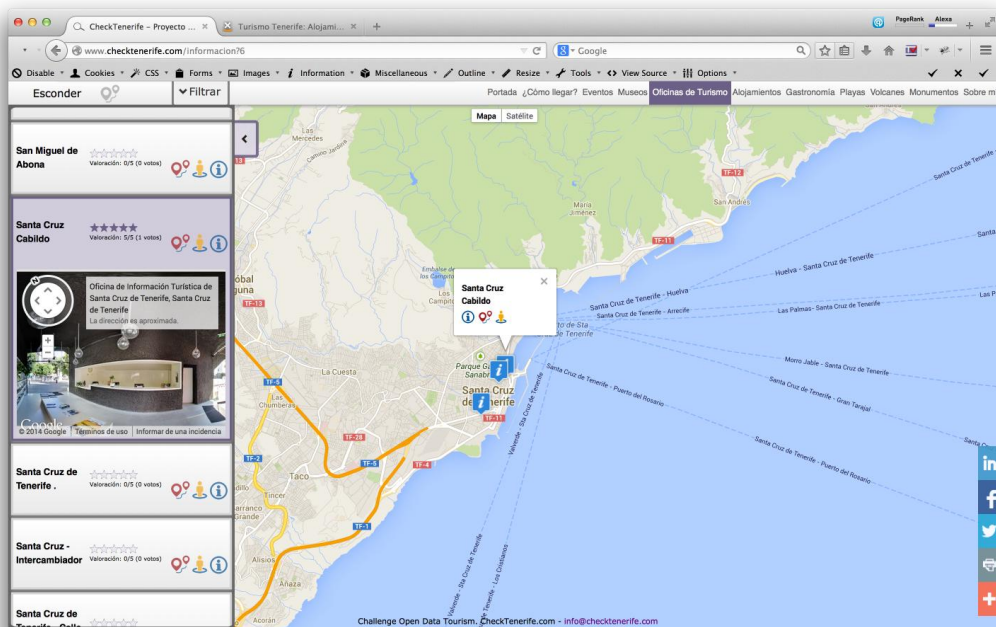




Figura 4.8.: Captura página de oficinas de turismo

Validación página de oficinas de turismo



+check ☰ 🔔 + 

Herramientas para webmasters de Google

Página principal

Herramienta de pruebas de datos estructurados

Asistente para el marcado de datos estructurados

Ayuda relacionada con:
Solución de problemas
Centro de ayuda

Herramienta de pruebas de datos estructurados

URL

HTML

VISTA PREVIA
Ejemplos ▾

Selecciona la pestaña HTML para ver el HTML recuperado y prueba a modificarlo.

Resultados de búsqueda de Google

Búsqueda personalizada de Google

Vista previa

CheckTenerife - Proyecto de Información turística de Tenerife

www.checktenerife.com/informacion

El fragmento de la página aparecerá aquí. El motivo por el que no podemos mostrar texto de tu página web es que el texto mostrado depende de la consulta realizada por el usuario.

Resultado de las pruebas de autoría

La página no contiene ningún marcado de autoría. [Más información](#)

Verificación por correo electrónico de la autoría

Introduce un perfil de Google+ para comprobar si el autor ha verificado correctamente una dirección de correo electrónico en el dominio www.checktenerife.com para establecer la autoría de esta página web. [Más información](#)

Verificar autoría

Editor

La página no contiene marcado de editor. [Más información](#)

Datos estructurados extraídos

Item

type: http://schema.org/webpage

property:

- image: http://www.checktenerife.com/images/logo.png
- significantlink: Portada
- significantlink: ¿Cómo llegar?
- significantlink: Eventos
- significantlink: Museos Eventos
- significantlink: Oficinas de Turismo
- significantlink: Alojamientos
- significantlink: Gastronomía
- significantlink: Playas
- significantlink: Volcanes
- significantlink: Monumentos
- significantlink: Sobre mí

Item

type: http://schema.org/touristinformationcenter

property:

- name: Centro de Visitantes Cruz del Carmen
- address: Item 1
- openinghours: Lun.- Dom.: 9:30 - 18:00
- geo: Item 2
- telephone: 922 63 35 76

Challenge of OpenData in Tenerife Tourism

email:	cvisitantes@cabtfe.es
uri:	Buscar en Google

Item 1	
type:	http://schema.org/postaladdress
property:	
streetaddress:	Ctra. Las Mercedes, Km.6
addresslocality:	San Cristóbal de la Laguna
postalcode:	38294
addressregion:	Spain

Item 2	
type:	http://schema.org/geocoordinates
property:	
latitude:	28.531297762237124
longitude:	-16.28023624420166

Item	
type:	http://schema.org/touristinformationcenter
property:	
name:	Arico
address:	<i>Item 3</i>
openinghours:	Lun-Vie. : 09:00 - 14:30
geo:	<i>Item 4</i>
telephone:	922 16 11 33
email:	turismo@ayuntamientodearico.com
uri:	http://www.ayuntamientodearico.com
uri:	Buscar en Google

Item 3	
type:	http://schema.org/postaladdress
property:	
streetaddress:	C/ Benítez De Lugo, 1
addresslocality:	Arico
postalcode:	38580
addressregion:	Spain

Item 4	
type:	http://schema.org/geocoordinates
property:	
latitude:	28.1666549
longitude:	-16.501374800000008

Item	
type:	http://schema.org/touristinformationcenter
property:	
name:	Candelaria OIT
address:	<i>Item 5</i>
openinghours:	De lunes a jueves 9:00 a 14:00 y de 15:00 a 17:00 Viernes y sábados de 9:00 a 13:00
geo:	<i>Item 6</i>
telephone:	922 03 22 30
faxnumber:	922 50 00 08
email:	oit@candelaria.es
uri:	http://www.candelaria.es
uri:	Buscar en Google

Figura 4.9.: Captura validación página de oficinas de turismo

4.7. Página de alojamientos

Esta página muestra el geoposicionamiento de las oficinas de turismo de Tenerife y la información de cada oficina siendo el tipo, categoría, situación, dirección, descripción, tipo de servicios, número de habitaciones, suites, y piscinas, distancia de la playa, puerto y parada bus, teléfono, fax, web, email y buscar en Google. Se puede ver la valoración de otros usuarios, puntuar, obtener la dirección de destino, ver el modo StreetView y filtrar las búsquedas por municipio, tipo y categoría o buscar por el nombre del alojamiento.

- **Enlace:** www.checktenerife.com/alojamientos

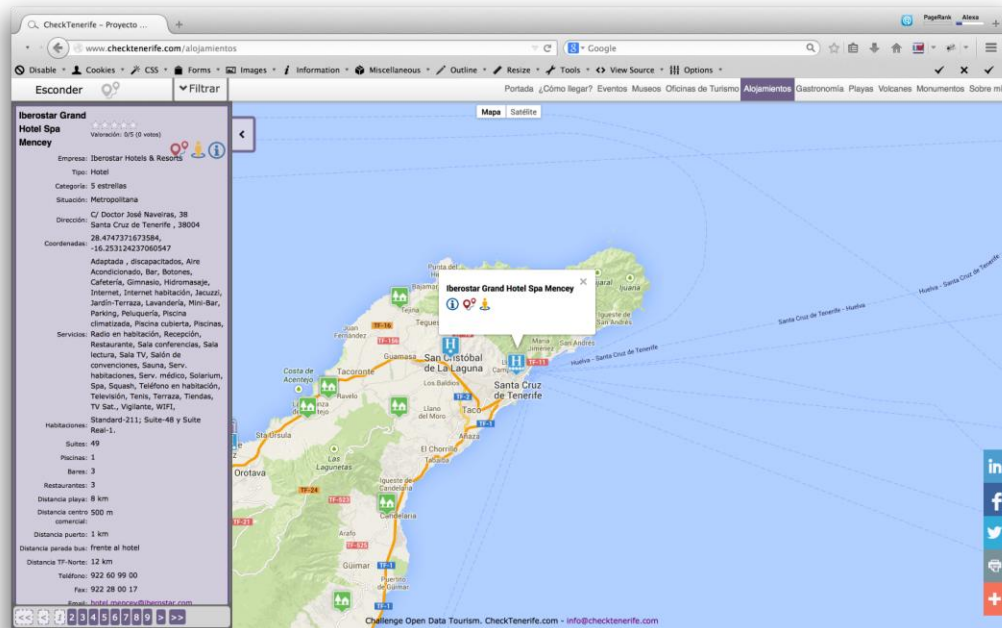




Figura 4.10.: Captura página de alojamientos

Validación página de alojamientos



+check ☰ 🔔 + 

Herramientas para webmasters de Google

[Página principal](#)

Herramienta de pruebas de datos estructurados

[Asistente para el marcado de datos estructurados](#)

Ayuda relacionada con:
[Solución de problemas](#)
[Centro de ayuda](#)

Herramienta de pruebas de datos estructurados

URL

HTML

VISTA PREVIA
Ejemplos ▾

Selecciona la pestaña HTML para ver el HTML recuperado y prueba a modificarlo.

Resultados de búsqueda de Google
Búsqueda personalizada de Google

Vista previa

CheckTenerife - Proyecto de Información turística de Tenerife

www.checktenerife.com/alojamientos

El fragmento de la página aparecerá aquí. El motivo por el que no podemos mostrar texto de tu página web es que el texto mostrado depende de la consulta realizada por el usuario.

Resultado de las pruebas de autoría

La página no contiene ningún marcado de autoría. [Más información](#)

Verificación por correo electrónico de la autoría

Introduce un perfil de Google+ para comprobar si el autor ha **verificado correctamente una dirección de correo electrónico** en el dominio www.checktenerife.com para establecer la autoría de esta página web. [Más información](#)

Verificar autoría

Editor

La página no contiene marcado de editor. [Más información](#)

Datos estructurados extraídos

Item

type: http://schema.org/webpage

property:

- image: http://www.checktenerife.com/images/logo.png
- significantlink: Portada
- significantlink: ¿Cómo llegar?
- significantlink: Eventos
- significantlink: Museos Eventos
- significantlink: Oficinas de Turismo
- significantlink: Alojamientos
- significantlink: Gastronomía
- significantlink: Playas
- significantlink: Volcanes
- significantlink: Monumentos
- significantlink: Sobre mí

Item

type: http://schema.org/lodgingbusiness

property:

- name: Agure
- alternatename: Hotel
- makesoffer: Item 1

description:	Metropolitana
address:	Item 2
geo:	Item 3
description:	Antiguo Palacio de los Marqueses de Torrehermosa y lleva 148 años como hotel, situado en el centro del conjunto histórico de San Cristóbal de La Laguna
description:	Cafetería, Internet, Peluquería, Recepción, Sala conferencias, Salón de convenciones, Teléfono en habitación, Televisión, Vigilante,
description:	22
description:	1
telephone:	922 25 94 90/922 25 19 67
faxnumber:	922 63 16 33
email:	reservas@hotelaguere.es
url:	http://www.hotelaguere.es
url:	Buscar en Google
description:	2013-11-27 12:08:41

Item 1	
type:	http://schema.org/offer
property:	
category:	1 estrella

Item 2	
type:	http://schema.org/postaladdress
property:	
streetaddress:	C/ Obispo Rey Redondo, 55
addresslocality:	San Cristóbal de la Laguna
postalcode:	38201
addressregion:	Spain

Item 3	
type:	http://schema.org/geocoordinates
property:	
latitude:	28.48967742919922
longitude:	-16.318952560424805

Item	
type:	http://schema.org/lodgingbusiness
property:	
name:	Finca Salamanca
alternatename:	Hotel rural
makesoffer:	Item 4
description:	Sur
address:	Item 5
geo:	Item 6
description:	Adaptada , discapacitados, Recepción, Televisión,
description:	20
description:	1
description:	
description:	3 Km
description:	3 Km
description:	En el hotel
description:	28 Km
telephone:	922 51 45 30
faxnumber:	922 51 40 61
email:	info@hotel-fincasalamanca.com
url:	http://www.hotel-fincasalamanca.com
url:	Buscar en Google

Figura 4.11.: Captura validación página de alojamientos

4.8. Página sobre la gastronomía

Esta página muestra el geoposicionamiento de las oficinas de turismo de Tenerife y la información de cada oficina siendo el tipo, la zona, dirección, horario de apertura, teléfono, fax, email, web y buscar en Google. Se puede ver la valoración de otros usuarios, puntuar, obtener la dirección de destino, ver el modo StreetView y filtrar las búsquedas por tipo, municipio o buscar por el nombre.

- **Enlace:** <http://www.checktenerife.com/gastronomia>

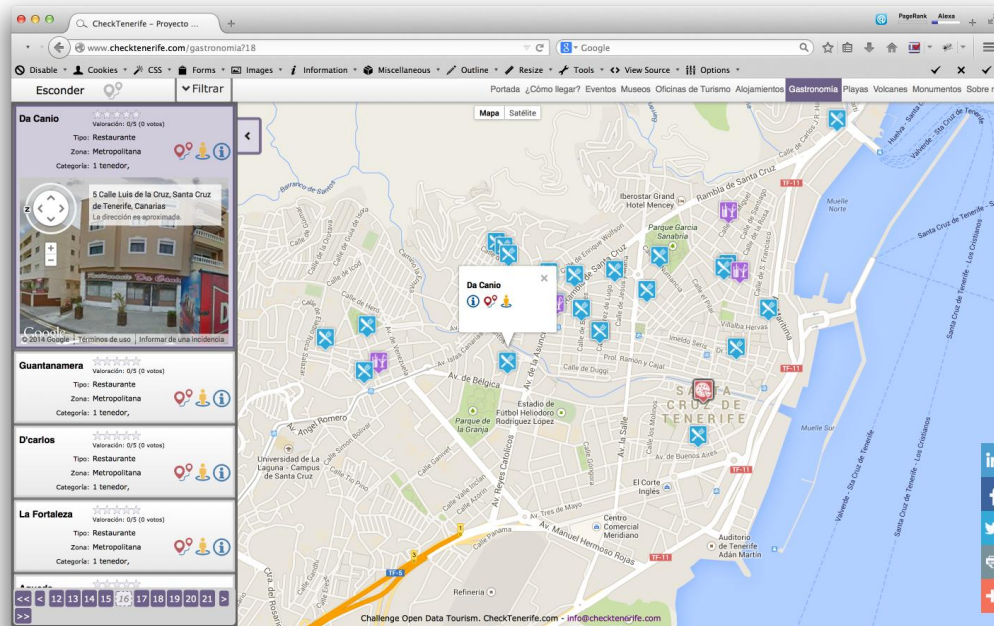




Figura 4.12.: Captura página de gastronomía

Validación página sobre la gastronomía



+check ⋮ 🔔 + 

Herramientas para webmasters de Google

[Página principal](#)

Herramienta de pruebas de datos estructurados

[Asistente para el marcado de datos estructurados](#)

Ayuda relacionada con:
[Solución de problemas](#)
[Centro de ayuda](#)

Herramienta de pruebas de datos estructurados

URL

HTML

VISTA PREVIA

Ejemplos ▾

Selecciona la pestaña HTML para ver el HTML recuperado y prueba a modificarlo.

Resultados de búsqueda de Google
Búsqueda personalizada de Google

Vista previa

CheckTenerife - Proyecto de Información turística de Tenerife

www.checktenerife.com/gastronomia

El fragmento de la página aparecerá aquí. El motivo por el que no podemos mostrar texto de tu página web es que el texto mostrado depende de la consulta realizada por el usuario.

Resultado de las pruebas de autoría

La página no contiene ningún marcado de autoría. [Más información](#)

Verificación por correo electrónico de la autoría

Introduce un perfil de Google+ para comprobar si el autor ha **verificado correctamente una dirección de correo electrónico** en el dominio www.checktenerife.com para establecer la autoría de esta página web. [Más información](#)

Verificar autoría

Editor

La página no contiene marcado de editor. [Más información](#)

Datos estructurados extraídos

Item

type: http://schema.org/webpage

property:

image: http://www.checktenerife.com/images/logo.png

significantlink: Portada

significantlink: ¿Cómo llegar?

significantlink: Eventos

significantlink: Museos Eventos

significantlink: Oficinas de Turismo

significantlink: Alojamientos

significantlink: Gastronomía

significantlink: Playas

significantlink: Volcanes

significantlink: Monumentos

significantlink: Sobre mí

Item

type: http://schema.org/foodestablishment

property:

name: Avencio

alternatename: Restaurante

description: Sur

<p>makesoffer: <i>Item 1</i> address: <i>Item 2</i> geo: <i>Item 3</i> telephone: 922178220 faxnumber: 922176040 uri: Buscar en Google description: 2011-05-31 13:35:49</p>
<p>Item 1 type: http://schema.org/offer</p>
<p>Item 2 type: http://schema.org/postaladdress property: streetaddress: Paseo Marcial García 2 - El Médano addresslocality: Granadilla de Abona postalcode: 38612 addressregion: Spain</p>
<p>Item 3 type: http://schema.org/geocoordinates property: latitude: 28.05986213684082 longitude: -16.5384521484375</p>
<p>Item type: http://schema.org/foodestablishment property: name: La Cueva alternatename: Bar description: Sur makesoffer: <i>Item 4</i> address: <i>Item 5</i> geo: <i>Item 6</i> openinghours: L a D 9.00-15.00. J a D 9.00-15.00 y 18.00-21.00 telephone: 922 52 91 93 uri: Buscar en Google description: 2012-01-11 09:27:09</p>
<p>Item 4 type: http://schema.org/offer</p>
<p>Item 5 type: http://schema.org/postaladdress property: streetaddress: C/ Edelmira Pérez Campos, 6. Puertito De Güímar addresslocality: Güímar postalcode: 38508 addressregion: Spain</p>
<p>Item 6 type: http://schema.org/geocoordinates property:</p>

Figura 4.13.: Captura validación página de gastronomía

4.9. Página de las playas

Esta página muestra el geoposicionamiento de las oficinas de turismo de Tenerife y la información de cada oficina siendo la dirección, descripción, tipo de servicios y accesibilidad, color arena, tipo de oleaje, longitud y anchura de la playa, teléfono, fax, web y buscar en Google. Se puede ver la valoración de otros usuarios, puntuar, obtener la dirección de destino, ver el modo StreetView y filtrar las búsquedas por municipio o buscar por el nombre de la playa.

- **Enlace:** <http://www.checktenerife.com/playas>

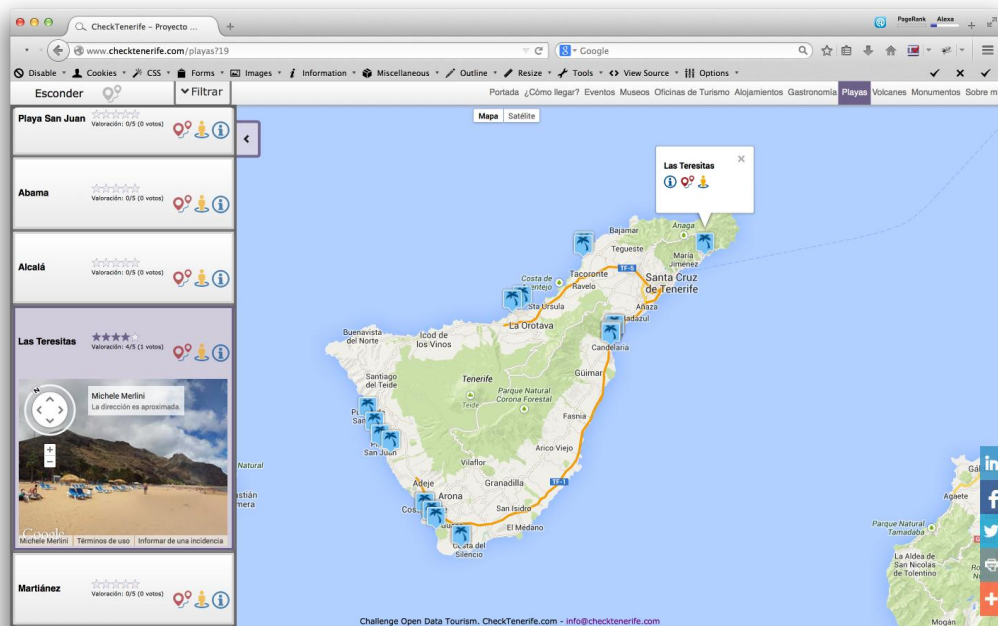




Figura 4.14.: Captura página de playas

Validación página de playas



+check ☰ 🔔 + 

Herramientas para webmasters de Google

[Página principal](#)

Herramienta de pruebas de datos estructurados

[Asistente para el marcado de datos estructurados](#)

Ayuda relacionada con:
[Solución de problemas](#)
[Centro de ayuda](#)

Herramienta de pruebas de datos estructurados

URL

HTML

VISTA PREVIA
Ejemplos ▾

Selecciona la pestaña HTML para ver el HTML recuperado y prueba a modificarlo.

Resultados de búsqueda de Google

Búsqueda personalizada de Google

Vista previa

CheckTenerife - Proyecto de Información turística de Tenerife

www.checktenerife.com/playas

El fragmento de la página aparecerá aquí. El motivo por el que no podemos mostrar texto de tu página web es que el texto mostrado depende de la consulta realizada por el usuario.

Resultado de las pruebas de autoría

La página no contiene ningún marcado de autoría. [Más información](#)

Verificación por correo electrónico de la autoría

Introduce un perfil de Google+ para comprobar si el autor ha [verificado correctamente una dirección de correo electrónico](#) en el dominio [www.checktenerife.com](#) para establecer la autoría de esta página web. [Más información](#)

Verificar autoría

Editor

La página no contiene marcado de editor. [Más información](#)

Datos estructurados extraídos

Item

type: http://schema.org/webpage

property:

- image: http://www.checktenerife.com/images/logo.png
- significantlink: Portada
- significantlink: ¿Cómo llegar?
- significantlink: Eventos
- significantlink: Museos Eventos
- significantlink: Oficinas de Turismo
- significantlink: Alojamientos
- significantlink: Gastronomía
- significantlink: Playas
- significantlink: Volcanes
- significantlink: Monumentos
- significantlink: Sobre mí

Item

type: https://schema.org/beach

property:

- name: Mesa del Mar-Tacoronte
- description:
- address: Item 1

description: Línea TITSA 021. Salida desde la estación de guaguas de Tacoronte.
 geo: *Item 2*
 description: Accesibilidad
 description: Gris
 description: Moderado
 description: Línea TITSA 021. Salida desde la estación de guaguas de Tacoronte.
 telephone: 922570015
 faxnumber: 922570015
 description: of_turismotacoronte@hotmail.com
 url: <http://www.tacoronte.es>
 url: [Buscar en Google](#)
 description: 2011-05-03 10:24:13

Item 1
type: <http://schema.org/postaladdress>
property:
 streetaddress: Mesa del Mar
 addresslocality: Tacoronte
 postalcode: 38350
 addressregion: Spain

Item 2
type: <http://schema.org/geocoordinates>
property:
 latitude: 28.505165100097656
 longitude: -16.42288589477539

Item
type: <https://schema.org/beach>
property:
 name: Barrio pesquero El Pris - Tacoronte
 description:
 address: *Item 3*
 description: Línea TITSA 023. Salida desde Estación de guaguas de Tacoronte.
 geo: *Item 4*
 description: Barrio pesquero con playa indicada para el baño y piscina natural.
 description: Gris
 description: Moderado
 description: Línea TITSA 023. Salida desde Estación de guaguas de Tacoronte.
 telephone: 922570015.922584560
 faxnumber: 922570015.922584560
 description: of_turismotacoronte@hotmail.com
 url: <http://www.tacoronte.es>
 url: [Buscar en Google](#)
 description: 2011-05-03 10:19:26

Item 3
type: <http://schema.org/postaladdress>
property:
 streetaddress: El Pris
 addresslocality: Tacoronte
 postalcode: 38350
 addressregion: Spain

Item 4

Figura 4.15.: Captura validación página de playas

4.9. Página de recursos volcánicos

Esta página muestra el geoposicionamiento de todos los recursos volcánicos de Tenerife y la información de cada uno siendo el tipo de recurso, subtipo, geoturismo, el municipio, y buscar en Google. Se puede ver la valoración de otros usuarios, puntuar, obtener la dirección de destino, ver el modo StreetView y filtrar la búsqueda por municipio o buscar por el nombre del recurso volcánico.

- **Enlace:** <http://www.checktenerife.com/volcanes>

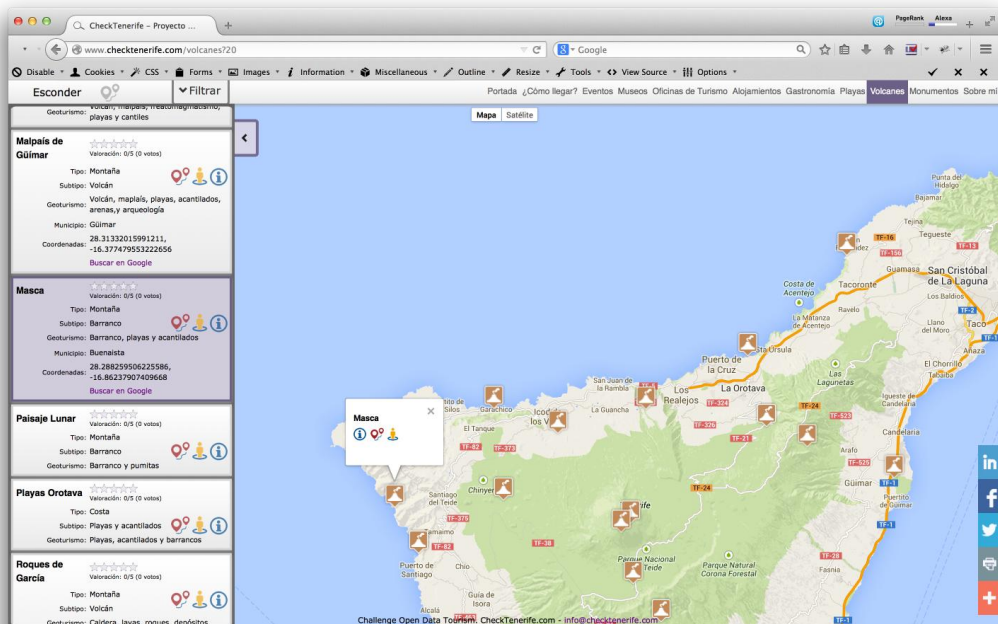




Figura 4.16.: Captura página de recursos volcánicos

Validación página de recursos volcánicos



+check ☰ 🔔 + 

Herramientas para webmasters de Google

Página principal

Herramienta de pruebas de datos estructurados

Asistente para el marcado de datos estructurados

Ayuda relacionada con:
[Solución de problemas](#)
[Centro de ayuda](#)

Herramienta de pruebas de datos estructurados

URL

HTML

VISTA PREVIA
Ejemplos ▾

Selecciona la pestaña HTML para ver el HTML recuperado y prueba a modificarlo.

Resultados de búsqueda de Google
Búsqueda personalizada de Google

Vista previa

CheckTenerife - Proyecto de Información turística de Tenerife

www.checktenerife.com/volcanes

El fragmento de la página aparecerá aquí. El motivo por el que no podemos mostrar texto de tu página web es que el texto mostrado depende de la consulta realizada por el usuario.

Resultado de las pruebas de autoría

La página no contiene ningún marcado de autoría. [Más información](#)

Verificación por correo electrónico de la autoría

Introduce un perfil de Google+ para comprobar si el autor ha [verificado correctamente una dirección de correo electrónico](#) en el dominio [www.checktenerife.com](#) para establecer la autoría de esta página web. [Más información](#)

Verificar autoría

Editor

La página no contiene marcado de editor. [Más información](#)

Datos estructurados extraídos

Item

type: http://schema.org/webpage

property:

- image: http://www.checktenerife.com/images/logo.png
- significantlink: Portada
- significantlink: ¿Cómo llegar?
- significantlink: Eventos
- significantlink: Museos Eventos
- significantlink: Oficinas de Turismo
- significantlink: Alojamientos
- significantlink: Gastronomía
- significantlink: Playas
- significantlink: Volcanes
- significantlink: Monumentos
- significantlink: Sobre mí

Item

type: https://schema.org/volcano

property:

- name: Montaña Amarilla
- description: Montaña
- description: Volcán

description:	Volcán hidromagmático, playas y acantilados
address:	<i>Item 1</i>
geo:	<i>Item 2</i>
uri:	Buscar en Google

Item 1	
type:	http://schema.org/postaladdress
property:	
addresslocality:	San Miguel
addressregion:	Spain

Item 2	
type:	http://schema.org/geocoordinates
property:	
latitude:	28.010860443115234
longitude:	-16.636510848999023

Item	
type:	https://schema.org/volcano
property:	
name:	Montaña Roja
description:	Montaña
description:	Volcán
description:	Volcán, dunas, playas, acantilados y sismitas
address:	<i>Item 3</i>
geo:	<i>Item 4</i>
uri:	Buscar en Google

Item 3	
type:	http://schema.org/postaladdress
property:	
addresslocality:	Granadilla de Abona
addressregion:	Spain

Item 4	
type:	http://schema.org/geocoordinates
property:	
latitude:	28.026649475097656
longitude:	-16.548940658569336

Item	
type:	https://schema.org/volcano
property:	
name:	Montaña Taco
description:	Montaña
description:	Volcán
description:	Volcán, malpais, freatomagmatismo, playas y cantiles
address:	<i>Item 5</i>
geo:	<i>Item 6</i>
uri:	Buscar en Google

Item 5	
---------------	--

Figura 4.17.: Captura validación página de recursos volcánicos

4.10. Página de monumentos naturales

Esta página muestra el geoposicionamiento de todos los monumentos naturales de Tenerife y la información de cada monumento siendo la dirección, descripción, teléfono, fax, email, web y buscar en Google. Se puede ver la valoración de otros usuarios, puntuar, obtener la dirección de destino, ver el modo StreetView y filtrar las búsquedas por municipio o buscar el nombre del monumento natural.

- **Enlace:** <http://www.checktenerife.com/monumentos>

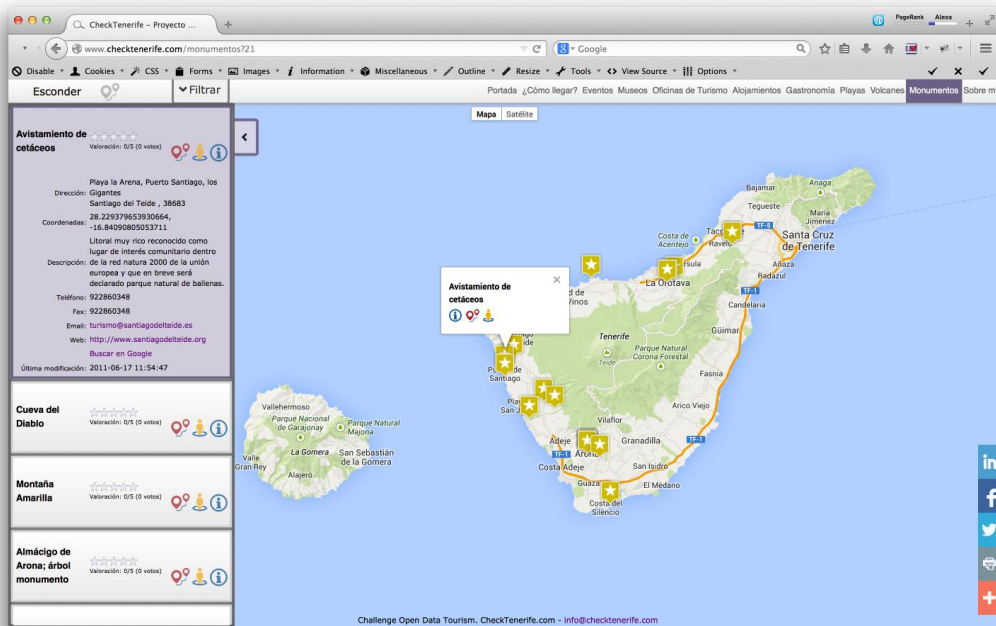




Figura 4.18.: Captura página de monumentos naturales

Validación página de monumentos naturales



+check ☰ 🔔 + 

Herramientas para webmasters de Google

Página principal

Herramienta de pruebas de

Asistente para el marcado de

datos estructurados

Ayuda relacionada con:

[Solución de problemas](#)

[Centro de ayuda](#)

Herramienta de pruebas de datos estructurados

URL
HTML

VISTA PREVIA
Ejemplos ▾

Selecciona la pestaña HTML para ver el HTML recuperado y prueba a modificarlo.

Resultados de búsqueda de Google
Búsqueda personalizada de Google

Vista previa

[CheckTenerife - Proyecto de Información turística de Tenerife](#)

www.checktenerife.com/monumentos

El fragmento de la página aparecerá aquí. El motivo por el que no podemos mostrar texto de tu página web es que el texto mostrado depende de la consulta realizada por el usuario.

Resultado de las pruebas de autoría

La página no contiene ningún marcado de autoría. [Más información](#)

Verificación por correo electrónico de la autoría

Introduce un perfil de Google+ para comprobar si el autor ha **verificado correctamente** una dirección de correo electrónico en el dominio [www.checktenerife.com](#) para establecer la autoría de esta página web. [Más información](#)

Verificar autoría

Editor

La página no contiene marcado de editor. [Más información](#)

Datos estructurados extraídos

Item

type: http://schema.org/webpage

property:

- image: http://www.checktenerife.com/images/logo.png
- significantlink: Portada
- significantlink: ¿Cómo llegar?
- significantlink: Eventos
- significantlink: Museos Eventos
- significantlink: Oficinas de Turismo
- significantlink: Alojamientos
- significantlink: Gastronomía
- significantlink: Playas
- significantlink: Volcanes
- significantlink: Monumentos
- significantlink: Sobre mí

Item

type: https://schema.org/touristattraction

property:

- name: Bosque de Agua García - TACORONTE
- address: Item 1
- description: Línea de TITSA 054 Línea de TITSA 022

<p>geo: <i>Item 2</i></p> <p>description: Bosque de Agua García en Tacoronte.- Bosque Reserva de Laurisilva, incluido dentro del Paisaje Protegido de Las Lagunetas.</p> <p>description: Línea de TITSA 054 Línea de TITSA 022</p> <p>telephone: 922570015.922584560.</p> <p>faxnumber: 922570015.922584560.</p> <p>description: of_turismotacoronte@hotmail.com</p> <p>url: http://www.tacoronte.es</p> <p>url: Buscar en Google</p> <p>description: 2011-05-03 10:06:16</p>
<p>Item 1</p> <p>type: http://schema.org/postaladdress</p> <p>property:</p> <p>streetaddress: Agua García</p> <p>addresslocality: Tacoronte</p> <p>postalcode: 38350</p> <p>addressregion: Spain</p>
<p>Item 2</p> <p>type: http://schema.org/geocoordinates</p> <p>property:</p> <p>latitude: 28.454355239868164</p> <p>longitude: -16.399240493774414</p>
<p>Item</p> <p>type: https://schema.org/touristattraction</p> <p>property:</p> <p>name: Volcán Chinyero</p> <p>address: <i>Item 3</i></p> <p>geo: <i>Item 4</i></p> <p>description: La erupción del Chinyero es la más reciente de las acaecidas en Tenerife y, por ello, la más estudiada y de la que más datos se tienen. Comenzó el 18 de noviembre de 1909 y duró diez...</p> <p>telephone: 922860348</p> <p>faxnumber: 922860348</p> <p>description: turismo@santiagodelteide.es</p> <p>url: http://www.santiagodelteide.org</p> <p>url: Buscar en Google</p> <p>description: 2011-06-09 12:46:45</p>
<p>Item 3</p> <p>type: http://schema.org/postaladdress</p> <p>property:</p> <p>streetaddress: Pueblo del Valle de Arriba</p> <p>addresslocality: Santiago del Teide</p> <p>postalcode: 38684</p> <p>addressregion: Spain</p>
<p>Item 4</p> <p>type: http://schema.org/geocoordinates</p> <p>property:</p> <p>latitude: 28.454355239868164</p> <p>longitude: -16.399240493774414</p>

Figura 4.19.: Captura validación página de monumentos naturales

4.11. Resultado del diseño web adaptativo RWD

A continuación podemos ver el diseño del portal web adaptado para cada dispositivo. El diseño se mantiene para todas las páginas a través de un menú desplegable.



Figura 4.20.: Captura del diseño web adaptativo RWD

3. Conclusiones y Trabajos Futuros

Durante el desarrollo del proyecto, he podido comprender la importancia que tiene actualmente la semántica en las páginas web y conocer la utilidad del uso de los datos públicos (OpenData) que actualmente se encuentra en crecimiento, para aprovechar las tecnologías de la información. He conocido diferentes técnicas que permitirán subir el SEO de la web, mejorando el posicionamiento de la información y mostrando los resultados más visibles RichSnippet a través de los buscadores de internet. He obtenido conocimientos sobre las tecnologías de programación Wicket, Spring y Hibernate, que utilizan la metodología de trabajo (MVC) muy utilizada actualmente. También, he mejorado los conocimientos obtenidos durante la carrera sobre los lenguajes de Java, SQL, HTML y jQuery al ponerlos en práctica sobre una plataforma web que ya se encuentra disponible para consultar todo tipo de información turística de Tenerife en la web “www.checktenerife.com”.

Por tanto, con el proyecto realizado se ha mejorado un servicio público sobre el turismo de Tenerife, permitiendo acceder fácilmente a cualquier información útil referida al turismo en un mismo portal web y accesible desde múltiples dispositivos. Con este servicio ofrecido, se espera poder mejorar el acceso a la información tanto para el turista como el propio ciudadano de la isla.

Otro aspecto a tomar en cuenta, es que si se consigue actualizar la información mensualmente desde los dataset que nos ofrece OpenData Canarias, se podría mostrar información con mayor objetividad, ya que los ficheros todavía no se actualizan hasta pasados varios meses.

Sería interesante la participación de otros usuarios para un continuo desarrollo del proyecto y para ello comparto el código para cualquier interesado:

Proyecto en Github: <https://github.com/alu0100508031/checktenerife>

Por último, mencionar que el proyecto podría tener una utilidad real en el turismo de Tenerife, pudiéndose subvencionar como servicio público para la isla.

Para ello, éste ha sido publicado en la radio Cadena SER, radio Emprendoteca y en otros medios de comunicación para darlo a conocer.

Radio Cadena SER: http://www.cadenaser.com/canarias/audios/checktenerife-com/csrsrpor/20140604csrsrcan_6/Aes

Fue presentado en Proyecto Ideas 2014 quedando como uno de los finalistas. En el minuto 42:25 se puede ver la presentación final del proyecto.

Final Proyecto Ideas 2014: <http://www.fg.ull.es/proyectoideas/?p=1711>

4. Conclusions

During the development of the Project, I have understood how important is the semantic in web sites and how helpful is to know about the use of public data (Open Data) that is growing constantly in order to take advantage of the information technologies. I have learned about the different techniques that will allow to rise the web's SEO thus improving the positioning of the information and making the results more visible (RichSnippet) by means of internet searching tools. I have also gained knowledge of Wicket, Spring and Hibernate programming technologies, which use MVC methodology, at present, widely adopted. I have also improved the knowledge acquired during my degree about SQL, HTML and jQuery Languages because I had the opportunity to put them into practice through a Web Platform that is already in use for the searching of all type of tourist information in Tenerife. This is the web site www.ckecktenerife.com.

Therefore, with the implementation of this project it has been improved a public service concerning Tourism in Tenerife, thus allowing the quick and easy access to any type of useful information related with tourism in the same web portal, being easily accessible from different devices. The aim of this service is to improve the access to the information of the island to the tourist as well as to the local citizen.

Another aspect to consider is to update the information monthly from the datasets that are offered by OpenData Canarias. In this way, the information offered would be more objective because the files until now are only updated after several months.

It would be interesting the participation of other users for the continuous development of the project. I share the code for anyone interested:

Project on Github: <https://github.com/alu0100508031/checktenerife>

Finally, I would like to mention that the project could have a real utility in Tenerife tourism and could be subsidized as a public service for the island.

Because of that, it has been presented in the Cadena Ser radio, Emprendoteca radio and other media in order to publicize it .

Radio Cadena SER: http://www.cadenaser.com/canarias/audios/checktenerife-com/csrsrpor/20140604csrsrscan_6/Aes

It was also presented in the 2014 Project Ideas, being one of the finalists. In the minute 42:25 you can see the final presentation of the project.

Final Project Ideas 2014: <http://www.fg.ull.es/proyectoideas/?p=1711>

Bibliografía

Bibliografía Física

Programador Certificado Java 2 “Curso práctico” (2a Edición)

De Antonio J. Martín Sierra, (Editorial Ra-MA) 2007.
Iniciación a la programación Java fácil de entender y de seguir, nivel básico medio.

Wicket in Action

De Martijn Dashorst y Eelco Hillenius, (Editorial Manning) 2008.

Bibliografía On-line

Comienzo del Proyecto con Wicket y Maven

Disponible: <http://wicket.apache.org/start/quickstart.html>
Último acceso: 09/07/2014

Manual “Wicket user guide” by Andrea del Bene (an.delbene@gmail.com)

Disponible: <http://wicket.apache.org/learn/books/freeguide.html>
Último acceso: 09/07/2014

Ejemplos de Wicket en Java

Disponible: <http://wicket.apache.org/learn/examples>
Último acceso: 09/07/2014

Base de Datos PostgreSQL

Disponible: <http://www.postgresql.org>
Último acceso: 09/07/2014

Configuración framework Spring

Disponible: <http://www.mysticcoders.com/blog/5-days-of-wicket-day-1>
Último acceso: 09/07/2014

Configuración framework Hibernate JPA

Disponible: <http://www.alexecollins.com/content/tutorial-hibernate-jpa-part-1>
Último acceso: 09/07/2014

Manual para usar las librerías GMAP con jQuery

Disponible: https://code.google.com/p/jquery-ui-map/wiki/jquery_ui_map_v_3_tutorial
Último acceso: 09/07/2014

Tutorial para representar los microdatos en el Mapa

Disponible: <http://www.ibm.com/developerworks/web/library/x-html5microdata1/>

Último acceso: 09/07/2014

Iconos gratuitos para el Mapa

Disponible: <http://mapicons.nicolasmollet.com/category/markers/>

Último acceso: 09/07/2014

Herramienta para realizar los test de los microdatos

Disponible: <http://www.google.com/webmasters/tools/richsnippets>

Último acceso: 09/07/2014

Repositorio de Github sincronizado con Eclipse:

Disponible: <https://github.com/alu0100508031/checktenerife>

Último acceso: 09/07/2014

Hosting Tomcat7 en EC2 Amazon

Disponible: <http://software.danielwatrous.com/java-wicket-and-hibernate-on-ec2-pre-interview-project/>

Último acceso: 09/07/2014

Registro de los dominios

Disponible: <http://www.masbaratoimposible.com/dominios.html>

Último acceso: 09/07/2014