



Universidad de Valladolid

E.T.S Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

**HdTourist: Una Aplicación Ligera
para el Consumo de Información
Semántica en Android**

Autor:

Elena Hervalejo Sevillano

Tutores:

Javier David Fernández García

Miguel Ángel Martínez Prieto

Joaquín Adiego Rodríguez

*Gracias a mis padres por su apoyo incondicional.
A Jesús por su ayuda, sin la que no habría terminado este proyecto.
Y a mis tutores, por su ánimo y su paciencia.*

Resumen

El actual "diluvio de datos" está inundando la Web con grandes volúmenes de información dando lugar a la Web de Datos. Esta es una simplificación de la Web Semántica, que proponía inicialmente superar las limitaciones de la red actual introduciendo descripciones explícitas del significado. Este tipo de datos semánticos se describen habitualmente mediante el lenguaje RDF.

DBpedia es la conversión de Wikipedia a un formato RDF. Al igual que la Wikipedia, DBpedia puede ser consultada de manera online, obteniendo múltiples resultados, cada uno de los cuales posee gran cantidad de enlaces a otras páginas. A diferencia de Wikipedia, la información semántica de DBpedia permite conocer el significado de cada uno de los enlaces, pudiendo extraer y filtrar información de manera más avanzada.

Este proyecto pretende crear un método de consulta móvil para DBpedia que se beneficie de esta característica para obtener una porción de DBpedia de interés, almacenarla localmente y poder consultarla sin necesidad de disponer de acceso a la red. Para ello se utilizará el formato HDT que permite reducir la redundancia del lenguaje RDF, para favorecer el procesamiento por parte de la máquina y la gestión de los datos.

Abstract

The current "data deluge" is flooding the Web with large volumes of information leading to the Web of Data. This is a simplification of the Semantic Web which initially proposed to overcome the limitations of the current network introducing explicit descriptions of meaning. This kind of semantic data are commonly described with the RDF language.

DBpedia is the conversion of Wikipedia to the RDF format. Similarly to Wikipedia, you can access DBpedia online and get multiple results, which links to countless pages. However, with DBpedia semantic information it is also possible to know the meaning of each one of these links. This way we can extract and filter the information in a more advanced way.

This project aims to create a mobile DBpedia search engine which can benefit from this feature to access a section of DBpedia that may be of interest and store it locally for further research, without needing to access to the network. To do so, we will use the HDT format that reduces the RDF redundancy to facilitate the engine's processing and the data management.

Índice

Resumen.....	5
Abstract.....	5
Índice	7
Índice de figuras.....	10
Índice de tablas.....	11
Capítulo 1. Introducción	13
1.1. Descripción del proyecto.....	15
1.1.1. Alcance del proyecto	15
1.1.2. Objetivos.....	15
1.2. Metodología: Proceso Unificado Ágil.....	15
1.3. Contenido de la memoria.....	18
1.4. Contenido del CD	19
Capítulo 2. La Web de Datos	21
2.1. Introducción.....	23
2.2. Web semántica.....	23
2.2.1. Aplicaciones de la Web Semántica.....	24
2.3. Linked Data	25
2.4. RDF.....	25
2.4.1. Ejemplo.....	27
2.5. SPARQL.....	30
2.6. HDT.....	30
Capítulo 3. El Proyecto	33
3.1. Introducción.....	35
3.2. DBpedia	35
3.3. Dispositivos móviles	36
3.3.1. Sistemas operativos móviles	36
3.3.2. Android	37
3.4. Estado del arte	40
3.4.1. DBpedia Mobile: A Location-Enabled Linked Data Browser	40
3.4.2. Map4RDFiOS: a tool for exploring Linked Geospatial Data	41
3.4.3. RDF On the Go: An RDF Storage and Query Processor	42
3.4.4. Linked data from your pocket.....	43
3.5. El proyecto: HdTourist	44

Capítulo 4. Desarrollo del proyecto	45
4.1. Planificación y Gestión	47
4.1.1. Estimaciones del proyecto.....	47
4.1.2. Planificación del proyecto.....	47
4.1.3. Gestión de riesgos	49
4.1.4. Seguimiento.....	54
4.2. Análisis.....	54
4.2.1. Objetivos.....	55
4.2.2. Interfaces	55
4.2.3. Restricciones	55
4.2.4. Especificación de requisitos	56
4.2.5. Modelo de casos de uso.....	58
4.2.6. Modelo de dominio.....	67
4.3. Diseño.....	69
4.3.1. Arquitectura del sistema	69
4.3.2. Diagrama de paquetes	72
4.3.3. Diagrama de clases.....	73
4.3.4. Diagramas de secuencia	77
4.4. Implementación.....	86
4.4.1. Versiones de Android compatibles con HDTourist.....	86
4.4.2. Entorno de desarrollo	86
4.4.3. Ficheros XML.....	86
4.4.4. Liberías utilizadas.....	93
Capítulo 5. Pruebas	95
5.1. Introducción.....	97
5.1.1. Pruebas de caja blanca	97
5.1.2. Pruebas de caja negra	98
Capítulo 6. Manual de Usuario	105
6.1. Instalación de HdTourist	107
6.2. Guía de uso.....	108
6.2.1. Descarga de ciudades.....	108
6.2.2. Lista de ciudades descargadas	111
6.2.3. Ver la información de una ciudad.....	113
6.2.4. Búsqueda.....	115
6.2.5. Información de archivo HDT	116

6.2.6. Configuración.....	118
Capítulo 7. Conclusiones	121
7.1. Objetivos alcanzados.....	123
7.2. Dificultades.....	123
7.3. Trabajo futuro	124
Capítulo 8. Bibliografía.....	125
Capítulo 9. Anexos.....	131
9.1. Consultas sparql para generar ficheros HDT	133
9.1.1. Madrid.hdt.....	133
9.1.2. Barcelona.hdt.....	135
9.1.3. Valladolid.hdt.....	137
9.1.4. Salamanca.hdt	139
9.1.5. London.hdt.....	142
9.2. Glosario de términos.....	145

Índice de figuras

Ilustración 1 Proceso Unificado Ágil, tareas fundamentales y fases	16
Ilustración 2 Diferencia entre la Web Sintáctica y la Web semántica.....	24
Ilustración 3 Grafo RDF	26
Ilustración 4 Grafo de ejemplo de descripción de RDF	28
Ilustración 5 HDT: Cabecera, Diccionario y Ternas.....	31
Ilustración 6 HDT: Ternas	32
Ilustración 7 Ejemplo de consulta sobre la ciudad de París.....	35
Ilustración 8 Historia de los Sistemas Operativos Móviles	37
Ilustración 9 Porcentaje de utilización de las versiones de Android	39
Ilustración 10 DBpedia Mobile: Posicionamiento de recursos	41
Ilustración 11 Map4RFDiOs: Visualización, puertos y capitales de España.....	42
Ilustración 12 Linked data from your pocket: RDF Browser.....	43
Ilustración 13 Planificación de fases e interacciones 1	48
Ilustración 14 Planificación de fases e interacciones 2	48
Ilustración 15 Diagrama de casos de uso	59
Ilustración 16 Diagrama de clases de dominio.....	68
Ilustración 17 Diagrama de despliegue.....	70
Ilustración 18 Estructura de directorios del servidor.....	70
Ilustración 19 Arquitectura lógica.....	72
Ilustración 20 Diagrama de paquetes	73
Ilustración 21 Diagrama de clases: Visualización de información de una ciudad.....	74
Ilustración 22 Diagrama de clases: Lectura de información HDT	75
Ilustración 23 Diagrama de clases: Listas de ciudades	76
Ilustración 24 Diagrama de secuencia: Inicio de la aplicación.....	78
Ilustración 25 Diagrama de secuencia: Lista de ciudades disponibles.....	79
Ilustración 26 Diagrama de secuencia: Lista de ciudades para descargar.....	81
Ilustración 27 Diagrama de secuencia: Descargar información de una ciudad	83
Ilustración 28 Diagrama de secuencia: Acceder a la información de una ciudad.....	85
Ilustración 29 Ficheros xml de configuración de información: pantalla generada.....	90
Ilustración 30 Ciclo de pruebas	97
Ilustración 31 HdTourist: Permisos de instalación.....	107
Ilustración 32 HdTourist: Confirmación de instalación.....	108
Ilustración 33 HdTourist: Pantalla de bienvenida	109
Ilustración 34 HdTourist: Inicio sin ciudades descargadas	110
Ilustración 35 HdTourist: Menú añadir ciudad.....	110
Ilustración 36 HdTourist: Confirmación para obtener el listado de ciudades.....	110
Ilustración 37 HdTourist: Pantalla de descarga de ciudades	111
Ilustración 38 HdTourist: Pantalla de filtro de ciudades para descargar	111
Ilustración 39 HdTourist: Ciudades para descargar	112
Ilustración 40 HdTourist: Menú en ciudades para descargar.....	112
Ilustración 41 HdTourist: Mapa de la ciudad con puntos geolocalizados	112
Ilustración 42 HdTourist: Pantalla principal de información	113
Ilustración 43 HdTourist: Categorías de información.....	113
Ilustración 44 HdTourist: Icono para descargar imágenes	114
Ilustración 45 HdTourist: Categoría lugares.....	114
Ilustración 46 HdTourist: Información sobre un lugar	114
Ilustración 47 HdTourist: Situación en mapa del lugar	115
Ilustración 48 HdTourist: Búsqueda de información.....	116

Ilustración 49 HdTourist: Resultados de la búsqueda	116
Ilustración 50 HdTourist: Menú de las pantallas de información.....	117
Ilustración 51 HdTourist: Información del fichero HDT	117
Ilustración 52 HdTourist: Configuración	118

Índice de tablas

Tabla 1 Fases e interacciones del proyecto.....	47
Tabla 2 Gestión de riesgos: Índice de gravedad.....	50
Tabla 3 RSK-01: Falta de tiempo	50
Tabla 4 RSK-02: Retraso en la planificación.....	51
Tabla 5 RSK-03: Falta de experiencia en la tecnología Android.....	51
Tabla 6 RSK-04: Proceso de diseño pobre.....	52
Tabla 7 RSK-05: Imposibilidad de reuniones con presencia física	52
Tabla 8 RSK-06: Pérdida de ficheros del proyecto.....	53
Tabla 9 RSK-07: Error en la captura de requisitos.....	53
Tabla 10 RSK-08: Inhabilitación del equipo utilizado	54
Tabla 11 FRQ-01: Seleccionar una ciudad para ver su información.....	56
Tabla 12 FRQ-02: Seleccionar de entre una lista de ciudades descargables	56
Tabla 13 FRQ-03: Buscar una ciudad para descargar su información.....	56
Tabla 14 FRQ-04: Visualizar información de una ciudad por categorías	56
Tabla 15 FRQ-05: Abrir vínculos de objetos relacionados.....	57
Tabla 16 FRQ-06: Búsqueda de información.....	57
Tabla 17 FRQ-07: Mostrar mapa.....	57
Tabla 18 FRQ-08: Configuración de la conexión	57
Tabla 19 NFR-01: Adaptar la visualización a la resolución de pantalla.....	57
Tabla 20 NFR-02: Adaptar la visualización a la orientación de la pantalla.....	57
Tabla 21 NFR-03: Mantener su funcionamiento sin necesidad de conexión.....	58
Tabla 22 NFR-04: Implementación de un crawler configurable	58
Tabla 23 CU-01 Acceder a la información de una ciudad.....	60
Tabla 24 CU-02 Seleccionar ciudad para descargar su información	61
Tabla 25 CU-03 Búsqueda de una ciudad para descargar su información	62
Tabla 26 CU-04 Visualizar ciudad por categorías	62
Tabla 27 CU-05 Abrir vínculo de información	63
Tabla 28 CU-06 Abrir vinculo de información geo-posicionado	64
Tabla 29 CU-07 Filtrar información de una ciudad.....	64
Tabla 30 CU-08 Mostrar mapa de una ciudad	65
Tabla 31 CU-09 Descargar mapa offline de una ciudad.....	66
Tabla 32 CU-10 Configurar opciones de conexión.....	66
Tabla 33 CU-11 Configurar el mapa a mostrar	67
Tabla 34 CU-12 Eliminar información de una ciudad	67
Tabla 35 Resultados de la consulta sparql para obtener el país	93
Tabla 36 Pruebas de caja negra realizadas.....	103

Capítulo 1. Introducción

1.1. Descripción del proyecto

El siguiente documento presenta el sistema “HdTourist: Una Aplicación Ligera para el Consumo de Información Semántica en Android”, realizado por Elena Hervalejo Sevillano como parte del Trabajo Fin de Grado del Curso Puente de acceso al Grado en Informática.

1.1.1. Alcance del proyecto

La Web de Datos enlazados es el mecanismo mediante el cual los datos estructurados con información semántica (conocidos como la Web Semántica) se vinculan e interconectan entre sí, de la misma forma que lo hacen los enlaces de las páginas web.

Hasta la fecha, el gran potencial de la Web de Datos en el ámbito de los dispositivos móviles no está siendo explotado en su totalidad. En primer lugar, existe un desconocimiento general sobre la Web de Datos, que también se traduce al ambiente de desarrollo móvil. Además, los problemas de escalabilidad inherentes al manejo de grandes cantidades de datos, se manifiestan de manera más acuciante en este tipo de dispositivos.

Sin embargo, existen notas positivas que llevan al optimismo. Por una parte, el intercambio de grandes volúmenes de información en RDF se viene realizando en plano, con su coste asociado debido a la redundancia de información. Sin embargo, ya existe el formato binario HDT que nos “relaja” esta restricción. Por otra, el almacenamiento y consulta offline en el dispositivo son operaciones costosas, pero también puede ser solventado en parte gracias a disponer de un almacenamiento comprimido.

1.1.2. Objetivos

Con el desarrollo de este proyecto se pretende realizar una primera aproximación hacia una interfaz gráfica de consulta móvil para DBpedia, centrándolo en el caso de turismo offline de ciudades. Esto es, el sistema hará uso de las capacidades del formato comprimido RDF llamado HDT, para almacenar grandes cantidades de información semántica en el móvil, facilitando su recuperación y consulta en cualquier momento.

Por tanto, los objetivos principales del proyecto son:

- Permitir descargar información HDT sobre una ciudad seleccionada.
- Permitir consultar la información HDT descargada localmente, visualizando parcial o totalmente su contenido.

El nombre de la aplicación desarrollada en este proyecto será HdTourist.

1.2. Metodología: Proceso Unificado Ágil

Para la gestión de proyecto se seguirá el Proceso Unificado Ágil.

El Proceso Unificado Ágil (AUP, del inglés Agile Unified Process) [1] es una versión simplificada del Proceso Unificado de Rational (Rational Unified Process, RUP) [2] desarrollada por Scott Ambler. Describe una aproximación al desarrollo de aplicaciones

que combina conceptos propios del proceso unificado tradicional con técnicas ágiles, con el objetivo de mejorar la productividad.

Supone un enfoque intermedio entre XP (eXtreme Programming) y el RUP, y tiene la ventaja de ser un proceso ágil que incluye explícitamente actividades y artefactos a los que la mayoría de desarrolladores ya están, de alguna manera, acostumbrados.

Al igual que el Proceso Unificado, el AUP consta de cuatro fases que el proyecto atraviesa de forma secuencial:

- **Inicio.** El objetivo de esta fase es identificar el alcance inicial del proyecto, una arquitectura potencial para el sistema y obtener, si procede, financiación para el proyecto y la aceptación por parte de los promotores del sistema.
- **Elaboración.** El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- **Construcción.** Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- **Transición.** En esta fase se valida y se lanza el producto.

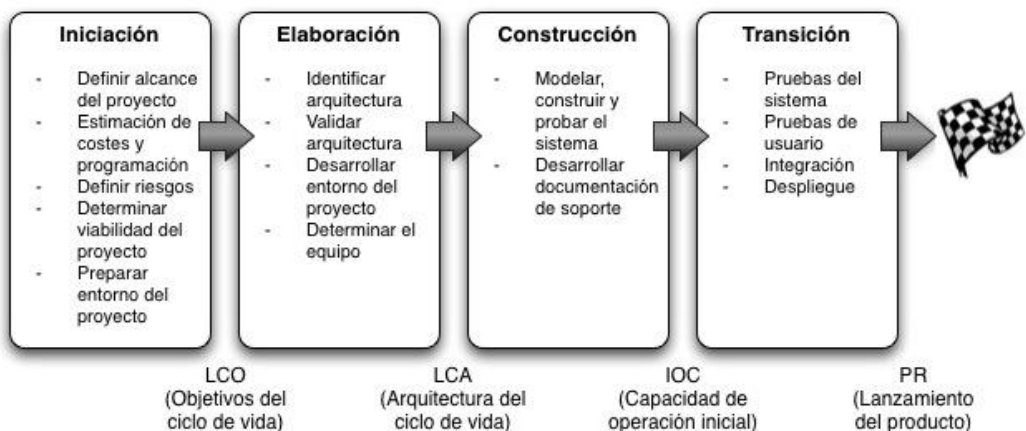


Ilustración 1 Proceso Unificado Ágil, tareas fundamentales y fases¹

A lo largo de las cuatro fases, se desarrollan actividades relativas a siete disciplinas de manera iterativa:

- **Modelado.** Su objeto es entender la lógica de negocio de la aplicación, el dominio del problema del proyecto e identificar una solución viable para el dominio del problema.
- **Implementación.** Transformar los modelos en código ejecutable y realizar pruebas básicas, en particular pruebas unitarias.
- **Pruebas.** Realizar una evaluación de los objetivos para asegurar la calidad. Esto incluye encontrar defectos, validar que el sistema funciona como fue diseñado y verificar que los requisitos se cumplen.

¹ Imagen de <http://nosolopau.com/>

- **Despliegue.** Planear la entrega del sistema y ejecutar el plan para hacer que el sistema quede disponible para los usuarios finales.
- **Gestión de la configuración.** Gestionar el acceso a los artefactos del proyecto. Esto incluye, además de la traza de versiones de los artefactos, el control de cambios y la gestión de los mismos.
- **Gestión del proyecto.** Dirige las actividades que tienen lugar dentro del proyecto, incluyendo gestión de riesgos, dirección del personal y coordinación.
- **Entorno.** Apoyar el resto del esfuerzo asegurando que los procesos, métodos y herramientas están disponibles para el equipo cuando los necesitan.

Respecto a la documentación [3], considerada parte intrínseca y necesaria del sistema, se propone un enfoque ágil y centrado en mantenerla lo más ligera posible, debido al coste añadido que implica mantenerla actualizada.

Para el Proceso Unificado Ágil, hay algunas razones válidas para crear documentación, resumidas en la siguiente lista:

- Los promotores del proyecto lo requieren.
- Para definir un modelo de contrato.
- Como ayuda a la comunicación con un grupo externo. Sin embargo, la documentación no debería ser el medio de comunicación principal.
- Como soporte a la memoria organizacional. No sólo se necesita desarrollar el software, sino que es necesario contar con documentación apropiada para su uso, soporte técnico y mantenimiento a lo largo del tiempo.
- Para labores de auditoría.
- Para ayudar a pensar sobre algo. El simple acto de escribir ideas en un papel puede contribuir a consolidarlas y a descubrir problemas en ellas.

En relación con la elaboración de documentación, en la metodología se exponen algunos puntos críticos y recomendaciones que se considera conveniente reproducir:

- Lo fundamental es la comunicación, no la documentación.
- Documentación de elementos estables.
- Utilización de un evolutivo sobre el desarrollo de documentación.
- La documentación bien escrita es útil como soporte a la memoria organizacional, pero es una manera pobre de comunicar en un proyecto.
- La documentación debería ser concisa: resúmenes y hojas de ruta son generalmente preferibles sobre otro tipo de documentación detallada.
- Con un código fuente de alta calidad y un conjunto de pruebas para respaldarlo, no necesitará demasiada documentación.
- Mantener la documentación tan ligera como sea posible. Cada artefacto que se crea y que se decide mantener necesita ser mantenido a lo largo del tiempo. Cada vez que resuelve mantener un modelo sacrifica agilidad a cambio de conservar esa información disponible para su equipo de una manera abstracta.
- La documentación debería ser sólo tan buena como sea necesario.
- La documentación comprensiva no asegura el éxito del proyecto. De hecho, incrementa las probabilidades de fracaso.
- Los modelos no son necesariamente documentos, y los documentos no son necesariamente modelos.

- La documentación forma tanta parte del sistema como el código fuente.
- El objetivo principal es desarrollar software, el objetivo secundario es facilitar su siguiente esfuerzo.
- El beneficio de tener documentación debe ser mayor que el coste de crearla y mantenerla.
- Los desarrolladores rara vez confían en la documentación, particularmente en la documentación detallada, porque generalmente no está actualizada con respecto al código.
- Cada sistema tiene sus propias necesidades de documentación.
- La inversión en documentación del sistema es una decisión de negocio, no una decisión técnica.
- Cree documentación sólo cuando la necesite en el punto adecuado del ciclo de vida.

1.3. Contenido de la memoria

La presente memoria está estructurada en diferentes capítulos, compuestos a su vez por diferentes bloques:

1. **Índices.** Guías de referencia para los contenidos y las figuras utilizadas.
2. **Introducción.** Presentación del proyecto y objetivos que se abordarán. Incluye la descripción del contenido de la memoria y la lista del contenido del CD adjunto.
3. **La Web de Datos.** Introducción a los principales conceptos de la Web Semántica, formatos RDF y HDT y lenguaje de consulta SPARQL.
4. **Proyecto.** Resumen del proyecto DBPedia y su evolución. Evolución de los dispositivos móviles y sus diferentes sistemas operativos haciendo especial hincapié en el sistema operativo Android. En este apartado se describirán también otros proyectos similares o relacionados con este.
5. **Desarrollo del proyecto.** Esta sección contiene la documentación necesaria para hacer un seguimiento del desarrollo del proyecto:
 - a. La planificación y gestión de riesgos
 - b. El modelo de Análisis
 - c. El modelo de Diseño
 - d. Implementación
6. **Pruebas.** En este capítulo se hará un resumen general de las pruebas realizadas sobre la aplicación para la depuración de errores.
7. **Manual de Usuario.** Esta sección mostrará el manual de utilización de la aplicación.
8. **Conclusiones.** Esta sección pretende mostrar posibles nuevas funcionalidades y futuros cambios a la vez que se hace una reflexión sobre los conocimientos adquiridos.
9. **Bibliografía.** Fuentes bibliográficas y referencias Web consultadas durante la ejecución del proyecto.
10. **Anexos.** En esta sección se incluirán otros documentos considerados de interés como el Glosario de términos.

1.4. Contenido del CD

El contenido del CD adjunto es el siguiente:

- **Memoria:** El documento correspondiente a dicha memoria, en un archivo único llamado "*Memoria.pdf*". Dentro de la memoria se encuentra el Manual de Usuario de la aplicación.
- **Instalable:** Archivo único instalable de la aplicación, llamado "*HDTourist.apk*".
- **Software:** Directorio que contiene el código fuente de la aplicación desarrollada y las consultas realizadas para generar los archivos HDT.
- **JavaDoc:** Documentación del código de la aplicación en formato HTML.

Capítulo 2. La Web de Datos

2.1. Introducción

La Web Semántica [4] es una extensión de la World Wide Web en la que el significado semántico de la información y de los servicios está bien definido, lo que permite devolver resultados más precisos ante una búsqueda de información.

La Web de Datos es la forma que tiene la Web Semántica de vincular los distintos datos que están distribuidos [5] en la Web, de forma que se referencian de la misma forma que lo hacen los enlaces de las páginas web.

El fin último es poder construir aplicaciones más ricas que aprovechen la gran cantidad de datos expuestos. El principal valedor de dicho modelo es la filosofía Linked Open Data [6], se trata de una corriente promovida por el propio inventor de la web y presidente del W3C, Tim Bernes Lee, cuyo objetivo es ampliar la web con una base de datos común mediante la publicación en la Web de bases de datos en el formato RDF y mediante el establecimiento de enlaces entre datos de diferentes fuentes.

2.2. Web semántica

Hoy casi todo está representado de una forma u otra en la web, y con la ayuda de un buscador, podemos encontrar información casi sobre cualquier cosa que necesitemos. No obstante, en este panorama tan favorable hay espacio para mejoras. La eficacia de los buscadores actuales también tiene sus límites. Por ejemplo, si buscamos un artículo sobre el propio García Márquez, introduciendo: “un artículo sobre García Márquez”, encontraremos decenas de artículos de García Márquez, pero ninguno que trate sobre este autor.

Esto es debido a la falta de capacidad de las representaciones en que se basa la web actual para expresar significados. Los contenidos y servicios en la web sintáctica se presentan en HTML cuyas etiquetas aportan valor estructural pero no aportan ningún valor semántico, por lo que pueden ser comprendidas por personas, pero no por máquinas.

Con estas condiciones es poco viable automatizar tareas mediante software sustituyendo a las personas. Un programa puede llevar al usuario hasta lugares en la web, generar, transportar y ofrecer la información a las personas, pero la máquina no sabe lo que esta información significa, y por tanto su capacidad de actuación autónoma es limitada. Esta misma limitación hace que la noción de semántica que manejan los buscadores web se limite a palabras clave con pesos, planas e inconexas, lo que no permite obtener resultados más elaborados.

La Web semántica [7] abandona la idea de buscar a través de la simple comparación de caracteres para avanzar hacia la búsqueda de conceptos. Esta aproximación se asemeja bastante al objetivo de la Inteligencia Artificial.

Si representamos ambas versiones de la web en forma de grafo, vemos que la diferencia fundamental entre la Web actual y la Web semántica, se centra en que en el caso de la Web actual los nodos siempre estarían compuestos por páginas HTML con arcos de relación entre páginas basados en hiperenlaces simples sin mayor clasificación. Esto supone que no existiría ninguna diferencia entre un blog personal o una página de reserva de hoteles.

El caso de la Web semántica es diferente porque cada nodo tendrá un dato de un tipo especificado y este nodo será la mínima unidad de información. Y cada arco representará el tipo de relación existente entre cada uno de estos nodos.

Por esto, se considera que la Web actual está orientada al documento y la Web semántica a los datos.

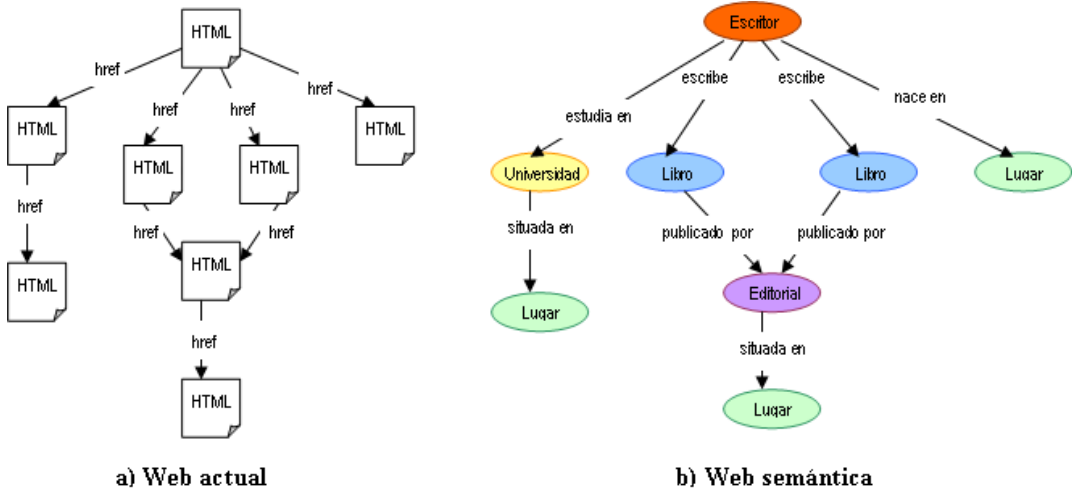


Ilustración 2 Diferencia entre la Web Sintáctica y la Web semántica²

2.2.1. Aplicaciones de la Web Semántica

Las tecnologías que apoyan a la Web semántica actualmente se utilizan en diferentes dominios de aplicaciones [8]:

- **Redes sociales, Blogs y plataformas comunitarias** (Flickr, Facebook): La Web semántica permite agregar grupos de datos (por ejemplo para una imagen: el lugar, la fecha, el nombre de las personas, el autor, la fecha de grabación, etc.) para enriquecer las posibilidades de búsqueda de informaciones y conectarse con otros miembros.
- **La búsqueda o la clasificación bibliográfica o documental:** Recientemente, la Biblioteca Nacional de Francia ha llevado a cabo un experimento destinado a mostrar las posibilidades de utilización de las técnicas de la Web semántica para efectuar búsquedas simultáneas a través de varias colecciones patrimoniales digitales. El objetivo ha sido crear vínculos entre los datos de los catálogos y hacerlos más visibles en la Web.
- **La búsqueda de información en Internet o Intranet:** Los motores de búsqueda van asimilando poco a poco las tecnologías de la Web semántica, como por ejemplo Tumbup, un nuevo motor de búsqueda vinculado a Facebook y que se apoya en la actividad de los miembros de la red social (recomendaciones sobre lugares, productos, etc.) para producir los resultados más significantes. También se puede citar los motores de búsqueda Wolfram Alpha, True Knowledge (interpretación del lenguaje natural).

² Imagen de <http://www.ekonsulta.net/>

- **La búsqueda de informaciones en la empresa:** Mediante programas que permiten recolectar, analizar y estructurar grandes volúmenes de datos.
- **La industria, la investigación y el desarrollo:** Especialmente en los dominios de tecnología (por ejemplo: aeronáutica, investigación médica) mediante la creación de ontologías (conjuntos de conceptos y de parámetros característicos de un dominio especializado) destinadas a agregar datos de diseño y de producción situadas en diferentes sistemas de información.
- **Comercio / comercio electrónico:** especialmente mediante la ontología "GoodRelations" Este vocabulario permite describir de una manera estructurada los productos, los precios, y las informaciones relativas a la empresa (por ejemplo: datos de contacto, horario de atención de las tiendas, la geolocalización, las opciones de pago, etc.). Por lo que permite especialmente a los motores de búsqueda explotar mejor estos datos esenciales para contextualizarlos en su contexto de búsqueda.

2.3. Linked Data

Los Datos Enlazados (o Linked Data) [4], son la forma que tiene la Web Semántica de relacionar datos semánticos que están distribuidos en diferentes localizaciones. De esta forma la Web de datos es una gran base de datos interconectados y distribuidos.

Para conseguir esta gran base de datos interconectada, la Web de datos se basa en los siguientes principios:

1. Usar URIs (Uniform Resource Identifier) como identificadores únicos para los recursos.
2. Seguir el protocolo HTTP para resolver la ubicación de los datos identificados mediante URIs.
3. Ofrecer información de los recursos utilizando RDF
4. Incluir enlaces a otras URIs para poder enlazar más datos.

Nombrar los conceptos o cosas mediante URIs es una forma estándar e inequívoca para referirnos a cualquier recurso, de forma que se consigue evitar ambigüedades. Se pretende el uso de URIs sobre HTTP para asegurar que cualquier recurso pueda ser buscado y accedido a través de la Web.

2.4. RDF

RDF es un framework de descripción de recursos (Resource Description Framework) [9]. Es un lenguaje de objetivo general para representar la información sobre recursos en la web. Fue desarrollado por el W3C como lenguaje para añadir metadatos legibles en la Web. RDF usa XML para su serialización.

Se basa en la idea de hacer declaraciones sobre los recursos en forma de expresiones conocidas como triples: (sujeto, predicado, objeto). El sujeto indica el recurso y el predicado denota aspectos del recurso o relación entre el sujeto y el objeto. El triple es la unidad básica de información dentro del formato RDF.

Un objeto de un triple puede funcionar como el sujeto de otro triple. Esto se puede representar en un grafo etiquetado dirigido, donde los recursos (sujetos y objetos) se corresponden con nodos, y los predicados se corresponden con las aristas. Así mismo, RDF permite una forma de reificación (una declaración de una declaración), que significa que cualquier declaración de RDF puede ser usada como sujeto en un triple.

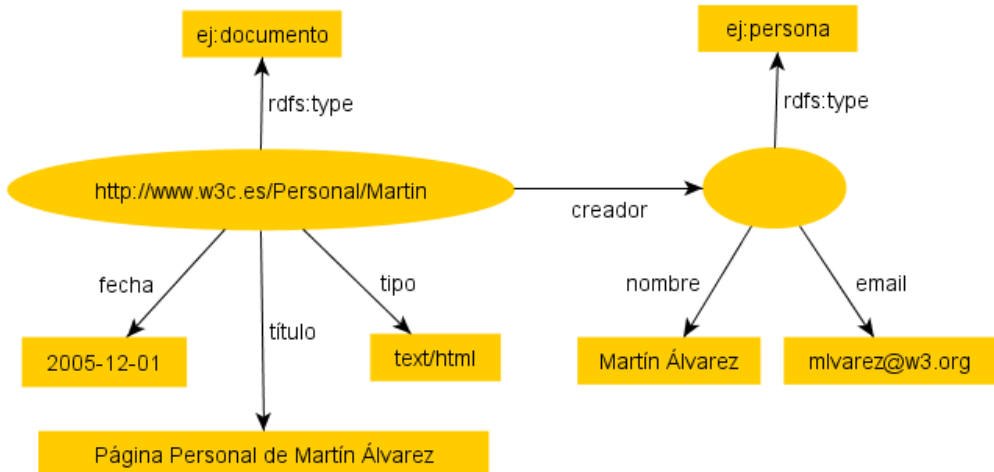


Ilustración 3 Grafo RDF³

Según el modelo de datos que propone RDF consiste en tres tipos de objetos:

- **Recursos:** Cualquier objeto de la Web identificable a través de una URI (Universal Resource Identifier). Un recurso puede ser un documento HTML, una parte de una página Web (como por ejemplo un elemento XML dentro de un documento fuente) o, generalizando la definición, cualquier objeto de información.
- **Propiedades:** Aspectos específicos, atributos o relaciones utilizadas para describir recursos. Cada tipo de propiedad tiene sus valores específicos y define los valores permitidos, los tipos de recursos que puede describir y las relaciones que existen entre las distintas propiedades.
- **Descripciones:** Son el conjunto formado por un recurso, un nombre de propiedad y el valor de esa propiedad.

Así pues, una declaración o sentencia está compuesta por 3 partes individuales:

- Sujeto: Recurso
- Predicado: Propiedad
- Objeto: Valor de la propiedad, que puede ser otro recurso, especificado por un URI, o un literal (una cadena simple de caracteres u otros tipos de datos primitivos definidos por XML). El contenido de un literal no es interpretado por RDF en sí mismo y puede contener marcado XML adicional. Los literales se distinguen de los recursos en que el modelo RDF no permite que los literales sean sujeto de una declaración.

³ Imagen de <http://www.w3c.es/>

Este conjunto de tres elementos se conoce como terna y es la unidad básica de información en RDF.

RDF no restringe la forma de serializar los triples, hay varios formatos posibles: RDF/XML [10], N3 [11], Turtle [12], Json-LD [13]... etc.

2.4.1. Ejemplo

El siguiente ejemplo está tomado de la página web de W3C [14] que describe un recurso con declaraciones "Hay una persona identificada por <http://www.w3.org/People/EM/contact#me> cuyo nombre es Eric Miller, cuyo correo electrónico dirección es em@w3.org, y cuyo título es el de Dr."

El recurso <http://www.w3.org/People/EM/contact#me> es el sujeto, que es una URI.

Los objetos son:

- "Eric Miller" (con el predicado "cuyo nombre es")
- <mailto:em@w3.org> (con el predicado "cuyo correo electrónico es")
- "Dr." (con el predicado "cuyo título es el de").

Los predicados también tienen URIs:

- "Cuyo nombre es": <http://www.w3.org/2000/10/swap/pim/contact#fullName>
- "Cuyo correo electrónico es":
<http://www.w3.org/2000/10/swap/pim/contact#mailbox>
- "Cuyo título es": <http://www.w3.org/2000/10/swap/pim/contact#personalTitle>

Además, el sujeto tiene tipo (con URI <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>), que es una persona (<http://www.w3.org/2000/10/swap/pim/contact#Person>).

Por tanto, la descripción de este recurso puede expresarse con los siguientes triples:

- <http://www.w3.org/People/EM/contact#me>,
<http://www.w3.org/2000/10/swap/pim/contact#fullName>, "Eric Miller"
- <http://www.w3.org/People/EM/contact#me>,
<http://www.w3.org/2000/10/swap/pim/contact#mailbox>, <mailto:em@w3.org>
- <http://www.w3.org/People/EM/contact#me>,
<http://www.w3.org/2000/10/swap/pim/contact#personalTitle>, "Dr."
- <http://www.w3.org/People/EM/contact#me>,
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>,
<http://www.w3.org/2000/10/swap/pim/contact#Person>



Ilustración 4 Grafo de ejemplo de descripción de RDF⁴

El grafo de la Ilustración 4 se describiría de la siguiente forma, en el formato estándar N-Triples [15]:

```
<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/2000/10/swap/pim/contact#fullName> "Eric
Miller".

<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/2000/10/swap/pim/contact#mailbox>
<mailto:em@w3.org>.

<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/2000/10/swap/pim/contact#personalTitle>
"Dr.".

<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2000/10/swap/pim/contact#Person>.
```

⁴ Imagen de <http://www.w3.org/>

De forma equivalente, puede ser escrito en formato N3 [11]:

```
@prefix eric:    <http://www.w3.org/People/EM/contact#>.
@prefix contact: <http://www.w3.org/2000/10/swap/pim/contact#>.
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

eric:me contact:fullName "Eric Miller".
eric:me contact:mailbox <mailto:em@w3.org>.
eric:me contact:personalTitle "Dr.".
eric:me rdf:type contact:Person.
```

En formato RDF/XML [10]:

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#"
xmlns:eric="http://www.w3.org/People/EM/contact#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description
rdf:about="http://www.w3.org/People/EM/contact#me">
<contact:fullName>Eric Miller</contact:fullName>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/People/EM/contact#me">
  <contact:mailbox rdf:resource="mailto:em@w3.org"/>
</rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/People/EM/contact#me">
  <contact:personalTitle>Dr.</contact:personalTitle>
</rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/People/EM/contact#me">
  <rdf:type
rdf:resource="http://www.w3.org/2000/10/swap/pim/contact#Person"
/>
  </rdf:Description>
</rdf:RDF>
```

2.5. SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) [16] es el lenguaje de consulta estandarizado para la consulta de grafos RDF, el formato con el que se modelan los datos en la Web de Datos. Se trata de un lenguaje con bastante similitud a SQL, la diferencia principal es que SQL supone que los datos están implementados en tablas y SPARQL supone que los datos están implementados en grafos.

SPARQL se puede utilizar para expresar consultas que permiten interrogar diversas fuentes de datos, si los datos se almacenan de forma nativa como RDF o son definidos mediante vistas RDF a través de algún sistema middleware. SPARQL contiene las capacidades para la consulta de los patrones obligatorios y opcionales de grafo, junto con sus conjunciones y disyunciones. SPARQL también soporta la ampliación o restricciones del ámbito de las consultas indicando los grafos sobre los que se opera. Los resultados de las consultas SPARQL pueden ser conjuntos de resultados o grafos RDF.

Con SPARQL podemos realizar consultas sobre un conjunto de triples utilizando variables, devolviendo como resultado los valores que coincidan con el grafo.

Siguiendo el ejemplo del grafo de la Ilustración 4, podríamos obtener el email de la persona cuyo nombre es “Eric Miller”:

```
SELECT ?email
WHERE {
?persona contact:fullName "Eric Miller" .
?persona contact:mailbox ?email .
}
```

2.6. HDT

La evolución de los Datos Enlazados está produciendo conjuntos de datos RDF cada vez de mayor tamaño. La necesidad de intercambiar esta información hace que ello sea un gran inconveniente ya que los datos se serializan como texto de forma muy detallada. Los teléfonos inteligentes y otros dispositivos portátiles se enfrentan a problemas similares, por lo que se hace necesario un formato más eficiente para la publicación, intercambio y consumo de RDF.

HDT (Header, Dictionary, Triples) [17] es una estructura de datos compacta e indexada para RDF que mantiene a grandes conjuntos de datos comprimidos para ahorrar espacio, manteniendo las búsquedas y la navegación por la información sin descompresión previa. Esto hace que sea un formato ideal para almacenar y compartir conjuntos de datos RDF en la Web.

Un conjunto de datos codificado con HDT está formado por tres secciones lógicas, como se muestra en la Ilustración 5, la cabecera, el diccionario y las ternas, diseñadas específicamente para lidiar con las peculiaridades de RDF.

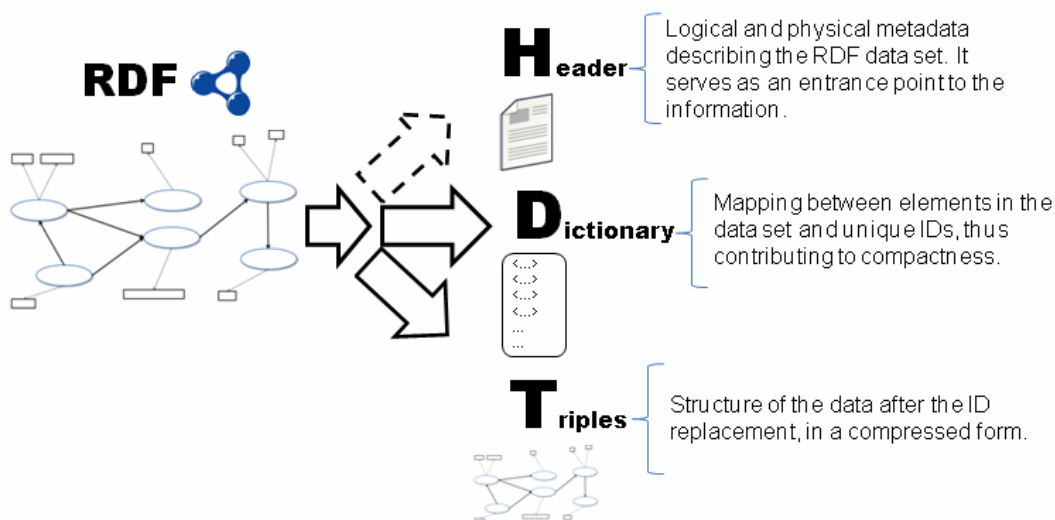


Ilustración 5 HDT: Cabecera, Diccionario y Ternas⁵

- **Cabecera:** La cabecera contiene los metadatos que describen el conjunto de datos semánticos del HDT, usando el RDF estándar. Actúa como un punto de entrada para el consumidor, que puede así hacerse una idea básica de las propiedades clave del contenido incluso antes de recuperar el conjunto de datos completo.
- **Diccionario:** El diccionario es un catálogo que comprende todos los términos utilizados en el conjunto de datos (URI's, literales y nodos vacíos). Cada término recibe un identificador único (ID), que permite la representación de una terna con tres identificadores, y cada identificador referencia dentro del diccionario a su respectivo término sujeto/predicado/objeto. Este es un primer paso de cara a lograr una buena compresión, ya que evita que términos largos se repitan una y otra vez. Además, las cadenas de caracteres similares se guardan de forma consecutiva dentro del diccionario, circunstancia que puede aprovecharse para mejorar aún más la compresión.
- **Ternas:** Como se ha explicado en el punto anterior, las ternas RDF pueden verse ahora como tuplas de tres identificadores. Por lo que esta última sección del fichero, mapea el grafo de relaciones entre los términos del conjunto de datos, tal y como se muestra en la Ilustración 6. Comprendiendo las propiedades típicas de un grafo RDF, podemos crear formas más eficientes de representar esa información, tanto para reducir su tamaño global como para ofrecer operaciones de búsqueda más eficientes.

⁵ Imagen de <http://www.rdfhdt.org/>

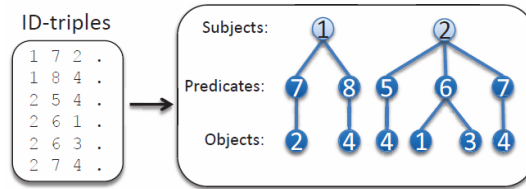


Ilustración 6 HDT: Ternas⁶

Algunas de las ventajas de HDT:

- El tamaño de los archivos es menor que en la serialización en RDF. Por lo que conlleva menos coste de ancho de banda en las transmisiones y tiempos de descarga menores.
- El archivo de HDT está indexado. Por lo que es posible realizar búsquedas con mayor rapidez sin necesidad de realizar procesos previos de análisis e indexado.
- Alto rendimiento de consulta. Por lo general, el cuello de botella de las bases de datos es el acceso a disco. Las técnicas de compresión internas de HDT permiten que la mayor parte de los datos (o incluso de todo el conjunto de datos) se pueda mantener en la memoria principal, más rápida que los discos.
- Altamente concurrentes. HDT es de sólo lectura, por lo que permite el procesamiento de consultas simultáneas.
- El formato es abierto y está en proceso de estandarización (W3C HDT Member Submission [18]).
- Las bibliotecas son de código abierto (LGPL). Se pueden adaptar según las necesidades, y la comunidad puede detectar y corregir problemas.

⁶ Imagen de <http://www.rdfhdt.org/>

Capítulo 3. El Proyecto

3.1. Introducción

Con este proyecto se pretende realizar una primera aproximación hacia una interfaz gráfica de consulta a la Web de Datos en un dispositivo móvil. Nos centraremos inicialmente en Dbpedia (conversión de Wikipedia a RDF) como un caso de aplicación. En concreto, realizaremos una interfaz de consulta de DBPedia para el turismo urbano.

Haremos uso de las capacidades del formato comprimido de RDF, el formato HDT, para almacenar grandes cantidades de información semántica en el móvil, facilitando su recuperación y consulta en cualquier momento.

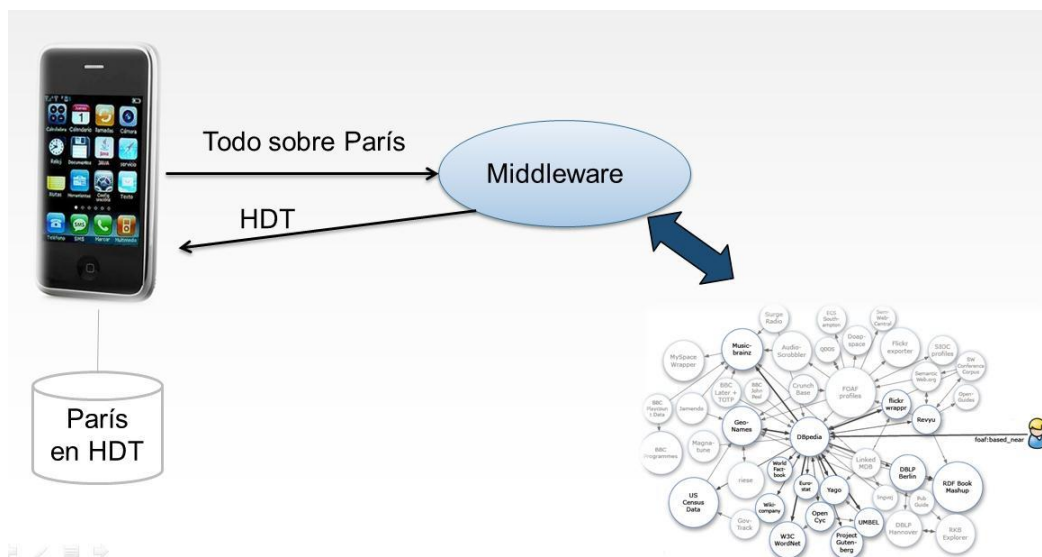


Ilustración 7 Ejemplo de consulta sobre la ciudad de París

La Ilustración 7 muestra cómo un terminal móvil consulta a un Middleware sobre toda la información relativa a París. Dicho Middleware captará la información sobre París en los contenidos de la Web de Datos relacionados (DBPedia en nuestro caso), y le devolverá un fichero HDT con toda la información. Una vez descargado el fichero, el dispositivo móvil posee la información localmente para su consumo.

3.2. DBpedia

El objetivo de DBpedia [19] es extraer información estructurada de Wikipedia y publicarla de forma abierta en la Web.

Surge en 2007 de la colaboración de la Free University of Berlin, la University of Leipzig y la compañía OpenLink Software. Ese mismo año se publica su primera versión, que ofrece consultas de forma más sofisticada (a través de SPARQL) y la posibilidad de que otras colecciones enlacen sus datos directamente con los contenidos de Wikipedia.

DBpedia aprovecha la existencia de información ya estructurada en Wikipedia, como categorías, imágenes, coordenadas geográficas e hipervínculos, tanto a otras webs como a traducciones del mismo contenido dentro de la propia Wikipedia.

DBpedia [20] está interconectada con GeoNames, Musicbrainz, CIA World Factbook, Proyecto Gutenberg, Eurostat entre otros.

En la base de datos, solo la versión en inglés, se describen 3,77 millones de entidades, entre ellas al menos 764 mil personas, 563 mil lugares, 112 mil álbumes de música, 72 mil películas y 18 mil videojuegos. Con todas las versiones se tienen 8 millones de enlaces a imágenes, 24,4 millones de enlaces a páginas externas, 27,2 millones de enlaces a datasets externos y 55,8 millones categorías de Wikipedia.2

El contenido de la base de datos está disponible bajo licencia CC-BY-SA 3.0 y GFDL (ya que el contenido se basa en la Wikipedia).

Además de la información extraída de la versión en inglés, en junio de 2011 se implementó la extracción de información de otras Wikipedias, comenzando por 15 de estas, como la versiones en español, alemán, francés, entre otras. En mayo de 2012 se lanzó el sitio web de DBpedia para el idioma español. Y en 2013 ya disponía de 111 versiones en distintos idiomas.

3.3. Dispositivos móviles

En las últimas dos décadas hemos vivido una revolución de las comunicaciones inalámbricas que ha facilitado la movilidad de las personas al reducir la dependencia del cable para comunicarnos. Gracias a esto también hemos visto una gran evolución de las prestaciones y características de los dispositivos móviles en los últimos años, llegando en muchos casos a ser un posible sustituto del ordenador portátil o de sobremesa. A la vez del gran incremento de las herramientas y lenguajes de programación para desarrollar aplicaciones sobre dispositivos móviles, así como, la creación de nuevas maneras de distribuir las mismas.

3.3.1. Sistemas operativos móviles

En este ámbito de desarrollo de aplicaciones, el salto cuantitativo proviene, junto con el desarrollo de los móviles de última generación, de la aparición de los sistemas operativos para estos dispositivos [21].

El primer sistema operativo dominante fue Symbian, cuya principal baza era contar con el apoyo de los principales fabricantes de terminales. También PalmOS para las PDA, en 2002 RIM sacó las BlackBerry, y en 2003 Microsoft comercializó Windows Mobile.

Aunque la evolución de los Smartphones empezó en 2007 cuando Apple presentó el iPhone que sentó las bases de los futuros dispositivos móviles: pantalla táctil, sensores para determinar la posición y el estado del terminal, sin teclado físico, con gran pantalla y grandes capacidades multimedia. Posteriormente se añadió GPS, compatibilidad con redes de datos y comunicaciones y una tienda de aplicaciones.

En 2005 Google compra Android Inc. y comienza a desarrollar el sistema operativo Android lanzando el primer dispositivo con Android en 2008. Este sistema operativo es código abierto, por lo que diversos fabricantes fabricaron sus propios teléfonos con Android.

En términos generales han existido 16 plataformas principales para móviles, 10 de las cuales están compitiendo en el mercado.

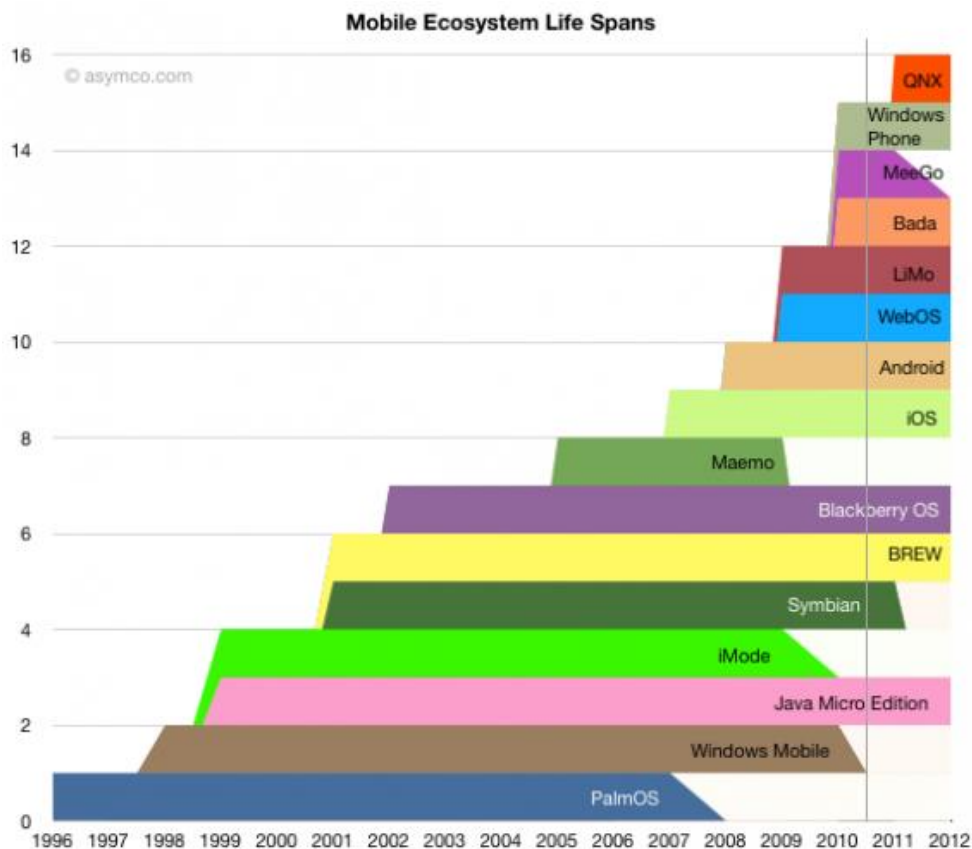


Ilustración 8 Historia de los Sistemas Operativos Móviles⁷

3.3.2. Android

Uno de los requisitos del proyecto es su desarrollo en el sistema operativo Android. Por lo que lo describiremos a continuación.

La estructura del sistema operativo Android [22] se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos. Estas se ejecutan sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución.

Los componentes principales del sistema operativo de Android son los siguientes [23]:

- **Aplicaciones:** Las aplicaciones base incluyen un cliente de correo electrónico, gestor de llamadas y de SMS, calendario, mapas, navegador, contactos y otros.
- **Marco de trabajo de aplicaciones:** Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes;

⁷ Imagen de <http://www.asymco.com/>

cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades.

- **Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Algunas son: System C library, bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima.
- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

3.3.2.1. Fragmentación

Podemos encontrar dispositivos Android con formas y tamaños completamente diferentes entre sí, con unos niveles de rendimiento igualmente desiguales entre unos terminales y otros y con unos tamaños y resoluciones de pantalla que también difieren. Del mismo modo podemos encontrar casi una decena de versiones diferentes, e incluso modificadas por fabricantes y operadores, de Android funcionando al mismo tiempo en un enorme abanico de smartphones, tablets, etc. Todo esto es lo que se define como fragmentación.

El principal problema de esta fragmentación masiva es la necesidad de los desarrolladores de probar y optimizar las aplicaciones para un número de dispositivos cada vez más grande.

Los desarrolladores también deben elegir la versión del sistema para la que desean realizar la aplicación ya que hay clases y métodos que están disponibles sólo a partir de una determinada versión. Por tanto, a la hora de seleccionar la versión del API de desarrollo hay que consultar si necesitamos alguna característica especial que solo esté disponible a partir de una versión. Todos los usuarios con versiones inferiores a la seleccionada no podrán instalar la aplicación. Por lo tanto, siempre es recomendable seleccionar la menor versión posible que nuestra aplicación pueda soportar y no olvidar consultar los porcentajes de utilización [24].

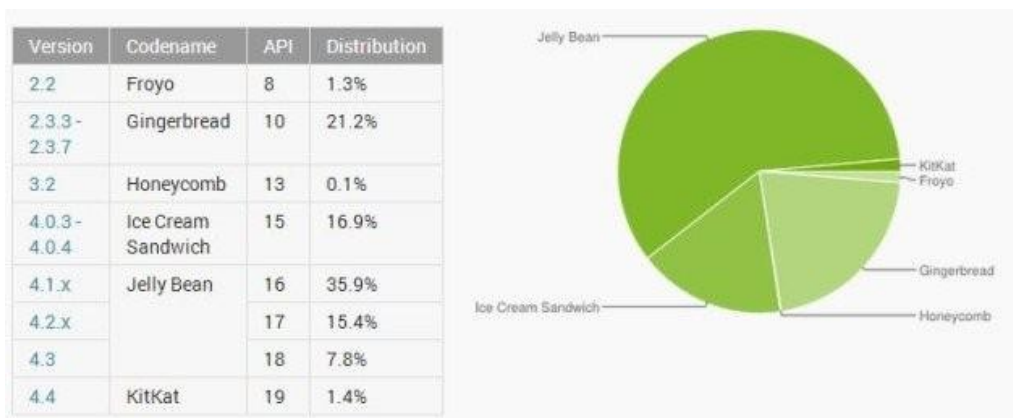


Ilustración 9 Porcentaje de utilización de las versiones de Android ⁸

3.3.2.2. Versiones

A continuación se describen las plataformas lanzadas hasta la fecha con una breve descripción de algunas de las novedades introducidas.

Las plataformas se identifican de tres formas alternativas: versión, nivel de API y nombre comercial. El nivel de API corresponde a números enteros comenzando desde 1. Para los nombres comerciales se han elegido postres en orden alfabético.

- **Apple Pie v1.0 API 1 (Septiembre 2008):** Primera versión de Android. Nunca se utilizó comercialmente.
- **Banana Bread v1.1 API 2 (Febrero 2009):** Se corrigieron algunos errores de la versión anterior.
- **Cupcake v1.5 API 3 (Abril 2009):** Se incorpora la posibilidad de teclado en pantalla con predicción de texto, se elimina la necesidad de teclado físico, capacidad de grabación avanzada de audio y vídeo. También aparecen los widgets de escritorio y live folders. Incorpora soporte para bluetooth estéreo. Y las transiciones entre ventanas se realizan mediante animaciones.
- **Donut v1.6 API 4 (Septiembre 2009):** Permite capacidades de búsqueda avanzada en todo el dispositivo. También se incorpora gestures y multi-touch. Permite la síntesis de texto a voz. Renueva la tienda de aplicaciones Play Store antes, Android Market. Mejoras en la aplicación de la cámara.
- **Eclair v2.0 API 5 (Octubre 2009):** Incorpora un API para manejar el bluetooth 2.1. Ofrece un servicio centralizado de manejo de cuentas. Mejora la gestión de contactos y ofrece más ajustes en la cámara. Se ha optimizado la velocidad de hardware. Se aumenta el número de tamaños de ventana y resoluciones soportadas. Nueva interfaz del navegador y soporte para HTML5.
- **Froyo v2.2 API 8 (Mayo 2009):** Mejora de velocidad de ejecución de las aplicaciones. Se puede preguntar al usuario si desea instalar una aplicación en un medio de almacenamiento externo. Las aplicaciones se actualizan de forma automática cuando aparece una nueva versión. Proporciona un servicio para la copia de seguridad de datos que se puede realizar desde la propia aplicación.

⁸ Imagen de <http://www.xataka.com>

Se mejora la conectividad al introducir tethering, tanto por USB como por Wi-Fi. También se añade el soporte a Wi-Fi IEEE 802.11n y notificaciones push.

- **Gingerbread v2.3 API 9 (Diciembre 2010):** Debido al éxito de Android en las nuevas tabletas, ahora soporta mayores tamaños de pantalla y resoluciones (WXGA y superiores). Mejora de la funcionalidad de “cortar, copiar y pegar” y un teclado en pantalla con capacidad multitáctil. Soporte nativo para varias cámaras, pensado en la segunda cámara usada en videoconferencia. También se mejora la gestión de energía y control de aplicaciones.
- **Honeycomb v3.0 API 11 (Febrero 2011):** Versión optimizada para dispositivos con pantallas grandes. Se mejora la reproducción de animaciones 2D/3D. Soporta procesadores multinúcleo. Varias mejoras multimedia, como listas de reproducción m3u.
- **Ice Cream Sandwich v4.0 API 14 (Octubre 2011):** Se unifican las dos versiones anteriores (2.x para teléfonos y 3.x para tabletas) en una sola compatible con cualquier tipo de dispositivo. Nuevo API de reconocimiento facial. Nuevo gestor de tráfico de datos por Internet.
- **Jelly Bean v4.1 API 16 (Julio 2012):** Introduce fluidez del interfaz de usuario. Se mejoran las notificaciones con un sistema de información expandible personalizada. Los Widgets de escritorio pueden ajustar su tamaño y hacerse sitio de forma automática al situarlos en el escritorio.
- **Jelly Bean v4.2 API 17 (Noviembre 2012):** Crear varias cuentas de usuario en el mismo dispositivo. Los Widgets de escritorio pueden aparecer en la pantalla de bloqueo. Mejoras menores en las notificaciones. Nueva aplicación de cámara que incorpora la funcionalidad Photo Sphere para hacer fotos panorámicas en 360°.
- **Jelly Bean v4.3 API 18 (Julio 2013):** Se centra en mejorar el rendimiento y la gestión del hardware. Añade pequeños cambios a ciertas funcionalidades. Mejoras en la seguridad.
- **KitKat v4.4 API 19 (Octubre 2013):** Establece una estrategia para llevar la última versión de Android a todos los dispositivos Android, tanto de gama baja como alta.

3.4. Estado del arte

Existen varios proyectos sobre la aplicación de datos semánticos en dispositivos móviles. A pesar de ello, la diferencia con este proyecto, es que ninguno de estos proyectos se basa en un formato comprimido capaz de ser almacenado y procesado por dispositivos móviles de manera offline como lo es HDT. A continuación describimos alguno de los proyectos encontrados.

3.4.1. DBpedia Mobile: A Location-Enabled Linked Data Browser

DBpedia Mobile [25] es una aplicación de localización centrada en el cliente de DBpedia para dispositivos móviles. Se basa en la posición del dispositivo para mostrar los recursos cercanos de DBpedia con etiquetas e iconos apropiados. A partir de este mapa, los usuarios pueden explorar información básica de los recursos y pueden navegar desde el conjunto de datos DBpedia a otros conjuntos de datos interrelacionados.

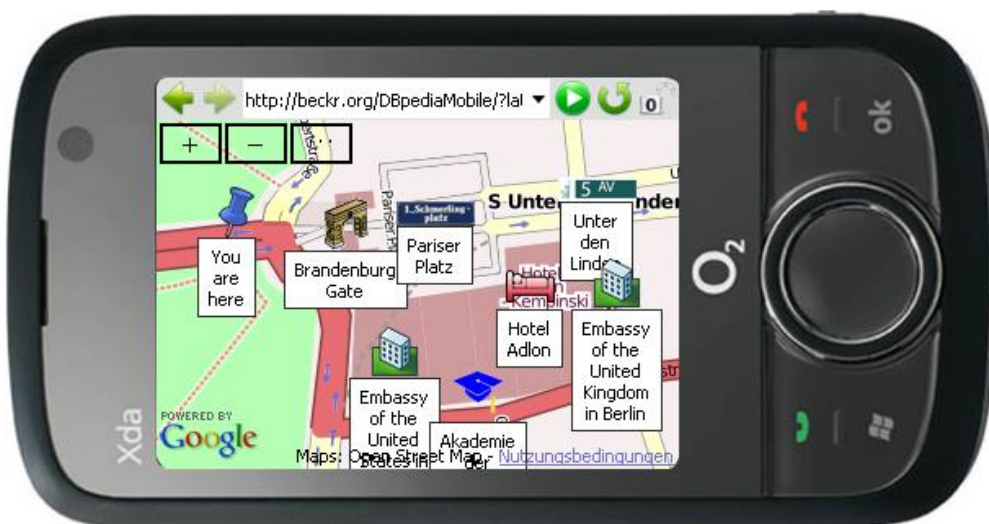


Ilustración 10 DBpedia Mobile: Posicionamiento de recursos⁹

DBpedia Mobile demuestra que el conjunto de datos DBpedia puede servir como un punto de partida útil para explorar la Web Semántica Geoespacial con un dispositivo móvil.

Para dispositivos móviles, está diseñado para las pantallas con un mínimo de QVGA (320x240 píxeles) de resolución. Actualmente, sólo está comprobado su funcionamiento en el navegador Opera Mobile 8. La mayoría de los demás navegadores móviles no tienen soporte para el API de Google Maps. En iOS actualmente no es funcional, ya que hay detalles que no se han implementado aún (en particular, el manejo del mapa con la pantalla táctil).

La aplicación también es accesible desde ordenadores convencionales. A pesar de que el posicionamiento basado en GPS es limitado en este escenario, la vista de escritorio puede ser utilizada para explorar recursos de DBpedia en todo el mundo. La aplicación ha sido probada con éxito con Firefox 2, Internet Explorer 7, Safari 3 y Opera 9.

3.4.2. Map4RDFiOS: a tool for exploring Linked Geospatial Data

Map4RDFiOS [26] es una aplicación para iOS configurable que permite explorar y visualizar conjuntos de datos RDF que se enriquecen con la información geográfica disponible a través de un endpoint SPARQL.

Por ejemplo, los usuarios finales pueden seleccionar el endpoint SPARQL con el que quieren que trabaje la aplicación o configurar en tiempo de ejecución las consultas que necesitan ser enviados a ese endpoint SPARQL. También tiene una gestión optimizada de la memoria, lo que lo hace correr mucho más suavemente que en su versión web cuando se utiliza a través de un dispositivo móvil.

Soporta múltiples servidores de mapas, como OpenStreetMap, Google Maps, Apple Maps y Nokia, que puede ser seleccionado por el usuario en tiempo de ejecución.

⁹ Imagen de <http://wiki.dbpedia.org/DBpediaMobile>

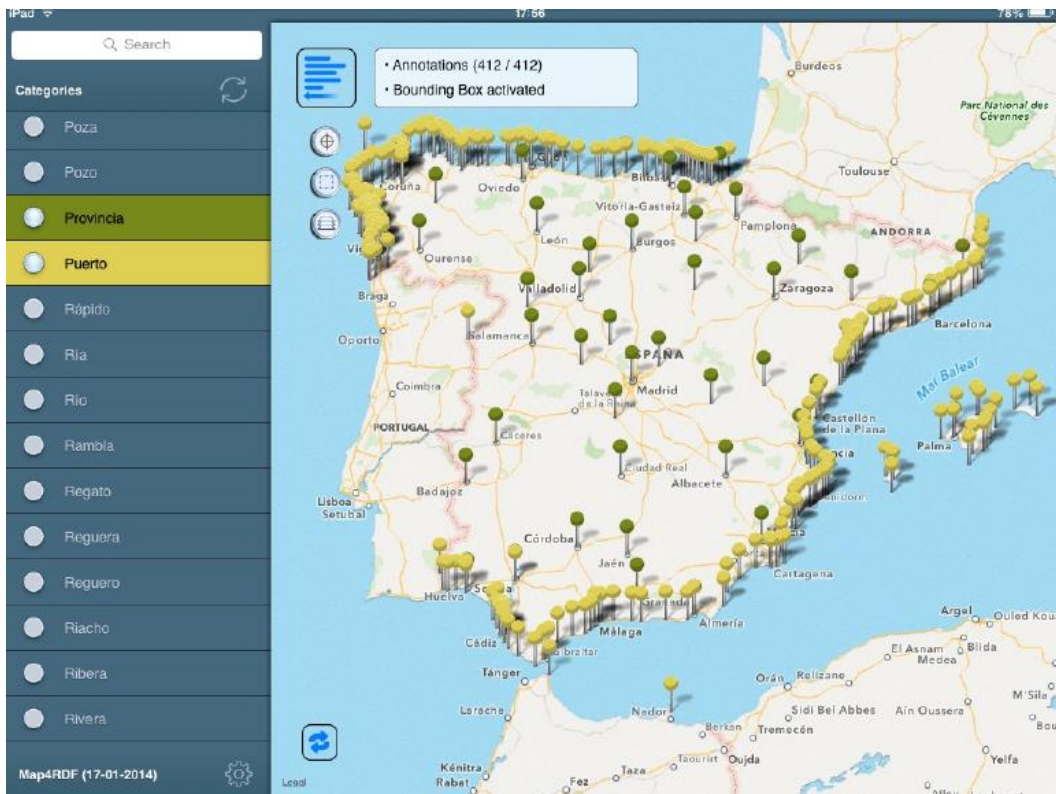


Ilustración 11 Map4RDFDiOs: Visualización, puertos y capitales de España¹⁰

Cuando se selecciona un punto en el mapa, el nombre y la categoría aparece en un cuadro de diálogo, junto con un botón de información. Al hacer clic en este botón, una nueva muestra varias opciones para obtener Información adicional sobre el punto seleccionado.

3.4.3. RDF On the Go: An RDF Storage and Query Processor

RDF On the GO [27] es un almacenamiento RDF y un procesador de consultas diseñado e implementado específicamente para los dispositivos móviles que cuentan con el sistema operativo Android.

Sigue un enfoque similar al Androjena, como las API básicas Jena incluyendo ARQ que ha sido adaptado a la plataforma Android para permitir a los desarrolladores operar directamente sobre RDF y manipular modelos de datos.

Mediante el almacenamiento y la consulta de datos RDF local en el dispositivo móvil, RDF On the Go contribuye a la mejora de la escalabilidad, la disminución de los costes de transferencia de datos, y el control del acceso a la información personal del usuario

¹⁰ Imagen de <http://www.w3.org/>

3.4.4. Linked data from your pocket

Es un framework que facilita el acceso a información semántica desde aplicaciones Android [28].

El API se debe incluir dentro de las aplicaciones que quieren tener acceso a los proveedores de RDF y dentro de las aplicaciones que desean definir nuevos proveedores de contenido RDF. La aplicación registra todos los proveedores de contenido RDF instalados tanto los localizados en el propio dispositivo como los que están en la Web, y dirige las consultas hacia proveedor pertinente.

También proporciona un servicio para compartir en la Web los proveedores de contenido RDF instalados en el dispositivo. Para mantener esta funcionalidad no se necesita la interacción con el usuario ya que está implementada como un servicio en segundo plano, pero si es necesario que el dispositivo se encuentre continuamente conectado a Internet.

Dispone de un navegador RDF Browser que actúa como un cliente de la Web Semántica. Dada una URI, el navegador hace una petición http que permite recibir la información del recurso. Si esta información contiene otras URIs, el usuario puede continuar la navegación a través de ellas.

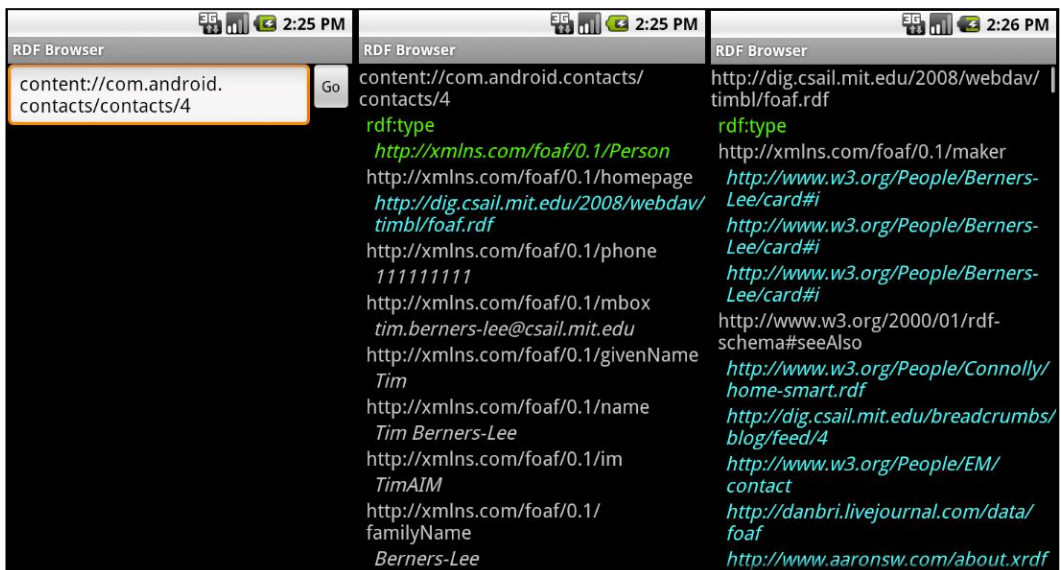


Ilustración 12 Linked data from your pocket: RDF Browser¹¹

¹¹ Imagen de Linked data from your pocket [28]

3.5. El proyecto: HdTourist

Lo que hace especial a este proyecto es el formato HDT. Ninguno de los proyectos o aplicaciones anteriores se basa en un formato comprimido capaz de ser almacenado y procesado por dispositivos móviles de manera offline como lo es HDT.

En nuestro caso, por ejemplo, el HDT generado que contiene los datos de la ciudad de Madrid, en nuestra aplicación ocupa 1.234 kilobytes mientras que el fichero RDF generado 4.988 kilobytes. Es decir, el fichero con formato HDT es 4 veces menor que el fichero con formato RDF. Esto es una ventaja a la hora de almacenarlo localmente en un dispositivo móvil y sobre todo a la hora de descargarlo a través de una red móvil, ya que el consumo de datos suele estar limitado.

Esto mismo constituye una de las motivaciones de utilizar HDT en esta aplicación. Un usuario que necesite conocer la información sobre una ciudad concreta, solo necesitará conectarse a Internet una vez para descargar el fichero que contiene la información necesaria. Posteriormente podrá consultar la información almacenada localmente siempre que lo necesite, incluso si se desplaza a zonas donde no le sea posible acceder a la red de datos (como zonas sin cobertura, o el extranjero).

Capítulo 4. Desarrollo del proyecto

4.1. Planificación y Gestión

El objetivo del de este apartado es recopilar toda la información necesaria para llevar a cabo la gestión, control y seguimiento del proyecto, desarrollando una planificación temporal e investigando los posibles riesgos pueden aparecer en el desarrollo del sistema.

4.1.1. Estimaciones del proyecto

El coste de un proyecto, en general, depende de varios aspectos. En este proyecto, el coste monetario en materiales se ha reducido a 0 gracias al uso de herramientas de libre distribución. Sin embargo el coste del tiempo requerido es un parámetro de mayor relevancia. Aproximadamente estimamos que el tiempo requerido estará comprendido entre 500 y 600 horas. Dispondremos de un recurso humano, un estudiante del Grado de Informática. La estimación temporal no viene impuesta por ninguna fecha de entrega del producto final en concreto y se ajustará a las circunstancias particulares de dicho estudiante.

4.1.2. Planificación del proyecto

La planificación será flexible debido a las circunstancias particulares del autor del trabajo. Y por este mismo motivo añadiremos un tiempo de holgura mayor a la planificación. No obstante, el proyecto deberá concluir con anterioridad a la convocatoria que tendrá lugar en junio de 2014. A continuación se expone el plan de desarrollo inicial de este proyecto.

Se ha optado por seguir el Proceso Unificado Ágil descrito anteriormente y por tanto, el desarrollo del proyecto se dividirá en cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada una dividida en iteraciones dependiendo de la complejidad de la misma.

Fases	Iteración	Descripción
Inicio	Iteración 1	Formación previa y planificación inicial.
Elaboración	Iteración 1	Análisis de casos de uso, análisis de requisitos, análisis de riesgos y modelos de análisis. Elaboración del plan de pruebas.
	Iteración 2	Modelos de diseño y diseño de la interfaz. Versión alfa del sistema.
Construcción	Iteración 1	Implementación de funcionalidades llegando a una versión beta del sistema. Realización de pruebas.
	Iteración 2	Resolución de incidencias. Implementación de funcionalidades llegando a una versión estable del sistema. Realización de pruebas.
Transición	Iteración 1	Realización del manual de usuario. Entrega de la versión final del sistema junto con la documentación.

Tabla 1 Fases e interacciones del proyecto

	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predeci
1		- Fase de Inicio	65 días	jue 02/05/13	mié 31/07/13	
2		+ Iteración 1	65 días	jue 02/05/13	mié 31/07/13	
9		- Fase de elaboración	109 días	jue 01/08/13	mar 31/12/13	1,8
10		+ Iteración 1	43 días	jue 01/08/13	lun 30/09/13	2
16		+ Iteración 2	66 días	mar 01/10/13	mar 31/12/13	10
21		- Fase de construcción	86 días	mié 01/01/14	mié 30/04/14	9
22		+ Iteración 1	43 días	mié 01/01/14	vie 28/02/14	16
28		+ Iteración 2	43 días	lun 03/03/14	mié 30/04/14	22
34		- Fase de transición	21 días	jue 01/05/14	jue 29/05/14	21
35		+ Iteración 1	21 días	jue 01/05/14	jue 29/05/14	28

Ilustración 13 Planificación de fases e interacciones 1

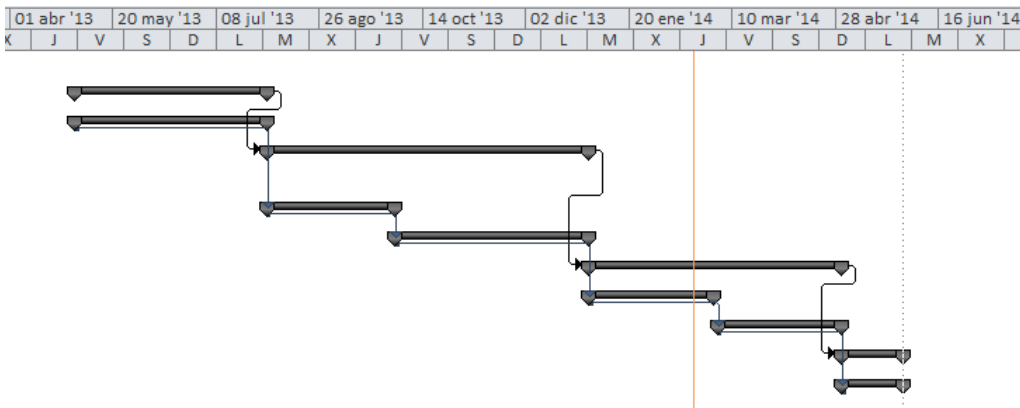


Ilustración 14 Planificación de fases e interacciones 2

4.1.2.1. Planificación de actividades

Dentro de cada fase se realizarán las siguientes actividades:

- Fase de Inicio:
 - ✓ Iteración 1
 1. Planificación de las fases y de la iteración inicial
 2. Estudio de los objetivos y el alcance del proyecto
 3. Estudio del framework Android
 4. Estudio sobre la Información semántica, RDF, HDT y Sparql
 5. Estudio de viabilidad técnica
 6. Estudio y selección de información relevante en DBPedia
 7. Seguimiento de la planificación y los riesgos
- Fase de Elaboración:
 - ✓ Iteración 1
 1. Captura y análisis de requisitos
 2. Análisis de los casos de uso
 3. Análisis de los riesgos
 4. Plan de pruebas

- 5. Seguimiento de la planificación y los riesgos
- ✓ Iteración 2
 1. Diseño de la arquitectura
 2. Diseño del front-end
 3. Desarrollo de un prototipo
 4. Seguimiento de la planificación y los riesgos
- Fase de construcción
 - ✓ Iteración 1
 1. Implementación de funcionalidades del sistema
 2. Implementación del crawler
 3. Diseño del back-end
 4. Realización de pruebas
 5. Seguimiento de la planificación y los riesgos
 - ✓ Iteración 2
 1. Resolución de incidencias
 2. Implementación de funcionalidades del sistema
 3. Diseño del front-end
 4. Realización de pruebas
 5. Seguimiento de la planificación y los riesgos
- Fase de transición
 - ✓ Iteración 1
 1. Finalización de la aplicación
 2. Elaboración del manual de usuario
 3. Balance del proyecto

4.1.3. Gestión de riesgos

En este apartado se tratarán aspectos relacionados con la planificación de la gestión de riesgos, la identificación y el análisis de riesgos, las respuestas a los riesgos y el seguimiento y control de riesgos del proyecto.

La descripción de los riesgos que se van a valorar tendrá los siguientes campos:

- **Identificador:** Número y título del riesgo
- **Descripción:** Breve descripción textual del riesgo.
- **Consecuencias:** Repercusiones en el proyecto de la aparición del riesgo.
- **Probabilidad:** Clasificación de la probabilidad de aparición del riesgo en 3 niveles: Alta, media y baja.
- **Impacto:** Clasificación de la amenaza para el desarrollo del proyecto en caso de que el riesgo se manifieste.
 - *Alto:* Afecta a la funcionalidad básica del proyecto y la duración.
 - *Medio:* Dependiendo de las circunstancias, el riesgo tendrá un plan de mitigación asociado o se asumirá su impacto en el proyecto.
 - *Bajo:* Se evitarán rápidamente, si es posible, o se asumirán como una contingencia en caso contrario.

- **Gravedad:**

Impacto	Probabilidad		
	Alta	Media	Baja
Alto	Alto	Alto	Medio
Medio	Alto	Medio	Medio
Bajo	Medio	Bajo	Bajo

Tabla 2 Gestión de riesgos: Índice de gravedad.

- **Mitigación:** Medidas a tomar en el proyecto para evitar la aparición del riesgo.
- **Contingencia:** Medidas a tomar para disminuir el impacto del riesgo una vez haya ocurrido.

4.1.3.1. Lista de riesgos

RSK-01	Falta de tiempo				
Descripción	Debido a las circunstancias personales del alumno puede ocurrir que tenga falta de tiempo por incompatibilidades con otras actividades.				
Consecuencias	Retraso en la presentación del proyecto.				
Contexto	Puede aparecer en cualquier etapa del desarrollo.				
Tipo	De proceso, de personal.				
Gravedad	Medio	Probabilidad	Alta	Impacto	Medio
Mitigación	La planificación del proyecto incluirá tiempos de holgura suficientes. Se planifica la elaboración del proyecto en 13 meses, con fecha de entrega en la convocatoria de junio de 2014.				
Contingencia	Dependiendo del grado de retraso: <ul style="list-style-type: none"> • Replanificar y aumentar las horas dedicadas al proyecto. • En caso de no ser posible la replanificación, asumir el retraso en la presentación del proyecto, adaptando de nuevo la planificación. No es asumible más tarde de la convocatoria de Septiembre de 2014. 				

Tabla 3 RSK-01: Falta de tiempo

RSK-02	Retraso en la planificación				
Descripción	A causa a una mala planificación.				
Contexto	Puede aparecer en cualquier etapa del desarrollo.				
Tipo	De proceso, de gestión.				
Consecuencias	Retraso en la presentación del proyecto.				
Gravedad	Medio	Probabilidad	Media	Impacto	Medio
Mitigación	La planificación del proyecto incluirá tiempos de holgura suficientes. Se planifica la elaboración del proyecto en 13 meses, con fecha de entrega en la convocatoria de junio de 2014.				
Contingencia	Dependiendo del grado de retraso: <ul style="list-style-type: none"> • Replanificar. • En caso de no ser posible la replanificación en el tiempo deseado, se asumirá el retraso en la presentación del proyecto, adaptando de nuevo la planificación. No es asumible más tarde de la convocatoria de Septiembre de 2014. 				

Tabla 4 RSK-02: Retraso en la planificación

RSK-03	Falta de experiencia en la tecnología Android				
Descripción	El alumno no tiene experiencia previa en la programación de aplicaciones en la plataforma Android.				
Consecuencias	Retraso en la implementación de la aplicación.				
Contexto	Este riesgo puede aparecer al final de la fase de elaboración y durante toda la fase de construcción.				
Tipo	De producto, de personal.				
Gravedad	Medio	Probabilidad	Media	Impacto	Medio
Mitigación	Se incluye en la fase de inicio tiempo de holgura suficiente para realizar un curso sobre la programación de aplicaciones en Android empleando un tiempo extra.				
Contingencia	El alumno tiene experiencia en la programación J2EE, por lo que esto mismo reduce el impacto del riesgo.				

Tabla 5 RSK-03: Falta de experiencia en la tecnología Android

RSK-04		Proceso de diseño pobre			
Descripción	Debido a la falta de experiencia el proceso de diseño puede no ser del todo correcto.				
Consecuencias	Un mal diseño o un diseño pobre podrían llevar a graves problemas en la fase de implementación e incluso a una repetición del proceso de diseño.				
Contexto	Este riesgo puede aparecer en las fases de elaboración y de construcción.				
Tipo	De proceso.				
Gravedad	Medio	Probabilidad	Baja	Impacto	Alto
Mitigación	Realizar una revisión exhaustiva en cada uno de los hitos de la etapa de diseño, para asegurar que se está conforme con lo realizado hasta el momento. También será necesario disponer de unos documentos de análisis correctos.				
Contingencia	No hay planes de contingencia, en este caso se asumirá el riesgo.				

Tabla 6 RSK-04: Proceso de diseño pobre

RSK-05		Imposibilidad de reuniones con presencia física			
Descripción	Imposibilidad de realizar reuniones con la presencia física del alumno y/o los tutores				
Consecuencias	Retraso en la realización del proyecto debido al aplazamiento de las reuniones.				
Contexto	Puede aparecer en cualquier etapa del desarrollo.				
Tipo	De proceso, de personal.				
Gravedad	Medio	Probabilidad	Alta	Impacto	Bajo
Mitigación	No hay planes de mitigación, se asumirá el riesgo.				
Contingencia	<ul style="list-style-type: none"> • Mantener la comunicación vía correo electrónico. • Se podrán celebrar reuniones mediante videoconferencia si llegar a ser necesario. 				

Tabla 7 RSK-05: Imposibilidad de reuniones con presencia física

RSK-06	Pérdida de ficheros del proyecto				
Descripción	Pérdida de ficheros				
Consecuencias	Pérdida de trabajo realizado y por tanto retraso en la planificación al tener que rehacerlo.				
Contexto	Puede aparecer en cualquier etapa del desarrollo.				
Tipo	De proceso, de herramientas.				
Gravedad	Medio	Probabilidad	Baja	Impacto	Alto
Mitigación	Se utilizará una herramienta de control de versiones, subversion, alojada en www.assembla.com . Se subirán al repositorio los archivos generados cada día al terminar las modificaciones.				
Contingencia	En caso de perder algún archivo, se obtendrá la última copia subida al repositorio.				

Tabla 8 RSK-06: Pérdida de ficheros del proyecto

RSK-07	Error en la captura de requisitos				
Descripción	Debido a cambios o falta de entendimiento puede ser necesario modificar el documento de requisitos.				
Consecuencias	Las consecuencias dependerán de en qué fase del proyecto se produzca el riesgo y de la importancia del requisito.				
Contexto	Este riesgo puede darse a lo largo de todo el proceso pero es mucho más probable en las fases iniciales. Debe tenerse en cuenta que en las fases finales esta modificación tendrá un mayor impacto.				
Tipo	De proceso.				
Gravedad	Medio	Probabilidad	Baja	Impacto	Medio
Mitigación	Revisión de la especificación de los requisitos y el modelo de los casos de uso con los tutores del proyecto.				
Contingencia	No hay planes de contingencia, en este caso se asumirá el riesgo.				

Tabla 9 RSK-07: Error en la captura de requisitos

RSK-08	Inhabilitación del equipo utilizado				
Descripción	El equipo personal de desarrollo queda inoperativo				
Consecuencias	Imposibilidad de seguir trabajando con el equipo habitual.				
Contexto	Puede aparecer en cualquier etapa del desarrollo.				
Tipo	De proceso, de herramientas.				
Gravedad	Bajo	Probabilidad	Baja	Impacto	Bajo
Mitigación	Realizar revisiones y mantenimiento del equipo de forma frecuente.				
Contingencia	En caso de no poder trabajar con el equipo habitual se trabajará con otro equipo prestado. Será necesario instalar de nuevo el IDE de desarrollo, pero se podrán recuperar los archivos del proyecto gracias al repositorio subversión.				

Tabla 10 RSK-08: Inhabilitación del equipo utilizado

4.1.4. Seguimiento

A continuación se describe el seguimiento de la planificación y la monitorización de los riesgos del proyecto.

Durante todo el desarrollo del proyecto, se ha dado varias veces el riesgo “RSK-01: *Falta de tiempo*” pero gracias al plan de mitigación (inclusión de tiempo de holguras suficientes) no ha sido necesaria la replanificación.

A partir de la fase de elaboración, en dicha fase, y en la fase de construcción, se ha dado el riesgo “RSK-05: *Imposibilidad de reuniones con presencia física*”, el plan de contingencia aplicado ha sido mantener la comunicación vía correo electrónico.

En la fase de construcción se ha producido el riesgo “RSK-03: *Falta de experiencia en la tecnología Android*”, ya que el alumno se ha encontrado con diversos problemas relacionados con la tecnología, esto ha provocado el retraso en la programación de algunas funcionalidades. La planificación con holguras ha mitigado las consecuencias de estos problemas.

4.2. Análisis

El análisis consiste en la comprensión y modelado de la aplicación y del dominio en el cual funciona. La entrada inicial de la fase de análisis es una descripción del problema que hay que resolver y proporciona una visión general conceptual del sistema propuesto.

El objetivo de este apartado es reconocer los elementos básicos del programa tal como lo percibe el usuario. La salida del análisis es un modelo formal, que muestra de una forma abstracta y resumida, los diferentes elementos del dominio, y que sirve como base para la fase de diseño del sistema.

4.2.1. Objetivos

Los objetivos principales del proyecto son:

- Permitir descargar información HDT sobre una ciudad seleccionada
- Permitir consultar la información HDT descargada localmente, visualizando parcial o totalmente su contenido.

4.2.2. Interfaces

4.2.2.1. Interfaces de sistema

La aplicación podrá descargar los ficheros de información semántica comprimida de un servidor de Internet. Una vez descargados estos ficheros, cuando se quiera consultar información sobre una ciudad en concreto, se cargará en memoria el fichero correspondiente para realizar las consultas de información.

4.2.2.2. Interfaces de usuario

La aplicación deberá ejecutarse en un dispositivo móvil con sistema operativo Android.

4.2.2.3. Interfaces Hardware

Todos los componentes se han de ejecutar en el dispositivo móvil con sistema operativo Android. La aplicación deberá funcionar en el mayor número de dispositivos posible, por lo que la versión mínima elegida será aquella que permita implementar las características requeridas.

4.2.2.4. Interfaces Software

Se usará como lenguaje de programación JAVA utilizando Android Software Development Kit. El desarrollo en un dispositivo Android y por tanto el uso de Java y el SDK de Android es un requisito de usuario.

4.2.2.5. Interfaces de comunicación

El usuario deberá disponer de conexión a Internet para descargarse la aplicación, y cada vez que busque información sobre una ciudad no descargada anteriormente. Una vez descargada no será necesaria la conexión a Internet para utilizar la aplicación. De igual modo se pretende poder hacer uso de mapas fuera de línea, pero en caso de no poder disponer de esta funcionalidad, podría solicitarse la conexión a internet.

4.2.3. Restricciones

Es un requisito del proyecto que la aplicación se ejecute en dispositivos móviles, por lo que será necesario tener en cuenta las limitaciones de memoria existentes en dichos dispositivos.

No será necesario disponer de una conexión de datos para el uso de la aplicación una vez descargados los ficheros con la información deseada. Inicialmente, tampoco se requerirá para el uso de mapas, pero los requisitos son flexibles en caso de no poder disponer de esta funcionalidad, pudiendo solicitarse la conexión a internet.

4.2.4. Especificación de requisitos

4.2.4.1. Requisitos funcionales

FRQ-01	Seleccionar una ciudad para ver su información
Descripción	Al iniciarse la aplicación mostrará una lista de ciudades con información ya descargada y permitirá seleccionar una de ellas para acceder a su información
Importancia	Vital

Tabla 11 FRQ-01: Seleccionar una ciudad para ver su información

FRQ-02	Seleccionar de entre una lista de ciudades descargables
Descripción	La aplicación mostrará una lista de ciudades disponibles para descargar su información. Si dicha información ya se encuentra en el dispositivo, el sistema pedirá confirmación para actualizarla.
Importancia	Importante

Tabla 12 FRQ-02: Seleccionar de entre una lista de ciudades descargables

FRQ-03	Buscar una ciudad para descargar su información
Descripción	El sistema permitirá buscar textualmente por una ciudad, realizando sugerencias de ciudades encontradas. Una vez seleccionada, se pedirá confirmación de la información a descargando. Si dicha información ya se encuentra en el dispositivo, el sistema pedirá confirmación para actualizarla
Importancia	Opcional

Tabla 13 FRQ-03: Buscar una ciudad para descargar su información

FRQ-04	Visualizar información de una ciudad por categorías
Descripción	El sistema permitirá visualizar información de una ciudad organizada a través de diferentes categorías. Cada categoría mostrará la información correspondiente de la ciudad de forma separada.
Importancia	Vital

Tabla 14 FRQ-04: Visualizar información de una ciudad por categorías

FRQ-05	Abrir vínculos de objetos relacionados
Descripción	El sistema permitirá acceder a la información de los objetos mostrados con información disponible.
Importancia	Importante

Tabla 15 FRQ-05: Abrir vínculos de objetos relacionados

FRQ-06	Búsqueda de información
Descripción	La aplicación permitirá realizar búsquedas textuales devolviendo aquella información que concuerde con las palabras claves de búsqueda introducidas
Importancia	Importante

Tabla 16 FRQ-06: Búsqueda de información

FRQ-07	Mostrar mapa
Descripción	La aplicación mostrará un mapa con todo lo geográficamente relacionado en un radio de búsqueda
Importancia	Importante

Tabla 17 FRQ-07: Mostrar mapa

FRQ-08	Configuración de la conexión
Descripción	La aplicación permitirá configurar la conexión a la red de datos de forma que: Se permita la conexión directa sin confirmación, no se permita la conexión sin confirmación o no se permita nunca la conexión.
Importancia	Vital

Tabla 18 FRQ-08: Configuración de la conexión

4.2.4.2. Requisitos no funcionales

NFR-01	Adaptar la visualización a la resolución de pantalla
Descripción	El sistema deberá adaptar la representación de la información a la resolución de pantalla del dispositivo en el que se ejecuta
Importancia	Vital

Tabla 19 NFR-01: Adaptar la visualización a la resolución de pantalla

NFR-02	Adaptar la visualización a la orientación de la pantalla
Descripción	El sistema deberá adaptar la representación de la información según la orientación de la pantalla en cada momento
Importancia	Importante

Tabla 20 NFR-02: Adaptar la visualización a la orientación de la pantalla

NFR-03	Mantener su funcionamiento sin necesidad de conexión continua a Internet
Descripción	El sistema deberá mantener su funcionamiento principal sin necesidad de conectarse a internet de forma continua.
Importancia	Importante

Tabla 21 NFR-03: Mantener su funcionamiento sin necesidad de conexión

NFR-04	Implementación de un crawler configurable
Descripción	Deberá de poderse configurar en el sistema la información a mostrar en cada categoría y como encontrar dicha información semántica. No obstante el proceso de crawler se hará muy puntualmente y por un “administrador” que no se considera dentro del sistema, por tanto no formará parte del diseño del proyecto.
Importancia	Media

Tabla 22 NFR-04: Implementación de un crawler configurable

4.2.5. Modelo de casos de uso

El objetivo de este apartado es detallar el Modelo de Casos de Uso de la aplicación. Con ellos describiremos el uso del sistema y cómo los usuarios interactúan con la aplicación. En los casos de uso se detallan paso a paso las acciones que realiza el sistema y que producen un resultado para un actor en particular.

4.2.5.1. Definición de actores

Consideramos actores a todas aquellas entidades externas que puedan interactuar con la aplicación.

Usuario de la aplicación: Este actor representa el usuario normal de la aplicación. Se encargará de la instalación, ejecución y uso de la aplicación. Se considera que los usuarios están familiarizados el uso de dispositivos móviles y sus aplicaciones.

4.2.5.2. Diagrama

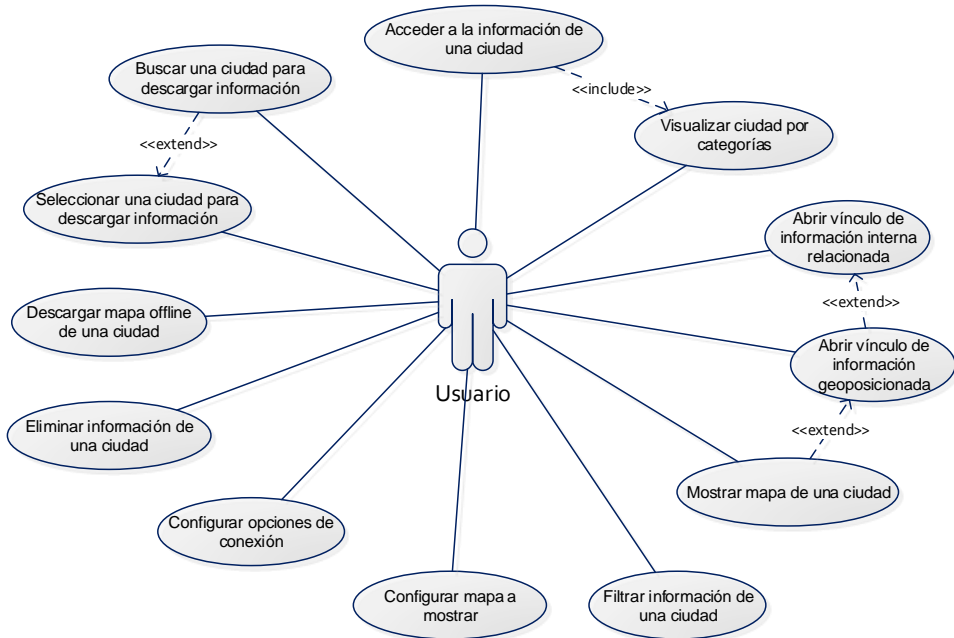


Ilustración 15 Diagrama de casos de uso

4.2.5.3. Especificación de casos de uso

CU-01	Acceder a la información de una ciudad	
Versión	20/03/13	
Requisitos asociados	FRQ-01	
Descripción	El sistema deberá comportarse como se describe a continuación cuando un usuario quiera acceder a la información de una ciudad.	
Precondición	El usuario debe de haber descargado información sobre alguna ciudad previamente.	
Secuencia normal	Paso	Acción
	P1	El usuario decide abrir la información disponible sobre una ciudad.
	P2	El sistema muestra la lista de ciudades descargas
	P3	El usuario selecciona la ciudad a la que quiere acceder
	P4	El sistema carga la información referente a la ciudad y se ejecuta el CU-04.
Postcondición	La información referente a la ciudad seleccionada es mostrada.	

Excepciones	Paso	Acción
	P2	No existe ninguna ciudad con información descargada. El sistema informa de la necesidad de descargar información.
	P4	El sistema no puede cargar la información, el sistema informa de este evento y el caso de uso finaliza.

Tabla 23 CU-01 Acceder a la información de una ciudad

CU-02	Seleccionar ciudad para descargar su información.		
Versión	20/03/13		
Requisitos asociados	FRQ-02		
Descripción	El sistema deberá comportarse como se describe a continuación cuando un usuario quiera descargar información de una nueva ciudad.		
Precondición	El sistema debe disponer de acceso a Internet.		
Secuencia normal	Paso	Acción	
	P1	El usuario decide descargar información sobre una ciudad	
	P2	P2a	Si la configuración del sistema permite la conexión a Internet, el caso de uso continua.
		P2b	Si la configuración del sistema no permite la conexión a Internet se comunica al usuario y el caso de uso finaliza.
		P2c	Si la configuración del sistema indica que hay que consultar antes de conectarse a Internet, el sistema pide confirmación al usuario antes de continuar el caso de uso.
	P3	El sistema muestra la lista de ciudades disponibles en el servidor. Si existe una búsqueda, muestra las ciudades que coincidan con los términos de búsqueda	
	P4	El usuario selecciona la ciudad deseada	
	P5	P5a	Si la ciudad seleccionada ya estaba descargada, se pide una confirmación para actualizarla.
		P5b	El sistema informa el tamaño de la información a descargar y pide la confirmación al usuario.
	P6	P6a	El usuario acepta la confirmación
		P6b	El usuario cancela la acción y el caso de uso finaliza
P7	El sistema descarga la información referente a la ciudad.		

Postcondición	La ciudad descargada se muestra en la lista de ciudades disponibles	
Excepciones	Paso	Acción
	P3	No se dispone de conexión a Internet. El sistema muestra un error y el caso de uso finaliza.
	P7	El sistema no puede descargar la información, el sistema informa de este evento y el caso de uso finaliza.

Tabla 24 CU-02 Seleccionar ciudad para descargar su información

CU-03	Búsqueda de una ciudad para descargar su información		
Versión	20/03/13		
Requisitos asociados	FRQ-03		
Descripción	El sistema deberá comportarse como se describe a continuación cuando un usuario quiera buscar una ciudad para descargar su información.		
Precondición	El sistema debe disponer de acceso a Internet.		
Secuencia normal	Paso	Acción	
	P1	El usuario decide descargar información sobre una ciudad	
	P2	P2a	Si la configuración del sistema permite la conexión a Internet, el caso de uso continúa.
		P2b	Si la configuración del sistema no permite la conexión a Internet se comunica al usuario y el caso de uso finaliza.
		P2c	Si la configuración del sistema indica que hay que consultar antes de conectarse a Internet, el sistema pide confirmación al usuario antes de continuar el caso de uso.
	P2	El sistema muestra un campo para introducir el criterio de búsqueda	
	P3	El usuario introduce el criterio de búsqueda	
	P4	P4a	Si existen resultados con ese criterio de búsqueda, el caso de uso continúa.
		P4b	Si no existen resultados con ese criterio de búsqueda se muestra un mensaje de aviso y se ejecuta el paso P2.
	P5	El sistema muestra las ciudades que coincidan con los términos de búsqueda.	
	P6	El usuario selecciona la ciudad deseada.	
P7	P7a	Si la ciudad seleccionada ya estaba descargada, se pide una confirmación para actualizarla.	

	P7b	El sistema informa el tamaño de la información a descargar y pide la confirmación al usuario.
	P8	P8a El usuario acepta la confirmación
		P8b El usuario cancela la acción y el caso de uso finaliza
	P9	El sistema descarga la información referente a la ciudad.
Postcondición	El sistema muestra las ciudades coincidentes con los criterios de búsqueda	
Excepciones	Paso	Acción
	P2	No se dispone de conexión a Internet. El sistema muestra un error y el caso de uso finaliza.

Tabla 25 CU-03 Búsqueda de una ciudad para descargar su información

CU-04	Visualizar ciudad por categorías	
Versión	20/03/13	
Requisitos asociados	FRQ-04	
Descripción	El sistema deberá comportarse como se describe a continuación cuando el usuario desee acceder a la información de una ciudad.	
Precondición	El caso de uso ejecutado anteriormente finaliza generando una lista de ciudades ya descargadas.	
Secuencia normal	Paso	Acción
	P1	El usuario selecciona una ciudad de la lista.
	P2	El sistema muestra las categorías de información disponibles.
	P3	El usuario accede a una categoría.
	P4	El sistema muestra la información correspondiente a la categoría y a la ciudad seleccionada
Postcondición	El sistema muestra la información de la ciudad y categoría seleccionadas.	
Excepciones	Paso	Acción
	P4	No se dispone de información. El sistema muestra un error y el caso de uso finaliza.

Tabla 26 CU-04 Visualizar ciudad por categorías

CU-05	Abrir vinculo de información interna relacionada	
Versión	20/03/13	
Requisitos asociados	FRQ-05	

Descripción	El sistema deberá comportarse como se describe a continuación cuando el usuario pulse sobre un vínculo de información interna relacionada con la ciudad sobre la cual hay más información accesible.	
Precondición	El caso de uso ejecutado anteriormente es el CU-04 que finaliza mostrando información en la cual hay vínculos de información interna relacionada con la ciudad y accesible.	
Secuencia normal	Paso	Acción
	P1	El usuario selecciona un enlace de información.
	P2	El sistema muestra la información disponible.
Postcondición	El sistema muestra la información del recurso vinculado.	
Excepciones	Paso	Acción
	P2	No se dispone de información. El sistema muestra un error y el caso de uso finaliza.

Tabla 27 CU-05 Abrir vínculo de información

CU-06	Abrir vinculo de información geo-posicionado		
Versión	20/03/13		
Requisitos asociados	FRQ-05		
Descripción	El sistema deberá comportarse como se describe a continuación cuando el usuario pulse sobre un vínculo de información geo-posicionado.		
Precondición	El caso de uso ejecutado anteriormente es el CU-04 que finaliza mostrando información en la cual hay vínculos de información geo-posicionados.		
Secuencia normal	Paso	Acción	
	P1	El usuario selecciona un punto de información geo-posicionado.	
	P2	P2a	Si la configuración indica que el mapa utilizado será google maps y
		P2a1	También se permite la conexión a Internet, se ejecuta P3.
		P2a2	No permite la conexión a Internet se comunica al usuario y el caso de uso finaliza.
		P2a3	Indica que hay que consultar antes de conectarse a Internet, el sistema pide confirmación al usuario antes de continuar con el P3.
	P2b	P2b	Si la configuración del sistema indica que los mapas deben ser offline
P2b1		Y los mapas offline de la ciudad están disponibles, se ejecuta el P3.	

		P2b2	Y los mapas offline de la ciudad no están disponibles, se comunica al usuario y el caso de uso finaliza.
	P3	El sistema muestra el mapa de la ciudad el punto geo-posicionado.	
Postcondición	El sistema muestra mapa con la información geo-posicionada.		
Excepciones	Paso	Acción	
	P2a1	No se dispone de conexión a Internet. El sistema muestra un error y el caso de uso finaliza.	

Tabla 28 CU-06 Abrir vinculo de información geo-posicionado

CU-07	Filtrar información de una ciudad			
Versión	20/03/13			
Requisitos asociados	FRQ-06			
Descripción	El sistema deberá comportarse como se describe a continuación cuando el usuario desee realizar una búsqueda textual.			
Precondición	Se ha accedido previamente a la información de una ciudad ya descargada CU-01			
Secuencia normal	Paso	Acción		
	P1	El usuario accede a la búsqueda dentro de una ciudad.		
	P2	El sistema muestra un campo para introducir el criterio de búsqueda.		
	P3	El usuario introduce el criterio de búsqueda		
	P4	P4a	Si hay más de un resultado el sistema muestra una lista accesible con información resumida	
		P4b	Si solo hay un resultado el sistema muestra la información encontrada.	
		P4c	Si hay más de 20 resultados se muestra un mensaje indicando que la búsqueda debe ser más restrictiva y se ejecuta el paso P2.	
		P4d	Si no hay resultados se muestra un mensaje de aviso y se ejecuta el paso P2	
Postcondición	El sistema muestra la información coincidente con los criterios de búsqueda			
Excepciones	Paso	Acción		

Tabla 29 CU-07 Filtrar información de una ciudad

CU-08	Mostrar mapa de una ciudad		
Versión	20/03/13		

Requisitos asociados	FRQ-07		
Descripción	El sistema deberá comportarse como se describe a continuación cuando el usuario desee ver en un mapa los elementos con geo-posicionamiento dentro de una ciudad.		
Precondición	Se ha accedido previamente a la información de una ciudad ya descargada CU-01		
Secuencia normal	Paso	Acción	
	P1	El usuario accede al mapa de una ciudad.	
	P2	P2a	Si los mapas offline de la ciudad están disponibles, se ejecuta el P3.
		P2b	Si los mapas offline de la ciudad no están disponibles, se comunica al usuario y el caso de uso finaliza.
	P3	El sistema muestra el mapa de la ciudad con los lugares con geo-posicionamiento.	
Postcondición	El sistema muestra el mapa en pantalla con los recursos posicionados		
Excepciones	Paso	Acción	

Tabla 30 CU-08 Mostrar mapa de una ciudad

CU-09	Descargar mapa offline de una ciudad		
Versión	20/03/13		
Requisitos asociados	FRQ-07 y NFR-03		
Descripción	El sistema deberá comportarse como se describe a continuación cuando el usuario solicite descargar el mapa offline de una ciudad		
Precondición	La información de la ciudad debe haber sido descargada previamente CU-02 o CU-03.		
Secuencia normal	Paso	Acción	
	P1	El usuario solicita la descarga del mapa offline de una ciudad.	
	P2	P2a	La configuración del sistema permite la conexión a Internet, se ejecuta P3.
		P2b	La configuración del sistema no permite la conexión a Internet se comunica al usuario y el caso de uso finaliza.
		P2c	La configuración del sistema indica que hay que consultar antes de conectarse a Internet, el sistema pide confirmación al usuario antes de continuar con el P3.
	P3	El sistema comprueba la existencia de una conexión Wifi, si no existe, se comunica al usuario y el caso de uso finaliza	
P4	El sistema inicia la descarga de los mapas offline		

Postcondición	El sistema descarga los mapas y estos se mantendrán disponibles sin necesidad de conexión.	
Excepciones	Paso	Acción
	P4	Si la conexión se corta se mostrará un error y el caso de uso finalizará sin que los mapas queden disponibles.

Tabla 31 CU-09 Descargar mapa offline de una ciudad

CU-10	Configurar opciones de conexión	
Versión	20/03/13	
Requisitos asociados	FRQ-08	
Descripción	El sistema deberá comportarse como se describe a continuación cuando el usuario acceda a la configuración del programa para configurar las limitaciones de la conexión.	
Precondición		
Secuencia normal	Paso	Acción
	P1	El usuario accede a la configuración del programa
	P2	El sistema muestra las opciones de configuración disponibles y los valores actuales
	P3	El usuario selecciona el modo de conexión
	P4	El sistema muestra las posibilidades para limitar o no limitar la conexión.
	P5	El usuario selecciona una opción
	P6	El sistema guarda la configuración
Postcondición	El sistema tendrá en cuenta la configuración de la conexión cada vez que se requiera un acceso a Internet.	
Excepciones	Paso	Acción

Tabla 32 CU-10 Configurar opciones de conexión

CU-11	Configurar el mapa a mostrar	
Versión	20/03/13	
Requisitos asociados	NFR-03	
Descripción	El sistema deberá comportarse como se describe a continuación cuando el usuario acceda a la configuración del programa para elegir el tipo de mapa que desea que se muestre.	
Precondición		
Secuencia normal	Paso	Acción

	P1	El usuario accede a la configuración del programa
	P2	El sistema muestra las opciones de configuración disponibles y los valores actuales
	P3	El usuario dispone en la opción de mapas el valor que dese
	P4	El sistema guarda la configuración
Postcondición	El sistema tendrá en cuenta la configuración de los mapas cada vez que el usuario desee visualizar un punto geolocalizado	
Excepciones	Paso	Acción

Tabla 33 CU-11 Configurar el mapa a mostrar

CU-12	Eliminar información de una ciudad	
Versión	01/03/14	
Requisitos asociados		
Descripción	El sistema deberá comportarse como se describe a continuación cuando el usuario decida eliminar la información de una ciudad.	
Precondición	La información de la ciudad debe haber sido descargada previamente CU-02 o CU-03.	
Secuencia normal	Paso	Acción
	P1	El usuario selecciona la opción de eliminar la información de una ciudad
	P2	El sistema elimina la información de la ciudad
Postcondición	La información de la ciudad ya no estará accesible hasta que no se vuelva a descargar.	
Excepciones	Paso	Acción

Tabla 34 CU-12 Eliminar información de una ciudad

4.2.6. Modelo de dominio

En este apartado se describen las clases conceptuales más significativas y se ofrece una descripción de la funcionalidad de sistema, de tal forma que se satisfagan todos y cada uno de los requisitos software desarrollados en los apartados anteriores.

4.2.6.1. Diagrama de dominio

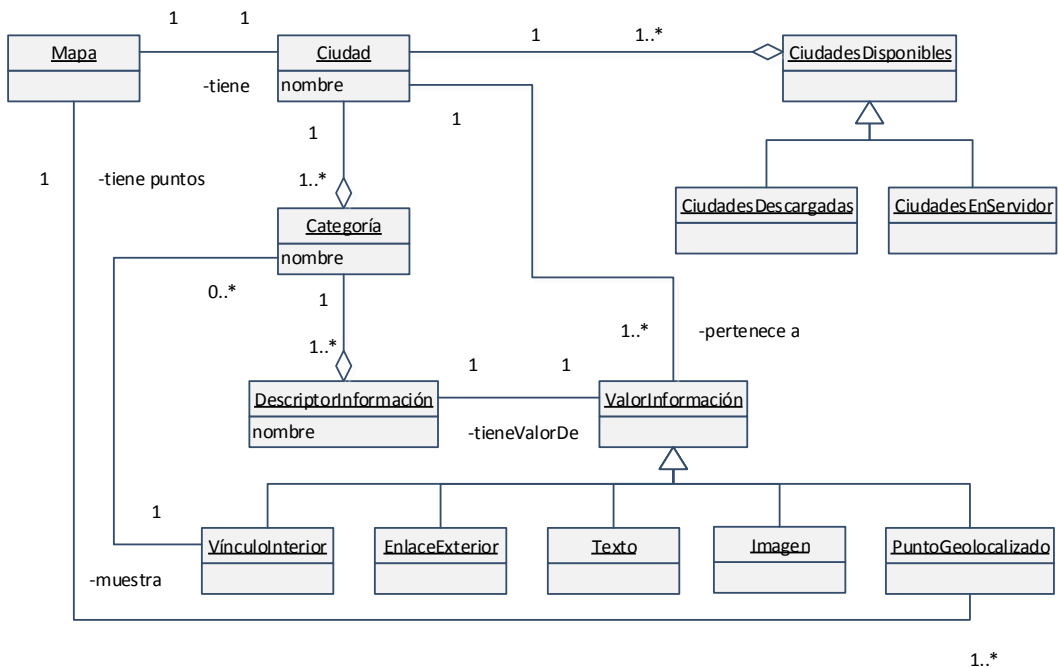


Ilustración 16 Diagrama de clases de dominio

4.2.6.2. Descripción de clases conceptuales

- **Ciudad:** Ciudad sobre la que se mostrará la información
- **Categoría:** Conjunto de categorías de información sobre las que la aplicación mostrará información. Una ciudad puede contener varias categorías de información: Información general, demografía, monumentos y lugares de interés... etc. Puede haber otras categorías, no relacionadas directamente con las ciudades, pero si con vínculos de información mostrados en otras categorías. Estos vínculos, al abrirse mostrarán información de la categoría correspondiente.
- **DescriptorInformación:** Una categoría mostrará uno o normalmente varios datos de información diferentes. Por ejemplo, dentro de la categoría demografía, se podría mostrar la densidad de población, la edad media de población, la proporción de hombres y mujeres... etc.
- **ValorInformación:** Contiene el valor del dato de información concreto, puede ser de varios tipos:
 - **VínculoInterior:** Un vínculo es un dato de información accesible, es decir, al seleccionarlo, podremos ampliar la información relacionada con la ciudad mostrada inicialmente.
 - **EnlaceExterior:** Un enlace a información exterior a la aplicación. Por ejemplo una página web.

- **Texto:** Un dato de información textual, como el gentilicio o el propio nombre de la ciudad.
- **Imagen:** Una imagen de un elemento concreto.
- **Geolocalización:** Se tratará de unas coordenadas geográficas que podrán ser representadas en un mapa.
- **Mapa:** Habrá un mapa accesible desde la ciudad donde se mostrarán todos los tipos geolocalizados.
- **CiudadesDisponibles:** Habrá una lista de ciudades disponibles:
 - **CiudadesDescargadas,** ciudades descargadas en el dispositivo.
 - **CiudadesEnServidor,** ciudades no descargadas pero disponibles en el servidor para descargar.

4.3. Diseño

La fase de diseño se define como el proceso en el cual se convierten las especificaciones de los requisitos de la fase de análisis en una representación del software que se quiere construir.

A lo largo de esta fase se transforma el modelo de dominio de la información creado durante el análisis en las estructuras de datos necesarias para implementar el software.

4.3.1. Arquitectura del sistema

La arquitectura de un sistema abarca diferentes dimensiones:

- **Despliegue de la arquitectura:** Describe el sistema en términos de la asignación de los procesos a unidades de procesos y describe la configuración de la red.
- **Arquitectura Lógica:** Describe el sistema en términos de su organización conceptual en capas, paquetes, frameworks importantes, clases, interfaces y subsistemas.

4.3.1.1. Diagrama de despliegue

La aplicación está diseñada para su uso en dispositivos móviles con Sistema Operativo Android.

Para la descarga de ficheros de información y de mapas offline es necesario un servidor que contenga los archivos solicitados por el terminal. En este proyecto utilizaremos el servidor proporcionado por los tutores del proyecto.

La conexión a este servidor podrá realizarse tanto a través de redes móviles como a través de redes Wifi, siendo recomendable para el usuario la conexión a través de redes Wifi sobre todo para la descarga de mapas offline, debido al tamaño de estos y a las limitaciones de las operadoras móviles.

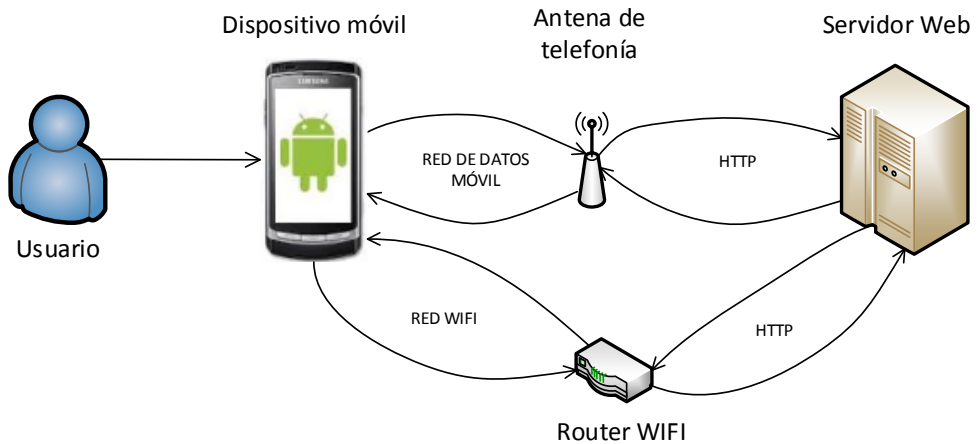


Ilustración 17 Diagrama de despliegue

Configuración del servidor

Para el correcto funcionamiento de las descargas en la aplicación, la estructura del servidor al que se dirige la aplicación, debe seguir el diagrama de la Ilustración 18.

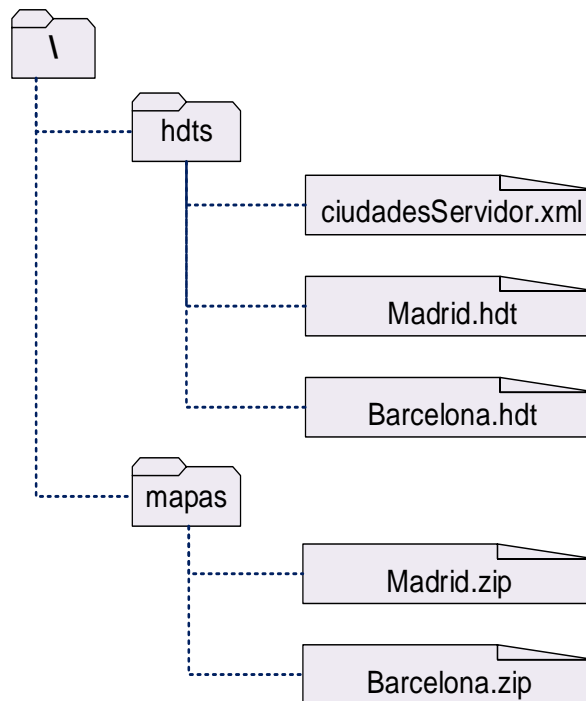


Ilustración 18 Estructura de directorios del servidor

En la carpeta “hds”, deben situarse los archivos hdt disponibles para descargar por la aplicación. En esta carpeta también se encontrará el archivo “ciudadesServidor.xml” que

describirá los ficheros HDT disponibles. Su estructura se explica en el capítulo 4.4.3.1 más adelante.

En la carpeta “mapas”, se situarán los mapas de las ciudades disponibles para su descarga y posterior consulta en modo offline, también explicados en el capítulo 4.4.4.2 más adelante.

4.3.1.2. Arquitectura lógica

En Android se utiliza el patrón de arquitectura Model View Controller (MVC) cuya principal ventaja consiste en separar los datos de una aplicación, la interfaz de usuario y la lógica de negocios en tres componentes distintos que se relacionarán entre sí.

Aunque la implementación nativa de Android gestiona parte de Controller de la vista, además de poseer capacidades de gestión de eventos. Por ello el patrón estándar de las aplicaciones se basa en un ejemplo más concreto de MVC, sin la parte del Controller gestionado por el SO. Este patrón es el Model View Presenter:

- **View:** Incluye las interfaces de usuario (Actividades, recursos, layouts, etc...)
- **Presenter:** Incluye nuestras clases de negocio, hacen de puente entre la vista y el modelo, incluyendo clases con métodos que alberguen la lógica de nuestra aplicación. Siempre los artefactos de la capa View invocarán a las clases albergadas en esta capa, nunca accederán al modelo directamente.
- **Model:** incluyen la definición lógica del modelo de datos con el que vamos a trabajar.

Además de este patrón de arquitectura, en este proyecto se van a utilizar otras librerías:

- **Java HDT library:** Es la librería que nos va a permitir realizar consultas HDT y ha sido proporcionada por el grupo DataWeb de la Universidad de Valladolid [29]. Esta librería interactúa con la capa de arquitectura del modelo.
- **Nutiteq SDK Maps:** Esta librería proporciona la posibilidad de visualizar mapas almacenados localmente en el dispositivo, se ha utilizado la versión con licencia de evaluación. Pertence a la empresa Nutiteq [30]. Esta librería interactúa con la capa de vista de la arquitectura.
- **Universal Image Loader:** Librería que permite gestionar la descarga de imágenes de recursos de Internet, manteniendo una caché en el dispositivo para minimizar el consumo de datos y gestionando la memoria para evitar leaks de memoria en el tratamiento de imágenes [31]. Esta librería interactúa con la capa de vista de la arquitectura.

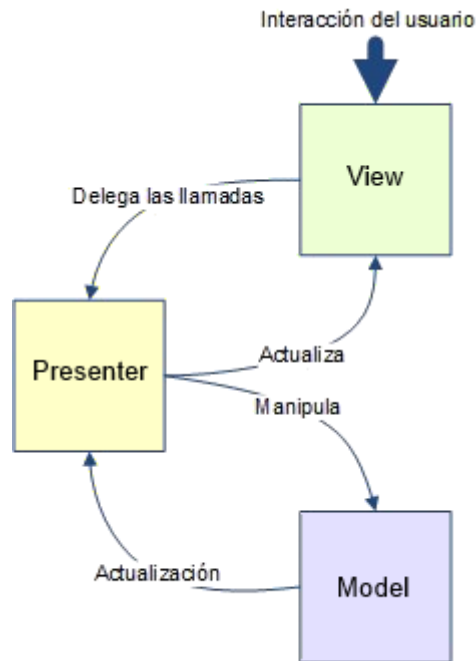


Ilustración 19 Arquitectura lógica

4.3.2. Diagrama de paquetes

En la Ilustración 20 vemos como se implementa el diagrama de paquetes siguiendo el patrón Model-View-Presenter.

Los paquetes *org.hdtourist.view* y *org.hdtourist.view.utils* forman parte de la vista. *org.hdtourist.presenter* pertenece a la capa del Presentador y *org.hdtourist.model*, *org.hdtourist.model.services* y *org.hdtourist.model.services.xml* pertenecen a la capa del modelo.

En la ilustración también se pueden ver la relación de estos paquetes con las librerías explicadas en el apartado anterior: *hdt-lib.jar*, *nutiteq-3d-sdk-2.3.1.jar* y *universal-image-loader-1.9.1.jar*, pertenecientes a *Java HDT Library*, *Nutiteq SDK Maps*, y *Universal Image Loader* respectivamente.

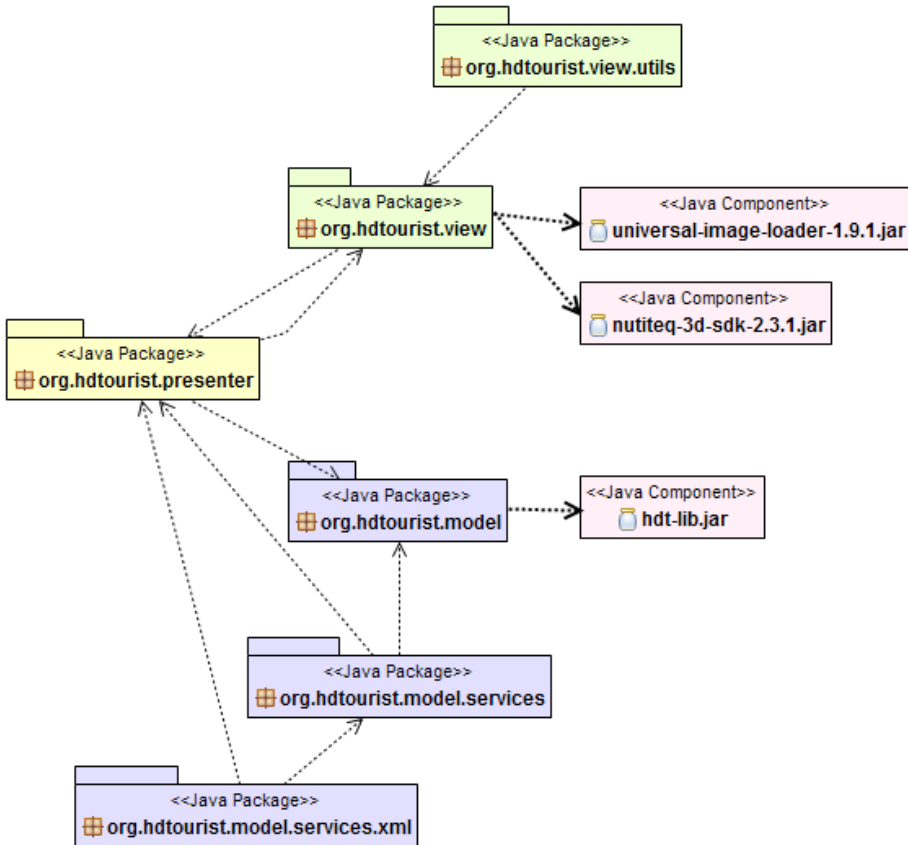


Ilustración 20 Diagrama de paquetes

4.3.3. Diagrama de clases

Para facilitar la comprensión de los diagramas de clase que presentaremos a continuación, describiremos las clases y conceptos básicos del paquete de Android:

- **Activity:** es una clase del paquete de Android cuya finalidad es mostrar los elementos de la interfaz de usuario que se implementan como **View** (elementos de la vista) y se describen en archivos XML. La clase Activity se amplía mediante clases de usuario que generan las pantallas de la aplicación.
- **Fragment:** representa una porción de una interfaz de usuario de una actividad. Se pueden combinar múltiples Fragments en una única actividad.
- **Main Thread y tareas asíncronas:** Cuando lanzamos una aplicación en el dispositivo, el sistema operativo Android crea un hilo conocido como Main Thread. Este hilo principal es el encargado de soportar la carga de procesamiento que conllevan las tareas principales y los componentes que forman parte de la interfaz de usuario recogidas en la aplicación. En el momento de que cualquier tarea del Main Thread bloquee la ejecución durante más de un tiempo pequeño (5 segundos), generará un error grave que cerrará la aplicación. Por regla general los componentes de la vista no pueden ser

modificados por hilos que no sean el Main Thread. Por estos motivos es necesario ejecutar tareas de forma asíncrona. En este proyecto utilizaremos la clase **AsyncTask**, que permite ejecutar una tarea en un nuevo hilo diferente al Main Thread y al terminar ejecutar en el Main Thread las actualizaciones oportunas.

A continuación describiremos el diagrama de clases de la aplicación. Lo dividiremos por funcionalidad de los casos de uso debido a su complejidad. Para simplificar los diagramas no incluiremos los atributos ni los métodos que componen las clases, estos y su descripción podrán ser consultados en el API en formato de javadoc entregado junto a la versión digital de esta memoria.

4.3.3.1. Diagrama de clases: Visualización de información de una ciudad.

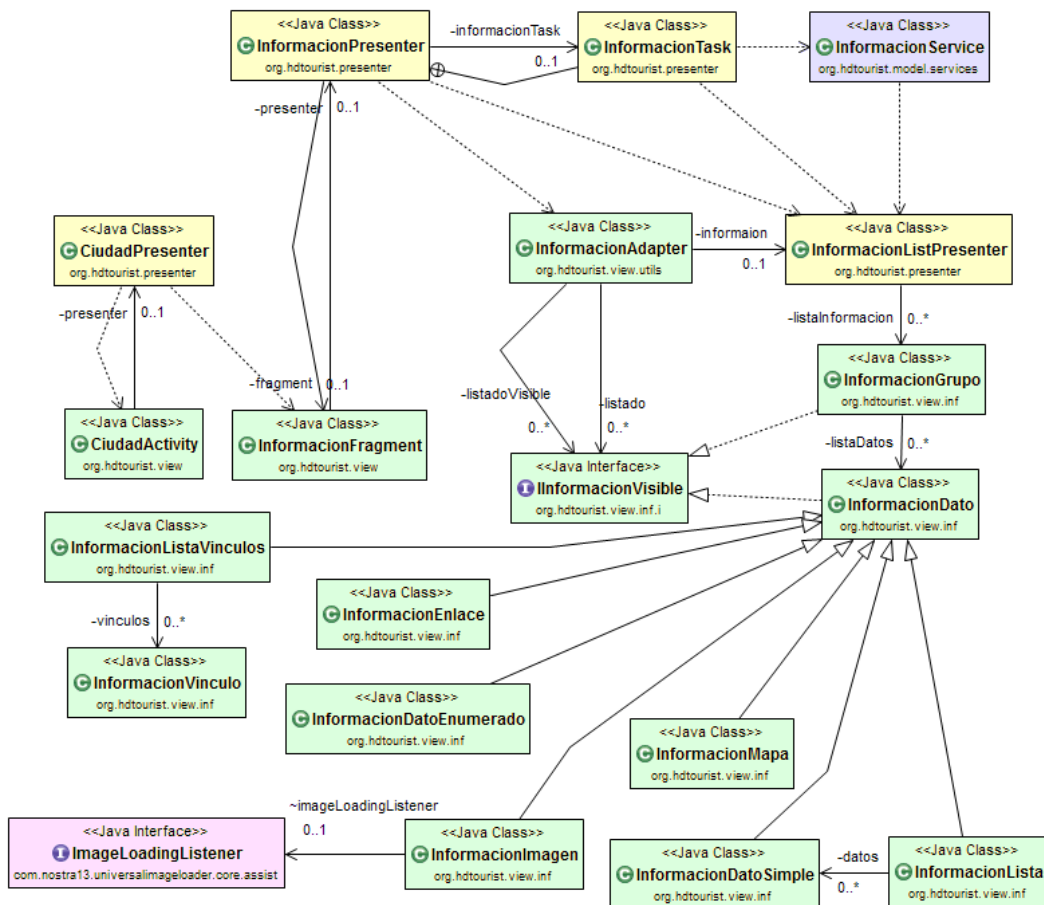


Ilustración 21 Diagrama de clases: Visualización de información de una ciudad

CiudadActivity representa una Activity que se compone de un Fragment de la clase **InformacionFragment**, ambos de la capa de vista.

InformacionPresenter de la capa de presentación, gestiona los Fragment que se muestran en cada categoría de información. Cada vez que se muestra una categoría de información, **InformacionPresenter** se encarga de ejecutar la tarea asíncrona **InformacionTask**, que se comunica con el modelo para realizar las consultas oportunas en el fichero HDT y obtener el objeto **InformacionListPresenter** que es el intermediario para intercambiar la información entre la capa de Presentación y el modelo.

InformacionFragment, contiene un elemento **ListView**, que muestra la lista de información consultada contenida en **InformacionListPresenter**. Esta lista está manejada por el Adaptador **InformacionAdapter** a través de **InformacionPresenter** y es encapsulada en grupos de información **InformacionGrupo** y este a su vez de datos **InformacionDato**. Ambos, grupos y datos, implementan la interfaz **InformacionVisible**, que se encarga de establecer la comunicación para devolver la parte visible (o **View**) del elemento de la lista en cuestión.

Los datos pueden ser de varios tipos, **InformacionListaVinculos**, **InformacionVinculo**, **InformaciónEnlace**, **InformacionLista**, **InformacionImagen**, **InformacionDatoSimple**, **InformacionDatoEnumerado** o **InformacionMapa**. Cada uno de ellos generará un elemento de la lista diferente (o **View**).

En el caso de **InformacionImagen**, para generar la vista utiliza la librería **Universal Loader Image**, para cachear las imágenes y permitir minimizar la necesidad de conexión a la red móvil.

4.3.3.2. Diagrama de clases: Lectura de información HDT

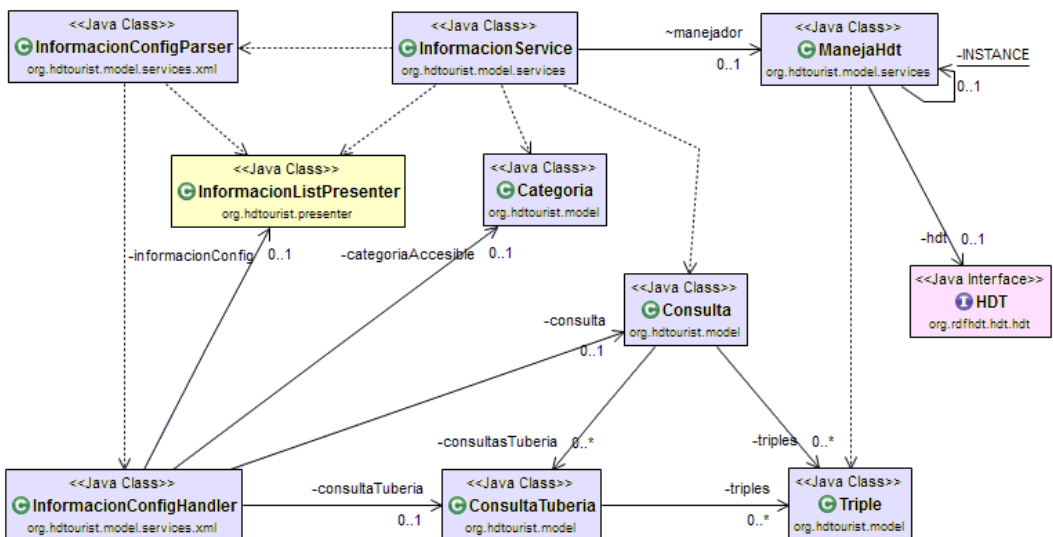


Ilustración 22 Diagrama de clases: Lectura de información HDT

InformacionService se encarga de obtener la información de una categoría, realizando las consultas a través de la clase **ManejaHdt**, que a su vez utiliza la librería **Java HDT** para consultar al fichero HDT.

Las consultas, se obtienen a través de ficheros XML de configuración leídos por **InformacionConfigParser** y **InformacionConfigHandler** y se traducen en listas de triples con un orden determinado que ejecuta **ManejaHdt**. A través de estas clases,

InformacionConfigParser y InformacionConfigHandler también se lee la estructura de la información a generar e **InformacionListPresenter** se encarga de traducirla a la vista después de haberse obtenido la información HDT.

4.3.3.3. Diagrama de clases: Listas de ciudades

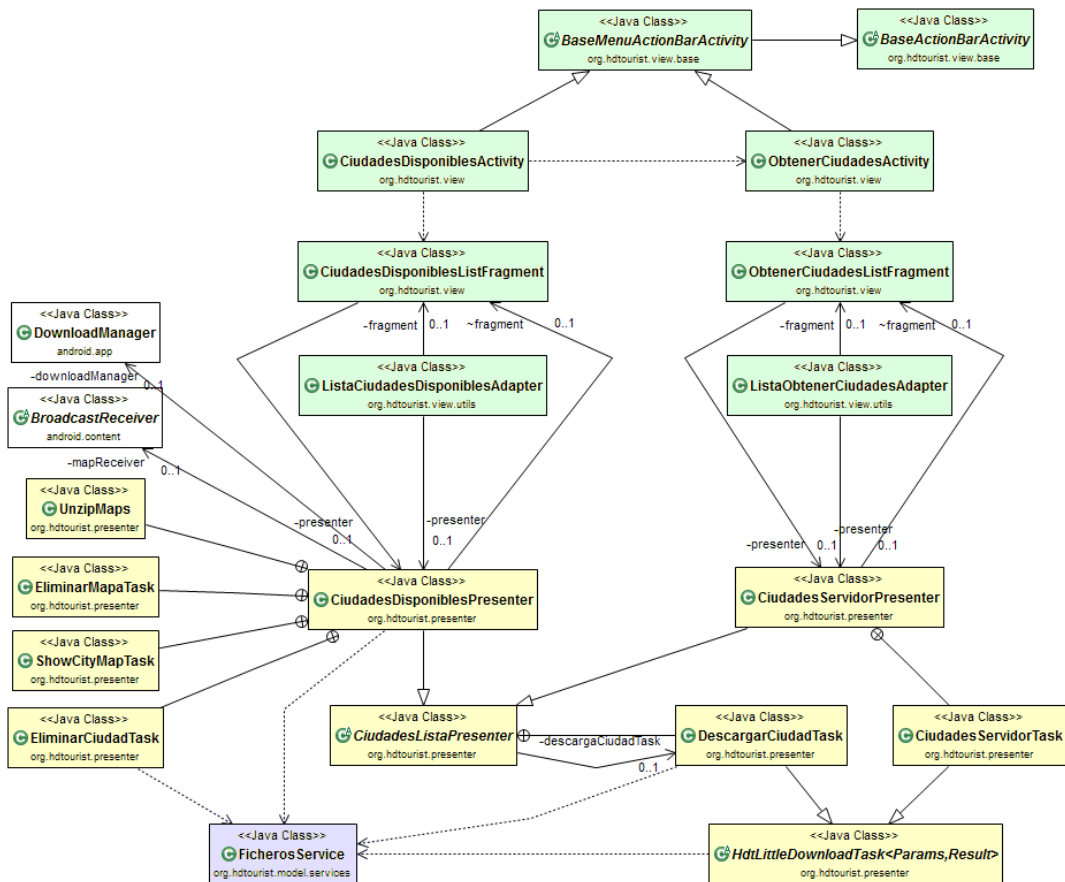


Ilustración 23 Diagrama de clases: Listas de ciudades

En la aplicación hay dos listas de información:

- Ciudades disponibles mostrada por las clases: **CiudadesDisponiblesActivity**, **CiudadesDisponiblesListFragment** y **ListaCiudadesDisponiblesAdapter**. **CiudadesDisponiblesPresenter**, gestiona las ciudades mostradas en la vista, estas ciudades se obtienen de archivos de configuración localizados en la memoria del dispositivo a través del modelo **FicherosService**. A su vez, **CiudadesDisponiblesPresenter** gestiona las tareas asíncronas de descarga y descompresión de mapas, eliminación de los mapas, eliminación de la ciudad o la obtención de los puntos geolocalizados para mostrar posteriormente un mapa con todos ellos.
- Ciudades en el servidor que se pueden descargar: **ObtenerCiudadesActivity**, **ObtenerCiudadesListFragment** y **ListaCiudadesAdapter**. Gestionadas por

CiudadesServidorPresenter, que gestiona la lista de ciudades con una tarea asíncrona **CiudadesServidorTask** para obtener la lista de ciudades en el servidor. Esta lista se carga a través de un fichero de configuración disponible en el servidor, por esta misma razón es necesaria la implementación de la tarea asíncrona.

Las dos Activities de la vista, CiudadesDisponiblesActivity y ObtenerCiudadesActivity descienden de **BaseMenuActionBarActivity** que gestiona el menú común para estas dos pantallas.

Ambas clases de la capa de presentación: CiudadesDisponiblesPresenter y CiudadesServidorPresenter, extienden la clase abstracta **CiudadesListaPresenter** que implementa una tarea asíncrona para la descarga de ficheros HDT: **DescargaCiudadTask**. Esta tarea, junto a CiudadesServidorTask extiende la tarea **HDTLittleDownloadTask**, puesto que ambas requieren descargar ficheros ligeros del servidor.

Para la descarga de ficheros pesados, es necesaria la utilización de las clases **DownloadManager** y **BroadCastReceiver** de la librería de Android, ya que las descargas pesadas se pueden demorar en el tiempo y de no implementarse mediante la clase DownloadManager o mediante un servicio, la gestión de memoria de Android puede terminar la descarga antes de que esta concluya.

4.3.4. Diagramas de secuencia

4.3.4.1. Diagrama de secuencia: Inicio de la aplicación

Al iniciar la aplicación se comprueba si los ficheros de configuración están en la memoria de dispositivo. Si no están copiados, se inicia una tarea asíncrona que copia los ficheros de configuración a la memoria. Cuando finaliza, se inicia la pantalla de ciudades disponibles de la aplicación. Véase la figura en la página siguiente.

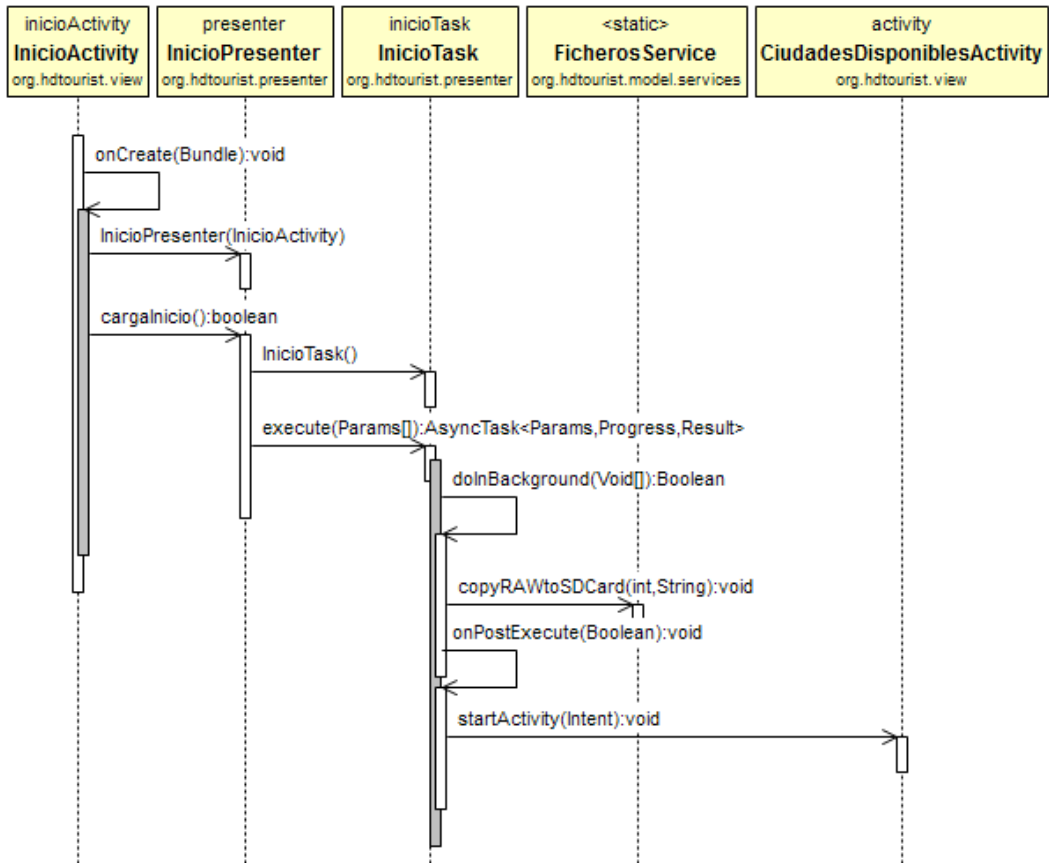


Ilustración 24 Diagrama de secuencia: Inicio de la aplicación

4.3.4.2. Diagrama de secuencia: Lista de ciudades disponibles

Al abrirse la pantalla de ciudades disponibles se crea el Fragment que tiene la lista de ciudades, esta lista se genera leyendo de la memoria del dispositivo un archivo de configuración que se actualiza cada vez que se borra o descarga una ciudad. Véase la figura en la página siguiente.

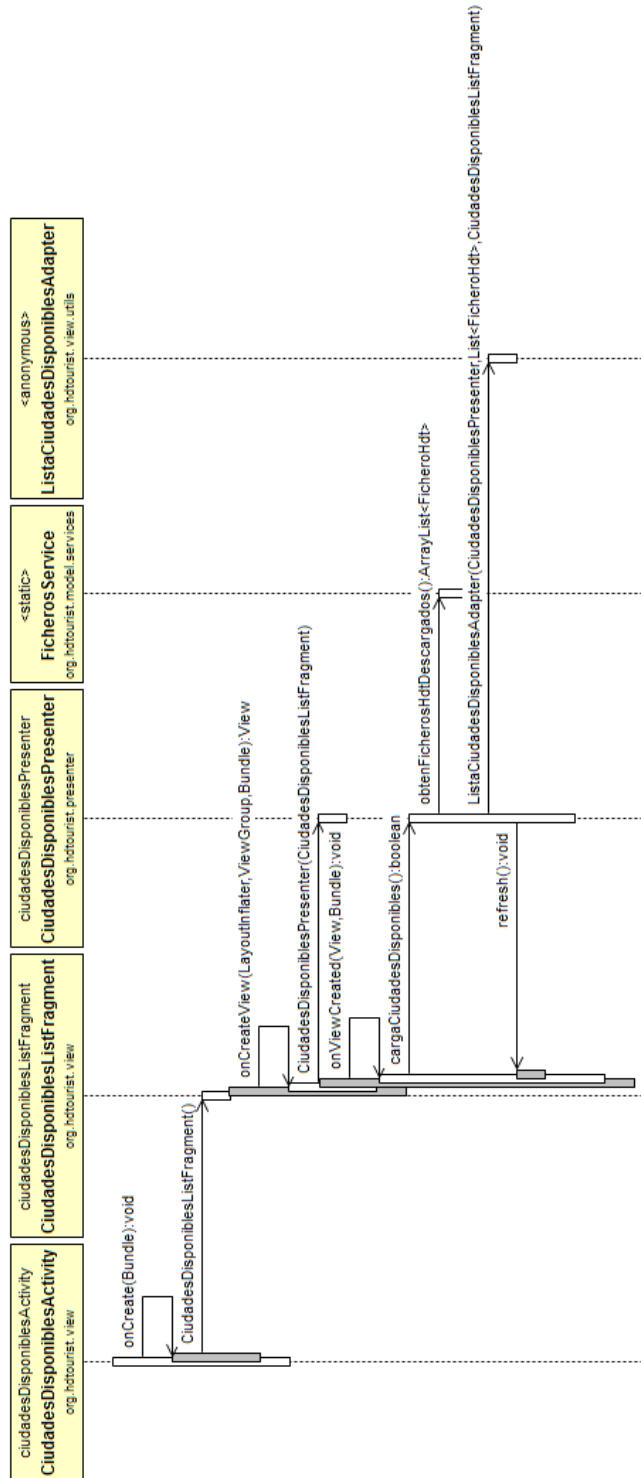


Ilustración 25 Diagrama de secuencia: Lista de ciudades disponibles

4.3.4.3. Diagrama de secuencia: Lista de ciudades para descargar

Al abrir la ventana de ciudades para descargar se ejecuta una tarea asíncrona que descarga de Internet un fichero que indica que ciudades hay en el servidor disponibles para descargar su información. Véase la figura en la página siguiente.

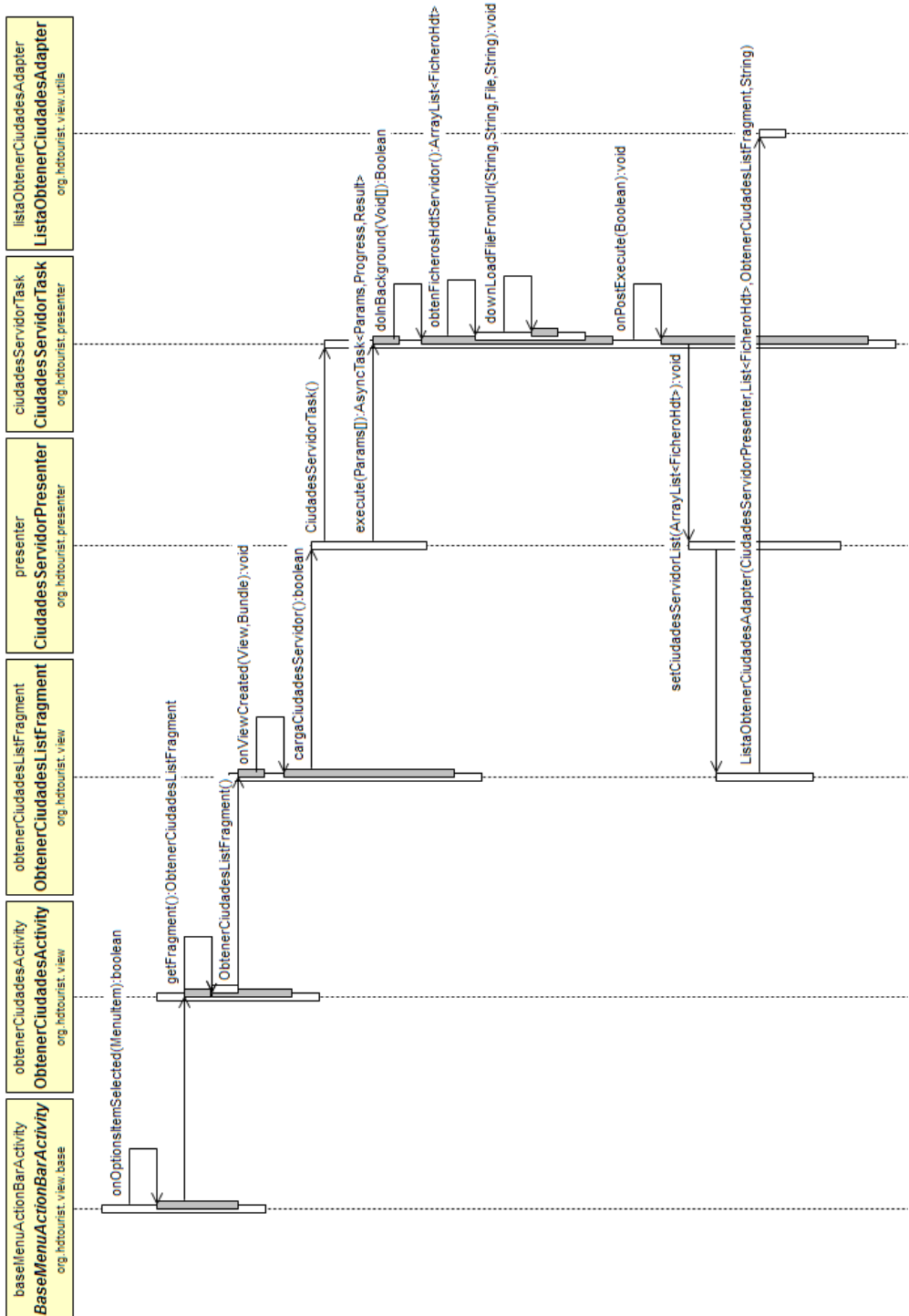


Ilustración 26 Diagrama de secuencia: Lista de ciudades para descargar

4.3.4.4. Diagrama de secuencia: Seleccionar una ciudad para descargar su información

Desde la lista de ciudades disponibles se detecta la acción del usuario al seleccionar una ciudad. Al detectar el evento, se ejecuta una tarea asíncrona que descarga el fichero HDT correspondiente del servidor. Véase la figura en la página siguiente.

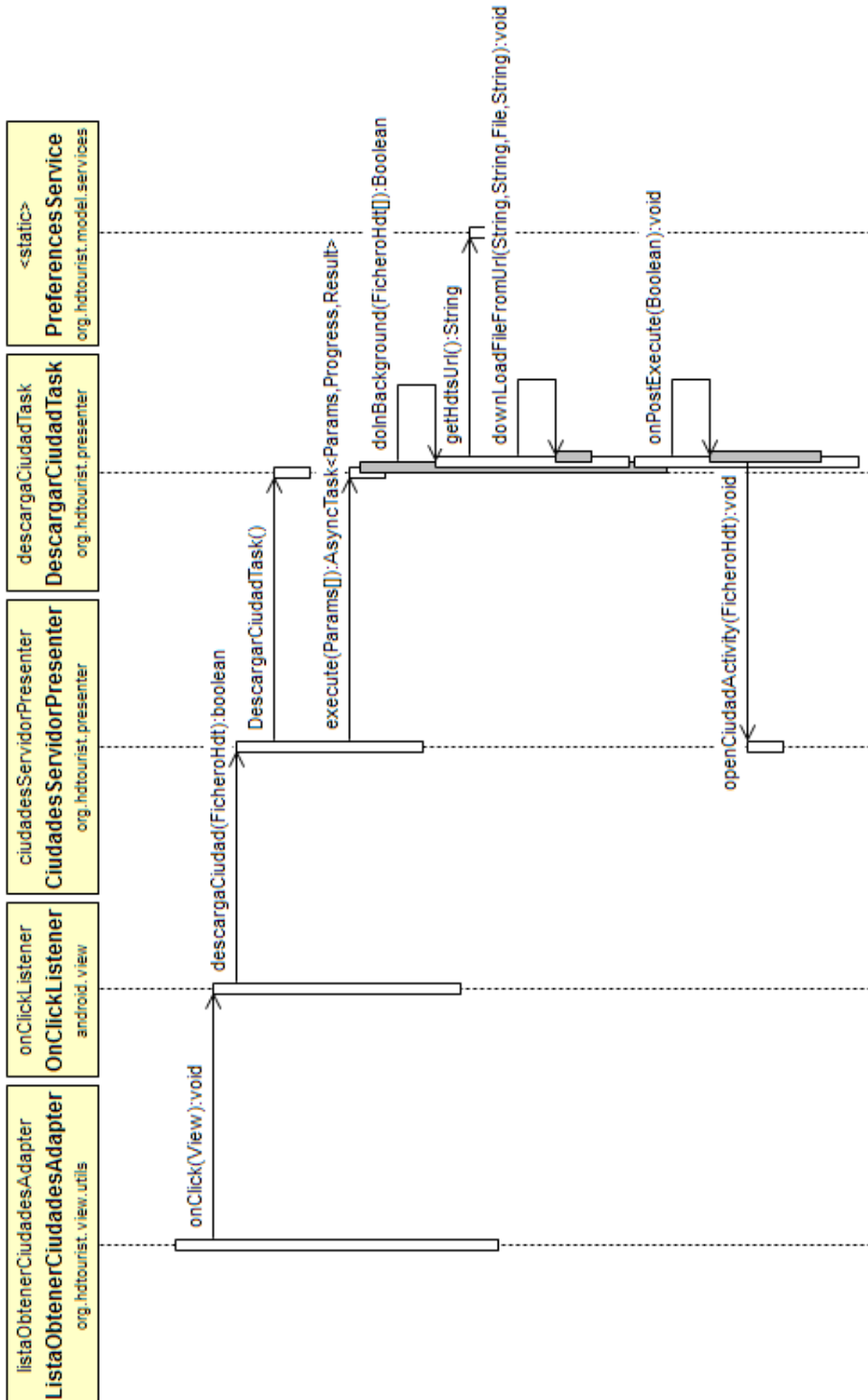


Ilustración 27 Diagrama de secuencia: Descargar información de una ciudad

4.3.4.5. Diagrama de secuencia: Acceder a la información de una ciudad

Al acceder a una ciudad se carga por defecto la información de la primera categoría de la ciudad en un fragment. Se ejecuta una tarea asíncrona para realizar las consultas de HDT necesarias para obtener la información que genere el contenido del fragment. Véase la figura en la página siguiente.

4.4. Implementación

La implementación del sistema consiste en la traducción del diseño realizado en el capítulo anterior a código de un lenguaje de programación.

4.4.1. Versiones de Android compatibles con HDTourist.

Debido a las múltiples versiones de Android explicadas en el apartado 3.3.2.2 más atrás, ha sido necesario elegir la versión mínima de Android que soportará nuestra aplicación y la versión más moderna con la que HDTourist será compatible.

Para que nuestra aplicación funcione en el mayor número de dispositivos posible, debemos elegir la versión mínima que permita implementar las características requeridas por los requisitos. Esta versión será *Gingerbread v2.3 API 9*, ya que en esta versión implementa funcionalidades básicas y da mayor soporte a las distintas resoluciones de pantalla, y como hemos podido ver en el punto 3.3.2.1 más atrás el porcentaje de utilización de versiones anteriores es mínimo.

Para que nuestra versión tenga mayor compatibilidad, actualizaremos el SDK de Android cada vez que salga una nueva versión.

4.4.2. Entorno de desarrollo

El objetivo principal del proyecto es desarrollar una aplicación para dispositivos móviles Android, por lo que esta deberá ser desarrollada en este lenguaje.

La versión de la SDK de Android se ha ido actualizando a lo largo del proyecto según aparecían nuevas versiones, la última versión actualizada ha sido 4.4.2 (API 19).

El software empleado para la implementación, ha sido el entorno de desarrollo *Android Developer Tools* en la versión 22.0, que incluye la plataforma de Eclipse 4.2.1 (Juno). Pero con la actualización de la última versión del SDK de Android, ha sido necesario actualizar el IDE a Eclipse Luna 4.4.0 (Kepler) ya que el anterior dejó de tener compatibilidad con dicho SDK.

4.4.3. Ficheros XML

XML, siglas en inglés de Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML, XAML.

Hemos utilizado ficheros XML para configurar diversos aspectos de la aplicación:

- Para almacenar la información básica de los archivos HDT descargados y de los archivos HDT disponibles en el servidor.

- Para configurar las categorías disponibles para las ciudades, accesibles mediante el menú lateral.
- Para configurar la información que se mostrará en cada categoría, así como su forma de obtener dicha información. Con este tipo de ficheros, también configuramos la forma de realizar la búsqueda de información dentro de una ciudad, de obtener los puntos que se mostrarán en el mapa completo y de obtener la dirección de la imagen de la ciudad que podrá verse en la lista de ciudades disponibles.

4.4.3.1. Archivo xml de configuración de ficheros

Como hemos dicho, este archivo xml almacena la información básica de los ficheros HDT. En el servidor, el archivo estará disponible en la ruta configurada:

RUTA_DE_SERVIDOR_CONFIGURADA/hdts/ciudadesServidor.xml

Y su contenido será similar al siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<ficheros>
  <fichero>
    <nombre>Madrid.hdt</nombre>
    <uri>http://dbpedia.org/resource/Madrid</uri>
    <nombreCiudad>Madrid</nombreCiudad>
  </fichero>
  <fichero>
    <nombre>Valladolid.hdt</nombre>
    <uri>http://dbpedia.org/resource/Valladolid</uri>
    <nombreCiudad>Valladolid</nombreCiudad>
  </fichero>
</ficheros>
```

Este fichero indicaría que hay dos ciudades disponibles para descargar desde el servidor, Madrid y Valladolid. Los nombres de los ficheros HDT serían respectivamente Madrid.hdt y Valladolid.hdt. También se indica la uri de los recursos correspondientes. Esta uri servirá para que la aplicación realice las consultas de información relativas a la ciudad correspondiente.

En el dispositivo, se mantiene un archivo xml con la misma estructura para guardar la información de los ficheros disponibles y almacenados en la memoria del terminal. Este archivo se actualiza cada vez que se descarga o elimina una ciudad:

RUTA_DE_APLICACION_EN_DISPOSITIVO/conf/ciudadesDescargadas.xml

4.4.3.2. Archivo xml de configuración de categorías de la ciudad

Este archivo configura las categorías de información que estarán disponibles para las ciudades descargadas. Al realizar la instalación de la aplicación se crea en la siguiente ruta:

RUTA_DE_APLICACION_EN_DISPOSITIVO/conf/conf_ciudad.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<categorias>
  <categoria>
    <nombre>Descripcion</nombre>
    <fichero>conf_ciudad_descripcion</fichero>
    <imagen>info</imagen>
  </categoria>
  <categoria>
    <nombre>Lugares de interés</nombre>
    <fichero>conf_ciudad_lugares</fichero>
    <imagen>turista</imagen>
  </categoria>
</categorias>
```

Para cada categoría se configura su nombre, el fichero de configuración que indicará la información que se mostrará en esa categoría y la forma de obtenerla y la imagen que se mostrará en el menú.

El contenido de este archivo indica que se mostrarían dos categorías para las ciudades, *Descripción* y *Lugares de interés*. Sus ficheros de configuración se encontrarán en:

RUTA_DE_APLICACION_EN_DISPOSITIVO/conf/conf_ciudad_descripcion.xml

y

RUTA_DE_APLICACION_EN_DISPOSITIVO/conf/conf_ciudad_lugares.xml

Este archivo puede ser modificado para añadir o eliminar categorías.

4.4.3.3. Archivo xml de configuración de información de categoría

En la aplicación existirá un archivo de configuración de información por cada categoría existente. Estos archivos configuran la información que se mostrará por pantalla y las consultas a realizar en el fichero HDT para obtenerla.

```
<?xml version="1.0" encoding="UTF-8"?>
<categoria id="Descripcion">
```



```

<agrupacionInformacion>
  <nombreAgrupacion>Informacion</nombreAgrupacion>
  <listaDatos>
    <dato>
      <etiqueta>Pais</etiqueta>
      <consultas type="OR">
        <tuberia>
          <consulta resultado="objeto">
            <sujeito>${URI}</sujeito>
            <predicado> http://dbpedia.org/ontology/country
            </predicado>
          </consulta>
          <consulta resultado="objeto"
encadenaAnterior="sujeito">
            <predicado>
              http://www.w3.org/2000/01/rdfschema#label
            </predicado>
          </consulta>
        </tuberia>
      </consultas>
    </dato>
    <dato type="imagen">
      <etiqueta>Imagen</etiqueta>
      <consultas type="OR">
        <consulta resultado="objeto">
          <sujeito>${URI}</sujeito>
          <predicado> http://xmlns.com/foaf/0.1/depiction
          </predicado>
        </consulta>
      </consultas>
    </dato>
  </listaDatos>
</agrupacionInformacion>

```

```
</categoria>
```

El archivo anterior define una pantalla de información con dos consultas, una para obtener el país, y otra para obtener la url de una imagen de la ciudad, ambas dentro de una agrupación de información, cuyo nombre, que se mostrará como un título en la pantalla es “Información”

La pantalla generada con esta configuración para la ciudad de Madrid sería la siguiente:



Ilustración 29 Ficheros xml de configuración de información: pantalla generada.

Estructura

El primer elemento o elemento raíz del fichero es “*categoria*” que contiene un atributo *id* que indica la categoría a la que pertenece el fichero.

Dentro del elemento “*categoría*” puede haber varios elementos “*agrupacionInformacion*”, que almacenan distintos tipos de información. Por ejemplo, en una categoría *lugares* puede haber varios grupos de información: *plazas*, *monumentos*, *museos*, *palacios*... etc. Dentro de este elemento, está “*nombreAgrupación*”, que almacena el nombre que se mostrará la pantalla a modo título para separar los grupos de información.

Dentro de cada agrupación, habrá un elemento “*listaDatos*” que contiene uno o varios elementos “*dato*”. Estos elementos “*dato*” son los datos que se mostrarán finalmente en la pantalla del dispositivo. Un elemento “*dato*” puede tener el atributo “*type*”, cuyos posibles valores son:

- *lista*: Indica que se mostrarán todos los resultados de la consulta en forma de lista.
- *listaVinculos*: Indica que se mostrarán todos los resultados de la consulta en forma de lista y que estos serán vínculos a otras pantallas de información. Siempre que haya vínculos, debe indicarse la categoría de esos vínculos mediante el elemento “*accesible*” explicado más abajo.
- *mapa*: Indica que el resultado de la consulta serán coordenadas geográficas. Por tanto, la aplicación permitirá acceder a un mapa con las coordenadas señaladas.
- *imagen*: Indica que el resultado de la consulta es una url de Internet de la que podrá descargarse una imagen.
- *imagen_text*: Siempre que haya un dato de este tipo, tiene que ir precedido de un dato de tipo *imagen*. Indica que es el texto que describe la imagen anterior, se mostrará solo cuando se muestre la imagen.
- *enlace*: Indica que es un enlace a un recurso de Internet. Pulsando sobre él, se abrirá el navegador del dispositivo con la url resultante de la consulta.

Dentro del elemento “*dato*”, habrá un elemento “*etiqueta*” que contendrá el texto mostrado por pantalla para identificar el dato. Si el dato tiene el atributo “*onlyText*” con valor *true*, indicará que la etiqueta no será visible.

Dentro del elemento *dato* debe haber un elemento “*consultas*” que defina la consulta de la cual se obtendrá la información. Dentro de este elemento puede haber uno o varios elementos que definirán la combinación de consultas, y el atributo “*type*” de “*consultas*”, indicará el tipo de join a realizar “*OR*” o “*AND*”. Estas consultas pueden ser elementos “*consulta*” o elementos “*tubería*.”

El elemento “*consulta*” tiene un atributo “*resultado*” que puede tener el valor: “*objeto*”, “*sujeto*” o “*todos*” e indica cual será el resultado de la consulta, el objeto, el sujeto o la combinación de sujeto-predicado-objeto. El elemento “*consulta*”, tendrá al menos un elemento “*sujeto*”, “*predicado*” o “*objeto*” para indicar la consulta a realizar.

El elemento “*tubería*” describe una combinación de consultas encadenadas. Cada una de ellas será un elemento “*consulta*” descrito anteriormente. En esta situación, el elemento “*consulta*” puede tener un atributo más “*encadenaAnterior*” que indica en qué posición de la consulta actual se colocará el resultado de la consulta anterior: “*objeto*” o “*sujeto*”.

Cuando un elemento es un vínculo, el elemento “*dato*” tiene un elemento “*accesible*” que permite indicar mediante el elemento “*ficheroconfig*” el nombre del fichero que configurará la información mostrada en la pantalla a la que el vínculo accede.

Ejemplo

Siguiendo la configuración de la consulta anterior que obtiene el nombre del país de una ciudad:

```
<consultas type="OR">
  <tuberia>
    <consulta resultado="objeto">
      <sujeito>${URI}</sujeito>
      <predicado> http://dbpedia.org/ontology/country
    </predicado>
    </consulta>
    <consulta resultado="objeto" encadenaAnterior="sujeito">
      <predicado>
        http://www.w3.org/2000/01/rdfschema#label
      </predicado>
    </consulta>
  </tuberia>
</consultas>
```

La consulta sparql generada, para la ciudad de Madrid, sería la siguiente:

```
select ?nombrePais
where {
  <http://dbpedia.org/resource/Madrid>
  <http://dbpedia.org/ontology/country> ?uriPais .
  ?uriPais
  <http://www.w3.org/2000/01/rdf-schema#label> ?nombrePais .
}
```

Esta consulta obtendría los siguientes resultados:

nombrePais
"西班牙"@zh
"Spain"@en
"Spanien"@de
"España"@es
"Spanje"@nl
"スペイン"@ja
"Espagne"@fr
"Испания"@ru

"Espanha"@pt
"Spagna"@it
"Hiszpania"@pl
"Spanien"@sv

Tabla 35 Resultados de la consulta sparql para obtener el país

De estos resultados, la aplicación obtendría el correspondiente al idioma del dispositivo, o en caso de no existir el resultado, se tomaría por defecto el resultado del idioma inglés.

Ficheros de información adicionales

Existen tres ficheros de configuración de información que siguen esta misma estructura pero no configuran de la misma forma la pantalla que muestran.

Estos ficheros, definen consultas para obtener información concreta que ayuda a generar parte de la pantalla:

- *conf_search.xml*: Define las consultas a realizar para generar los resultados cuando se realiza una búsqueda de información dentro de una ciudad:
- *conf_mapa.xml*: Define las consultas para obtener los puntos que se mostrarán en el mapa completo de una ciudad. Define las siguientes consultas:
 - *conf_obtener_uri_puntos*: consulta para obtener todos los puntos geolocalizados de la ciudad.
 - *conf_obtener_nombre_punto*: consulta para obtener el nombre del punto geolocalizado.
 - *conf_obtener_posicion*: consulta para obtener el punto geolocalizado.
 - *conf_obtener_centro_ciudad*: consulta para obtener el centro geolocalizado de la ciudad para centrar el mapa.
- *conf_obtener_imagen_ciudad.xml*: Define la consulta necesaria para obtener la dirección de la imagen de la ciudad que podrá verse en la lista de ciudades disponibles.

En todas las consultas realizadas se pueden utilizar ciertas palabras clave:

- $\{\text{TEXTO}\}$ Se sustituye por el texto buscado, para este caso concreto el atributo *type* del elemento consultas debe ser *SEARCH*, que equivale al valor *OR*, con la peculiaridad de que no se hace una búsqueda por sujeto, predicado u objeto completo, si no que se busca el texto dentro de los mismos.
- $\{\text{NOMBRE}\}$ Se sustituye por el nombre del recurso consultado
- $\{\text{URI}\}$ Se sustituye por la uri del recurso consultado

4.4.4. Librerías utilizadas

4.4.4.1. Consultas HDT

Para realizar las consultas al fichero HDT ha sido necesario utilizar la librería *Java HDT Library*. Esta librería ha sido proporcionada por el grupo DataWeb de la Universidad de Valladolid [29]. Esta librería interactúa con la capa de arquitectura del modelo y permite obtener una lista de triples coincidentes con el patrón sujeto-predicado-objeto pasado por parámetro.

4.4.4.2. Mapas Offline

La visualización de puntos geográficos es un requisito de la aplicación, así como mantener la funcionalidad sin necesidad de conexión a Internet. Por ello, ha sido necesario buscar una forma de visualizar mapas de forma offline.

Se han barajado varias posibilidades:

- La utilización de Google Maps, una aplicación exterior, tanto para mapas con conexión como mapas sin conexión. Google Maps permite en algunas versiones descargar zonas para poder visualizarlas posteriormente sin conexión a Internet. Esta solución no se ha utilizado debido a las diferencias de comportamiento entre diferentes versiones de la aplicación Google Maps y a que no todos los mapas están disponibles en modo offline. Otra desventaja, es que la gestión de mapas disponibles de manera offline, debe hacerse desde la propia aplicación de Google Maps, no pudiendo configurar que mapas visualizar offline desde HdTourist.
- La utilización de otras aplicaciones, como MapsWithMe [32]. Es una aplicación que requiere ser instalada con independencia de HdTourist. Proporciona una librería para la comunicación entre aplicaciones. No se ha elegido esta solución por la necesidad de instalar una segunda aplicación.
- Finalmente la solución elegida ha sido la utilización de una librería más versátil como la que proporciona *Nutiteq SDK Maps* [30].

Nutiteq SDK Maps

Nutiteq SDK es un kit de herramientas de avanzadas para el desarrollo de aplicaciones en Android. Permite la visualización de mapas en 2D y 3D, independientemente de las fuentes de datos.

Para integrarlo con HdTourist, ha sido necesario añadir la librería de nutiteq a la aplicación y realizar la llamada al api indicando la ruta de almacenamiento de los mapas.

Los mapas han sido generados con la aplicación gMapMaker [33], que genera carpetas según el zoom configurado y dentro de cada carpeta genera las imágenes del mapa. Estas imágenes serán descargadas de forma comprimida desde el servidor configurado, la aplicación HdTourist los descomprimirá una vez descargados y colocará en el directorio correspondiente al cual accederá la librería de Nutiteq que los gestionará y mostrará de forma adecuada.

4.4.4.3. Imágenes

HdTourist puede mostrar imágenes que están alojadas en Internet. Para evitar el consumo excesivo de datos, se ha visto la necesidad de mantener una caché en disco de imágenes ya descargadas. De esta forma, se podrían ver las imágenes sin necesidad de descargarlas más de una vez, también se podrán visualizar imágenes descargadas cuando la aplicación tenía conexión a Internet cuando ya no se dispone de esta conexión. Para la gestión de esta caché en disco se ha utilizado la librería *Universal Image Loader* [31].

Capítulo 5. Pruebas

5.1. Introducción

El último paso realizado tras terminar la implementación del código fuente, es la realización de pruebas para asegurar el correcto funcionamiento de la aplicación. Se ha llevado a cabo una serie de casos de prueba diseñados para determinar si los requisitos establecidos se han cumplido total o parcialmente.

El desarrollo de las pruebas se ha llevado a cabo desde dos perspectivas diferentes, no excluyentes, si no complementarias:

- **Pruebas de caja blanca.** Son aquellas que se realizan sobre funciones internas de un módulo, de esta forma se comprueba la lógica interna del programa.
- **Pruebas de caja negra.** En estas pruebas no conocemos la implementación del código, sólo la interfaz. Se centran en lo que se espera de un módulo, por ello se denominan pruebas funcionales, y el probador se limita a suministrar datos de entrada y a estudiar la salida.

En el momento de detectar un error en una de estas pruebas, el procedimiento que se ha seguido ha sido el siguiente:

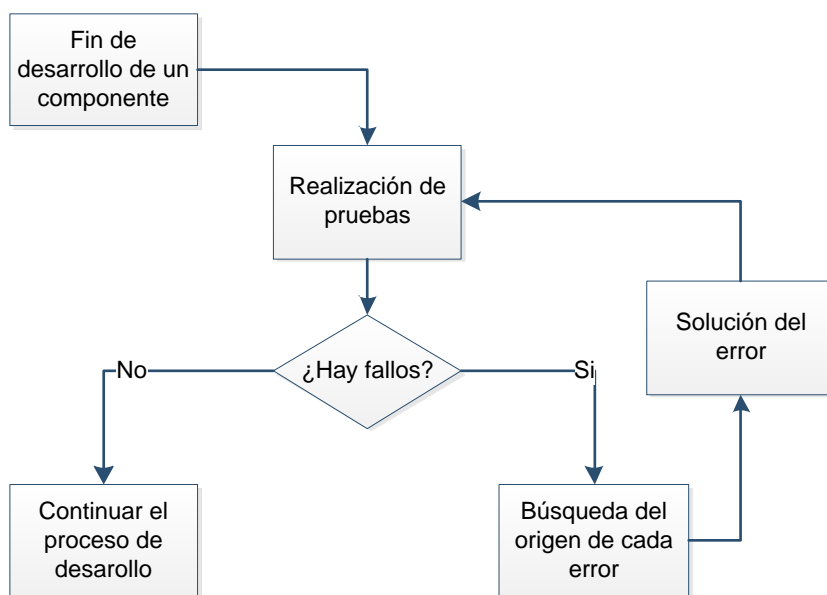


Ilustración 30 Ciclo de pruebas

5.1.1. Pruebas de caja blanca

Estas pruebas se han ido realizando manualmente a medida que desarrollaba el código, de forma que cada método, clase y la colaboración entre todas ellas han sido probadas.

5.1.2. Pruebas de caja negra

Este tipo de pruebas está enfocado a comprobar que los requisitos funcionales y no funcionales se han cumplido. Es decir, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos de un programa. No son una alternativa a las pruebas de caja blanca sino que se han de realizar las dos por separado y de forma complementaria, con el fin de detectar diferentes tipos de errores.

Durante el desarrollo y al terminar la implementación se han realizado las siguientes pruebas para comprobar los requisitos funcionales y no funcionales vistos en los apartados 4.2.4.1 y 4.2.4.2 más atrás.

Pasos	Resultado esperado	Resultado obtenido	
FRQ-01 Seleccionar una ciudad para ver su información.			
Al iniciarse la aplicación mostrará una lista de ciudades con información ya descargada y permitirá seleccionar una de ellas para acceder a su información			
1. Iniciar la aplicación.	El sistema muestra la pantalla de carga al inicio y posteriormente muestra de la lista de ciudades descargadas.	Se muestra la pantalla de inicio y posteriormente la lista de ciudades.	OK
1. Iniciar la aplicación. 2. Seleccionar una ciudad.	El sistema muestra la información básica de la ciudad y el acceso a las categorías.	Se muestra la información de la ciudad seleccionada.	OK
FRQ-02 Seleccionar de entre una lista de ciudades descargables			
La aplicación mostrará una lista de ciudades disponibles para descargar su información. Si dicha información ya se encuentra en el dispositivo, el sistema pedirá confirmación para actualizarla.			
1. Iniciar la aplicación. 2. Seleccionar descargar nuevas ciudades. 3. Seleccionar una ciudad no descargada.	El sistema pide confirmación para el acceso a los servicios de datos.	Se pide la confirmación al acceder a la pantalla de ciudades disponibles para descargar.	OK
1. Iniciar la aplicación. 2. Seleccionar descargar nuevas ciudades. 3. Seleccionar una ciudad ya descargada.	El sistema indica que la ciudad ya está disponible en el dispositivo y pregunta si debe actualizarla indicando el tamaño del archivo de actualización.	Se indica que la ciudad ya está descargada y se pregunta si se quiere actualizar.	OK

<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar descargar nuevas ciudades. 4. Seleccionar una ciudad no descargada. 5. Aceptar el mensaje de confirmación. 	<p>El sistema descargará la el archivo de la ciudad y procederá a abrirla al terminar.</p>	<p>Se descarga la ciudad en el dispositivo y se muestra su información al terminar el proceso.</p>	<p>OK</p>
FRQ-03 Buscar una ciudad para descargar su información			
<p>El sistema permitirá buscar textualmente por una ciudad, realizando sugerencias de ciudades encontradas. Una vez seleccionada, se pedirá confirmación de la información a descargar, indicando el tamaño de la información previa a la descarga. Si dicha información ya se encuentra en el dispositivo, el sistema pedirá confirmación para actualizarla</p>			
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar descargar nuevas ciudades. 3. Seleccionar el campo de búsqueda. 4. Empezar a introducir un nombre de ciudad. 	<p>El sistema mostrará una lista con las ciudades cuyo nombre coincide con el texto introducido.</p>	<p>Se filtran las ciudades disponibles coincidentes con el texto introducido.</p>	<p>OK</p>
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar descargar nuevas ciudades. 3. Seleccionar el campo de búsqueda. 4. Introducir el nombre de la ciudad existente. 	<p>El sistema mostrará la lista de ciudades coincidentes.</p>	<p>Se muestra la ciudad correspondiente al nombre introducido.</p>	<p>OK</p>
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar descargar nuevas ciudades. 3. Seleccionar el campo de búsqueda. 4. Introducir el nombre de la ciudad no existente. 	<p>El sistema indicará que no hay ciudades disponibles con ese nombre.</p>	<p>Se indica que no se han encontrado ciudades coincidentes con esa búsqueda.</p>	<p>OK</p>
FRQ-04 Visualizar información de una ciudad por categorías			
<p>El sistema permitirá visualizar información de una ciudad organizada a través de diferentes categorías. Cada categoría mostrará la información correspondiente de la ciudad de forma separada.</p>			
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Abrir el menú lateral. 	<p>El sistema muestra las categorías de información disponibles.</p>	<p>Se muestra un menú con las categorías disponibles.</p>	<p>OK</p>

<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Abrir el menú lateral. 4. Seleccionar una categoría. 	<p>El sistema muestra la información disponible de dicha categoría.</p>	<p>Se muestra la información de la ciudad y la categoría seleccionada.</p>	<p>OK</p>
FRQ-05 Abrir vínculos de objetos relacionados			
El sistema permitirá acceder a la información de los objetos mostrados con información disponible.			
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Abrir el menú lateral. 4. Seleccionar una categoría. 5. Seleccionar un punto de información con vínculo. 	<p>El sistema mostrará una nueva pantalla con información a cerca del punto de información seleccionado.</p>	<p>Se muestra la información sobre el vínculo seleccionado.</p>	<p>OK</p>
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Abrir el menú lateral. 4. Seleccionar una categoría. 5. Seleccionar un punto de información sin vínculo. 	<p>El sistema no hará nada si no es un dato con vínculo, ni mapa, ni un texto que no entre en la pantalla.</p>	<p>No se realiza ninguna acción. A excepción si el texto de información no entra en el campo, que la acción realizada es extender (o contraer) el campo.</p>	<p>OK</p>
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Abrir el menú lateral. 4. Seleccionar una categoría. 5. Seleccionar un punto de información con vínculo. 6. Volver a la pantalla anterior. 	<p>El sistema debe volver a la pantalla anterior que contenía el vínculo.</p>	<p>Se vuelve a la pantalla que contenía el vínculo seleccionado.</p>	<p>OK</p>
FRQ-06 Búsqueda de información			
La aplicación permitirá realizar búsquedas textuales devolviendo aquella información que concuerde con las palabras claves de búsqueda introducidas			
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Seleccionar la opción de búsqueda. 	<p>El sistema muestra un campo de texto para introducir los criterios de búsqueda.</p>	<p>Se muestra el campo de búsqueda para introducir el criterio de búsqueda.</p>	<p>OK</p>

<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Seleccionar la opción de búsqueda. 4. Introducir el texto buscado. 	El sistema muestra los resultados de la búsqueda.	Se muestran los resultados coincidentes con el término de búsqueda.	OK
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Seleccionar la opción de búsqueda. 4. Dejar vacío el campo de texto. 	El sistema sigue solicitando que se introduzca un criterio de búsqueda.	No se realiza ninguna acción y se sigue mostrando activo el campo de búsqueda.	OK
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Seleccionar la opción de búsqueda. 4. Introducir un criterio de búsqueda no coincidente con ninguna información. 	El sistema indicará que no se ha encontrado información.	Se indica que no se ha encontrado información coincidente.	OK
FRQ-07 Mostrar mapa			
La aplicación mostrará un mapa con todo lo geográficamente relacionado en un radio de búsqueda			
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Abrir el menú lateral. 4. Seleccionar una categoría. 5. Seleccionar un punto de información con mapa. 	El sistema mostrará una nueva pantalla de un mapa con el punto geolocalizado.	Se muestra el acceso al mapa y seleccionándolo muestra el punto geolocalizado.	OK
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Abrir el menú lateral. 4. Seleccionar una categoría. 5. Seleccionar un punto de información sin mapa. 	El sistema no hará nada si no es un dato con mapa, ni vínculo, ni un texto que no entre en la pantalla.	No se realiza ninguna acción.	OK

<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar una ciudad. 3. Abrir el menú lateral. 4. Seleccionar una categoría. 5. Seleccionar un punto de información con mapa. 6. Volver a la pantalla anterior. 	<p>El sistema deberá volver a la pantalla que contenía el punto de información con mapa.</p>	<p>Se vuelve a la pantalla de información del lugar.</p>	<p>OK</p>
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar el mapa de una ciudad con mapa descargado. 	<p>El sistema deberá mostrar un mapa con los puntos de la ciudad geolocalizados.</p>	<p>Se muestra un mapa de la ciudad con los puntos geolocalizados.</p>	<p>OK</p>
<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Seleccionar el mapa de una ciudad con mapa no descargado. 	<p>El sistema deberá indicar que el mapa no está descargado y dar la posibilidad de descargarlo.</p>	<p>Se indica que el mapa de la ciudad no está disponible de manera offline y se pregunta si se quiere descargar indicando su información.</p>	<p>OK</p>
NFR-01 Adaptar la visualización a la resolución de pantalla			
<p>El sistema deberá adaptar la representación de la información a la resolución de pantalla del dispositivo en el que se ejecuta</p>			
<p>Se probará la aplicación en un dispositivo con pantalla de entre 3 y 4 pulgadas.</p>	<p>El sistema adaptará la representación de la información.</p>	<p>El contenido de las pantallas se adapta a la resolución del dispositivo.</p>	<p>OK</p>
<p>Se probará la aplicación en un dispositivo con pantalla de entre 4 y 5 pulgadas.</p>	<p>El sistema adaptará la representación de la información.</p>	<p>El contenido de las pantallas se adapta a la resolución del dispositivo.</p>	<p>OK</p>
<p>Se probará la aplicación en un dispositivo con pantalla de más de 5 pulgadas.</p>	<p>El sistema adaptará la representación de la información.</p>	<p>El contenido de las pantallas se adapta a la resolución del dispositivo.</p>	<p>OK</p>
NFR-02 Adaptar la visualización a la orientación de la pantalla			
<p>El sistema deberá adaptar la representación de la información según la orientación de la pantalla en cada momento</p>			
<p>Se probarán las pantallas de la aplicación a las posiciones horizontal y vertical del dispositivo.</p>	<p>El sistema adaptará la representación de la información.</p>	<p>El contenido de las pantallas se adapta a la disposición del dispositivo.</p>	<p>OK</p>
NFR-03 Mantener su funcionamiento sin necesidad de conexión continua a Internet			
<p>El sistema deberá mantener su funcionamiento principal sin necesidad de conectarse a internet de forma continua.</p>			

Se realizarán las pruebas anteriores modificando la configuración de la conexión para que la aplicación no se conecte nunca (Excepto las pruebas de los requisitos FRQ-02 y FRQ-03).	El sistema no debe consumir datos de la red y las pruebas deben obtener los mismos resultados.	El sistema no se conecta a la red en ninguna de las pantallas de información. No se puede acceder a la pantalla de ciudades para descargar, la aplicación muestra un aviso de que es necesaria la conexión a Internet.	OK
Se descargarán los mapas offline de una de las ciudades y se realizará las pruebas del requisito FRQ-07.	El sistema no debe consumir datos de la red y las pruebas deben tener los mismos resultados.	El sistema no se conecta a la red para mostrar el mapa.	OK
NFR-04 Implementación de un crawler configurable			
Deberá de poderse configurar en el sistema la información a mostrar en cada categoría y como encontrar dicha información semántica.			
<ol style="list-style-type: none"> 1. El usuario modifica el archivo xml de configuración de una categoría a través de un editor externo. 2. Iniciar la aplicación. 3. Seleccionar una ciudad. 4. Abrir el menú lateral. 5. Seleccionar la categoría modificada. 	El sistema mostrará la nueva configuración de la información.	El sistema muestra la información según la nueva configuración.	OK
Otras pruebas			
Se realizarán las pruebas anteriores en diferentes dispositivos Android.	El sistema debe tener el comportamiento esperado independientemente del dispositivo empleado.	El resultado de las pruebas anteriores es correcto para los siguientes dispositivos: <ul style="list-style-type: none"> • Samsung i9000 • Samsung i9300 • Samsung i9505 • Nexus 4 • Motorola Moto G 	OK
Se realizarán las pruebas anteriores en diferentes máquinas del emulador de Android para ordenadores.	El sistema debe tener el comportamiento esperado independientemente de la máquina configurada.	El sistema mantiene el comportamiento con diferentes configuraciones del emulador.	OK

Tabla 36 Pruebas de caja negra realizadas

Capítulo 6. Manual de Usuario

6.1. Instalación de HdTourist

HdTourist es una aplicación para dispositivos móviles Android, por lo que su archivo de instalación tiene la extensión *apk*.

Hay dos formas de instalar aplicaciones apk en Android: desde Google Play y copiando directamente el archivo de instalación en el dispositivo. HdTourist no está disponible actualmente en Google Play, por lo que la forma de instalación será la última.

Antes de comenzar la instalación necesitamos que Android permita la instalación de programas fuera de Google Play. Para ello activaremos la opción encontrada en “Ajustes → Seguridad → Orígenes desconocidos”.

Posteriormente, debemos copiar el archivo de instalación incluido en el CD que acompaña a esta memoria, al dispositivo Android y una vez copiado ejecutarlo para comenzar la instalación.

Al comenzar la instalación el dispositivo indicará los permisos que la aplicación requiere (Ilustración 31):

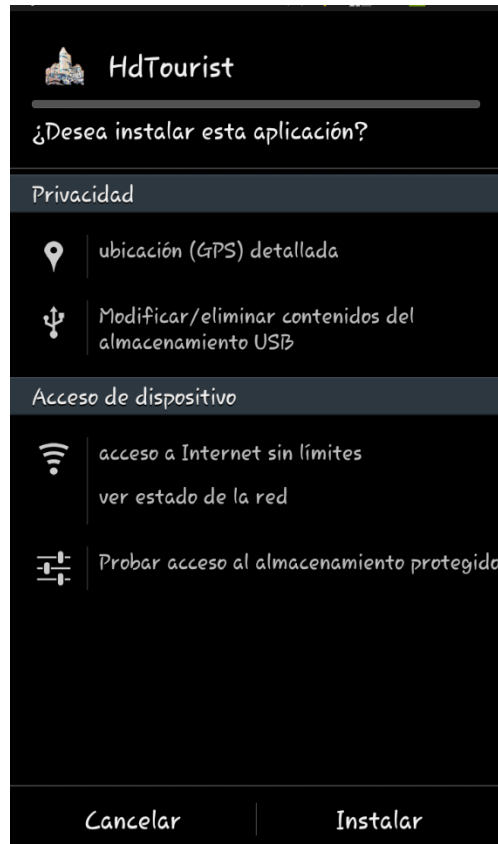


Ilustración 31 HdTourist: Permisos de instalación

Una vez instalada aparecerá un mensaje de confirmación (Ilustración 32):



Ilustración 32 HdTourist: Confirmación de instalación

6.2. Guía de uso

6.2.1. Descarga de ciudades

Cada vez que se inicie la aplicación, se mostrará la pantalla de bienvenida (Ilustración 33):

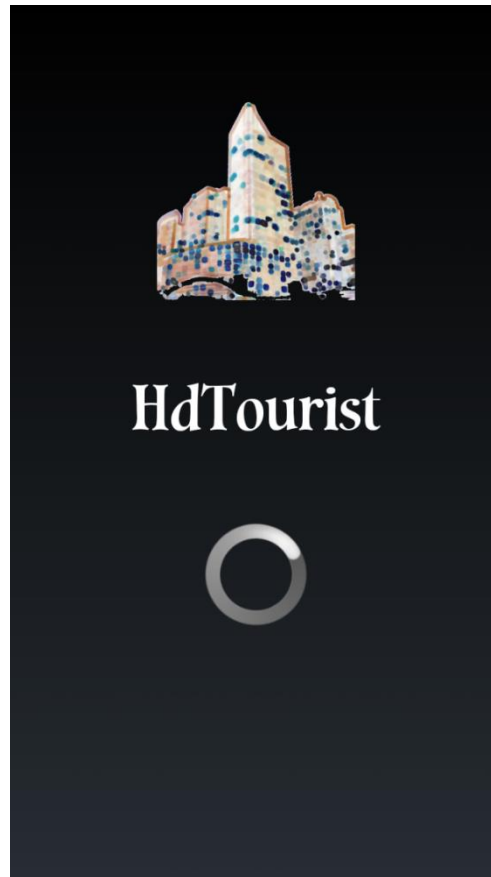


Ilustración 33 HdTourist: Pantalla de bienvenida

La pantalla de inicio de la aplicación muestra las ciudades descargadas en el dispositivo. La primera vez que se ejecuta la aplicación, no hay ciudades descargadas, por lo que se mostrará la siguiente pantalla indicando que deben descargarse (Ilustración 34):

Para descargar la información de una nueva ciudad, se debe acceder al menú superior derecho (Ilustración 35):

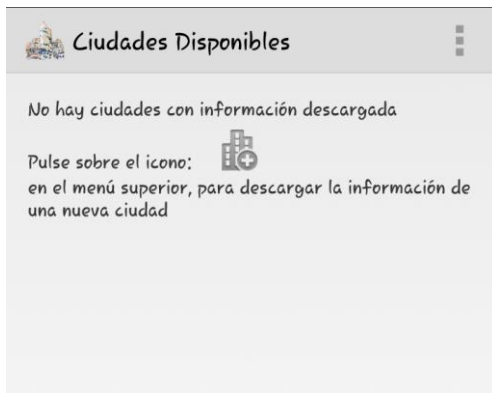


Ilustración 34 HdTourist: Inicio sin ciudades descargadas

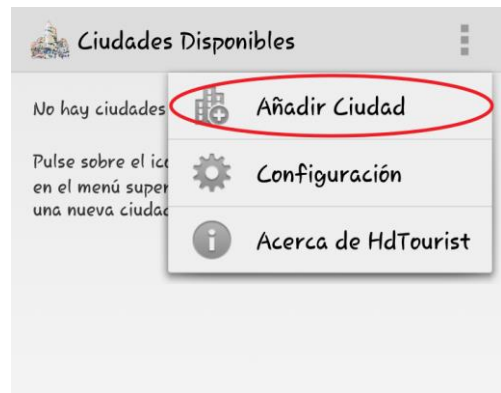


Ilustración 35 HdTourist: Menú añadir ciudad

Dependiendo de la configuración de la conexión a internet, aparecerá un mensaje preguntando si se quiere continuar ya que para obtener la lista de ciudades descargables es necesario acceder a la red de datos (Ilustración 36). Para poder descargar nuevas ciudades debemos aceptar el mensaje pulsando el botón "Sí". La configuración por defecto establece que siempre se pregunte antes de conectarse a Internet.

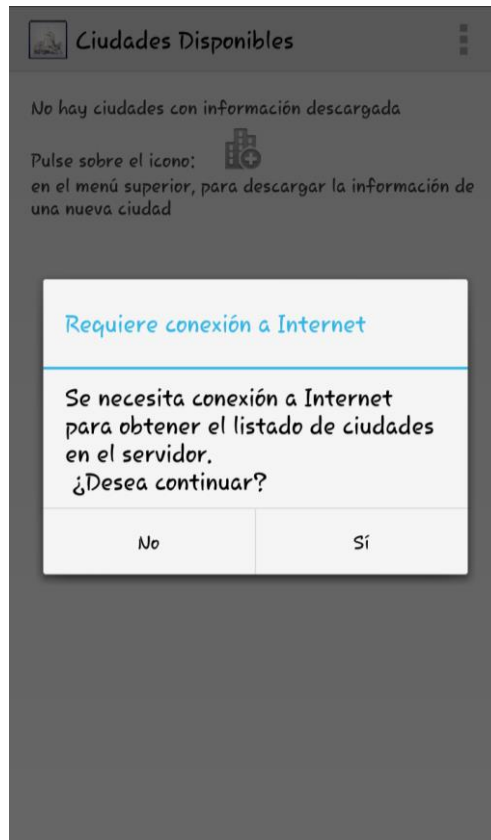


Ilustración 36 HdTourist: Confirmación para obtener el listado de ciudades

Una vez aceptado aparecerá la pantalla de obtención de ciudades con todas las ciudades disponibles para descargar (Ilustración 37). Pulsando el icono con la lupa podremos filtrar por el nombre de la ciudad (Ilustración 38).

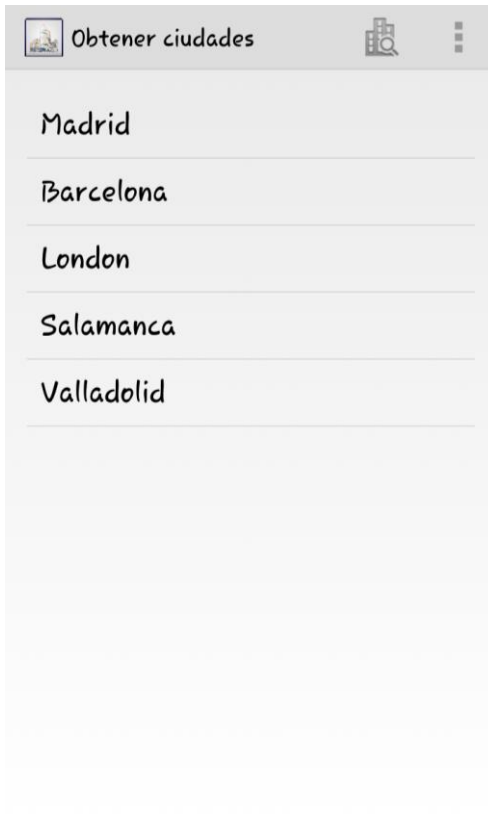


Ilustración 37 HdTourist: Pantalla de descarga de ciudades



Ilustración 38 HdTourist: Pantalla de filtro de ciudades para descargar

Seleccionando una ciudad comenzará el proceso de descarga de la misma. Y una vez finalizado, se mostrará la información principal de la ciudad descargada (Ilustración 42).

6.2.2. Lista de ciudades descargadas

Una vez que hemos descargado alguna ciudad, la pantalla principal de la aplicación mostrará las ciudades disponibles almacenadas en el dispositivo. Seleccionando una de ellas accederemos a su información (Ilustración 42).

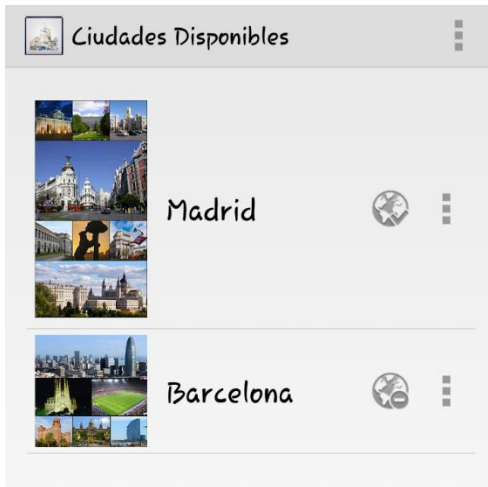


Ilustración 39 HdTourist: Ciudades para descargar



Ilustración 40 HdTourist: Menú en ciudades para descargar

Desde la pantalla de ciudades descargadas (Ilustración 39) podemos descargar mapas offline de las ciudades disponibles, eliminar los mapas si estos están descargados, eliminar la ciudad y acceder a la información de los ficheros HDT (Ilustración 40). Si los mapas están descargados, podemos acceder al mapa offline de la ciudad con los puntos geolocalizados (Ilustración 41).

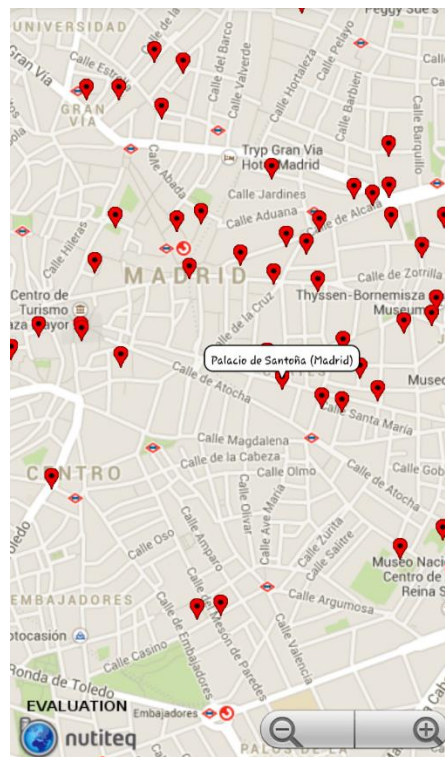


Ilustración 41 HdTourist: Mapa de la ciudad con puntos geolocalizados

6.2.3. Ver la información de una ciudad

Podemos acceder a la pantalla principal de información de una ciudad (Ilustración 42) a través del listado principal de ciudades disponibles o al terminar la descarga de una ciudad.

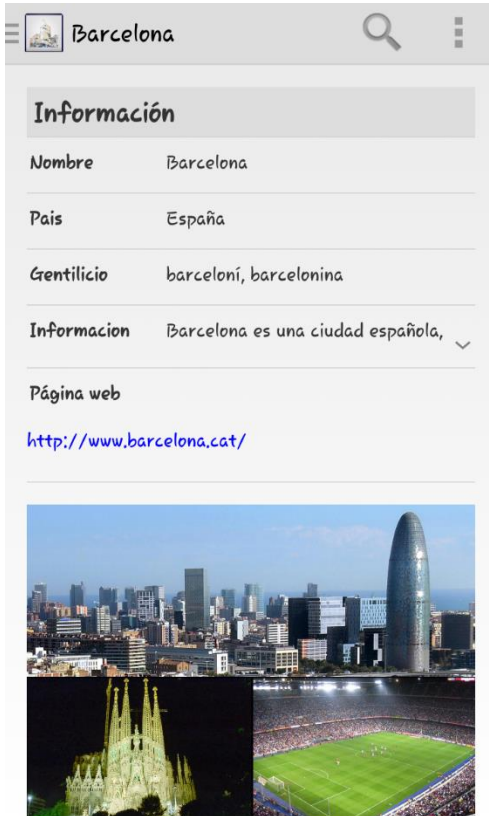


Ilustración 42 HdTourist: Pantalla principal de información

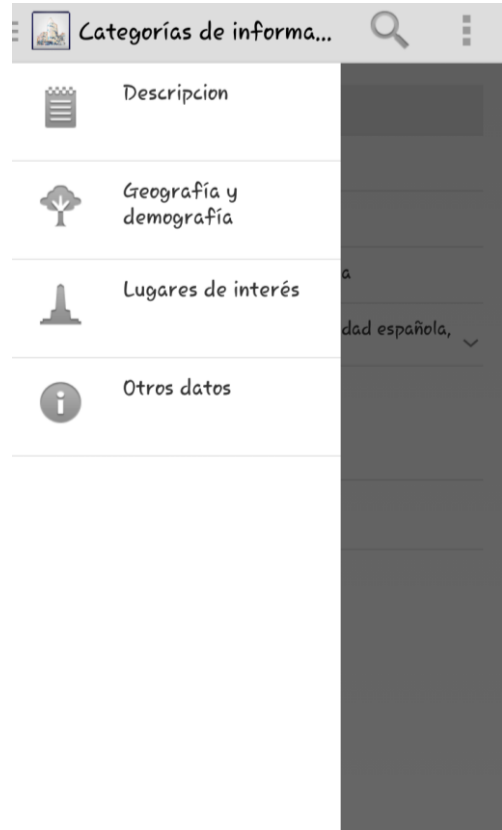


Ilustración 43 HdTourist: Categorías de información

La información de la ciudad se divide en categorías, accesibles a través de un menú lateral de la pantalla de información (Ilustración 43). Pulsando en cualquiera de las opciones accederemos a la información de la categoría elegida.

Las pantallas de información pueden mostrar imágenes (como en la Ilustración 42) si estas se han activado en la configuración. Las imágenes son obtenidas de recursos de Internet, por lo que para descargarlas se necesita el permiso del usuario. Cuando una pantalla de información contiene una imagen y estas están activadas, se mostrará un icono con un texto indicando que se puede descargar la imagen pulsando sobre el icono (Ilustración 44).

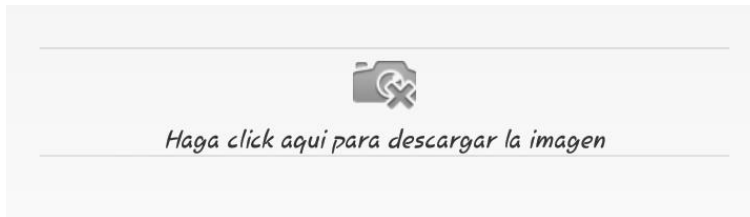


Ilustración 44 HdTourist: Icono para descargar imágenes

Las imágenes descargadas se almacenan en el teléfono por lo que no será necesario descargarlas de nuevo.

Otra de las posibilidades de las pantallas de información son los vínculos. Los vínculos permiten mostrar información ampliada acerca del elemento mostrado. Por ejemplo, en la categoría "Lugares" (Ilustración 45), se pueden ver varias listas de lugares ubicados en la ciudad (plazas, museos, palacios... etc), y podemos visualizar más información pulsando el lugar deseado (Ilustración 46):

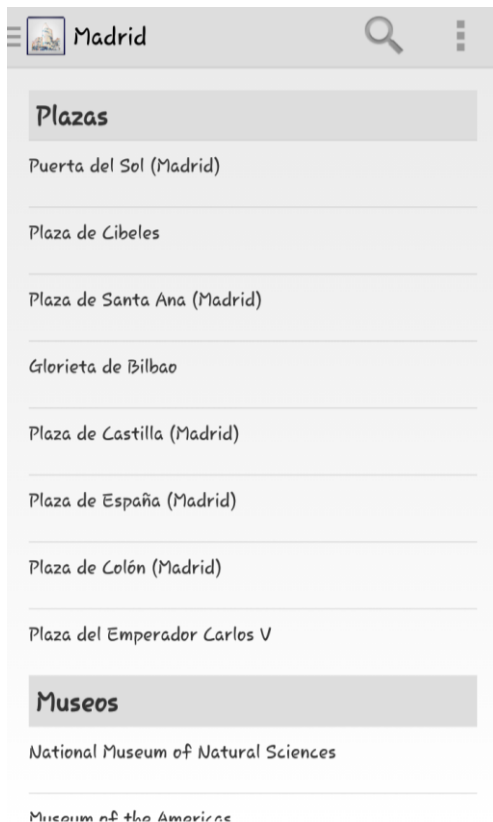


Ilustración 45 HdTourist: Categoría lugares

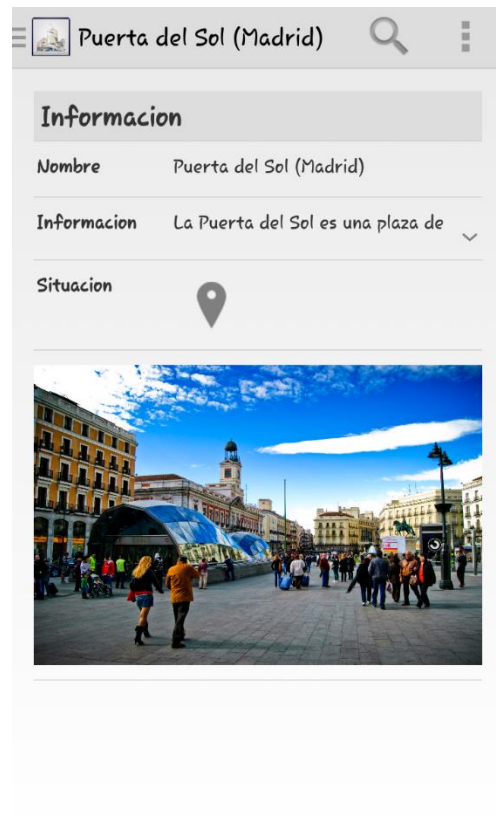


Ilustración 46 HdTourist: Información sobre un lugar

En la Ilustración 46 se muestra la información sobre un lugar de una ciudad. Otro vínculo visible en esta ilustración es la "Situación" que abre un mapa con la localización del lugar en cuestión (Ilustración 47).

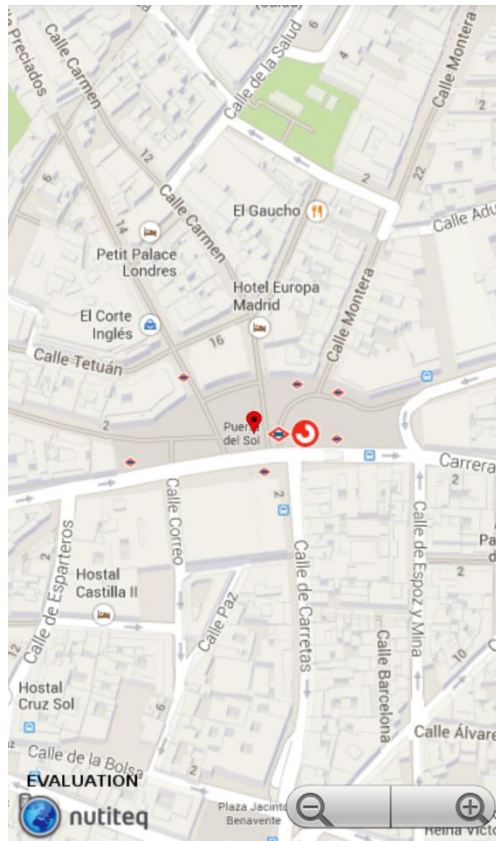


Ilustración 47 HdTourist: Situación en mapa del lugar

6.2.4. Búsqueda

Desde las pantallas de información de la ciudad podemos buscar información accediendo a través de la lupa visible en la parte superior derecha (Ilustración 48). Los resultados mostrarán los lugares y las personas cuyo nombre coincida en todo o en parte con la clave buscada (Ilustración 49).



Ilustración 48 HdTourist: Búsqueda de información



Ilustración 49 HdTourist: Resultados de la búsqueda

6.2.5. Información de archivo HDT

Desde el menú de las pantallas de ciudades disponibles (Ilustración 40) y de las pantallas de información de una ciudad (Ilustración 50), se puede acceder a la información del fichero HDT.



Ilustración 50 HdTourist: Menú de las pantallas de información

En la pantalla de información del fichero HDT, se mostrarán datos como el nombre y el tamaño del fichero, la cantidad de triples que contiene y otra información semántica.

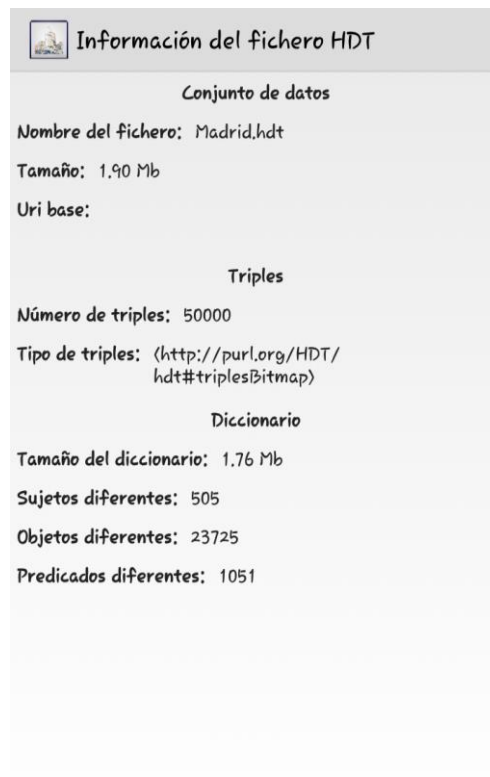
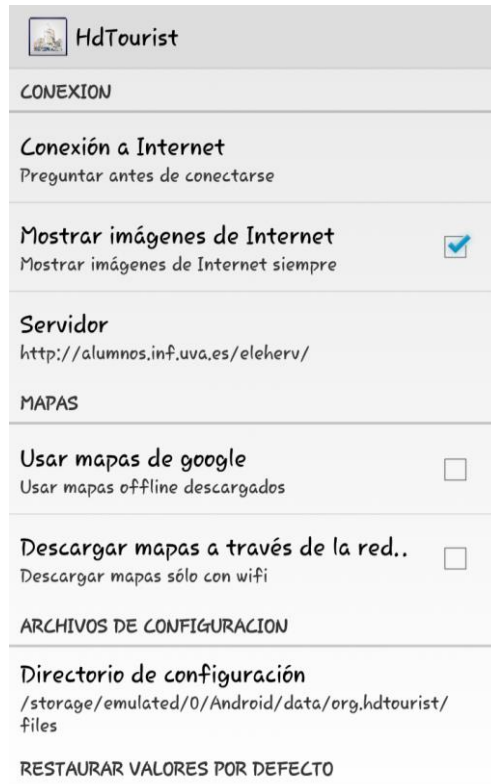


Ilustración 51 HdTourist: Información del fichero HDT

6.2.6. Configuración

La aplicación permite configurar parte de su funcionamiento a través de la pantalla de configuración (Ilustración 52) accesible desde los menús de las pantalla de ciudades disponibles y de información de ciudad (Ilustración 35 y Ilustración 50).



Restaurar valores por defecto

Ilustración 52 HdTourist: Configuración

6.2.6.1. Conexión

Esta opción permite configurar los permisos de acceso de la aplicación a Internet, sus opciones son:

- Preguntar antes de conectarse: Si está opción está seleccionada, se pedirá una confirmación al usuario antes de conectarse a Internet.
- No conectarse nunca: Si esta opción está seleccionada, se mostrará un mensaje indicando la limitación de acceso a Internet y no se continuará la acción solicitada.
- Conectarse siempre: Se conectará siempre que sea necesario a Internet, si necesidad de confirmación del usuario.

6.2.6.2. Descarga de imágenes

Las pantallas de información de las ciudades pueden contener imágenes que necesitan descargarse de Internet. Debido al consumo mayor de datos necesario para la descarga de imágenes, se puede desactivar la visualización de las mismas.

Una vez desactivadas las imágenes, estas solo se mostrarán si anteriormente ya se habían descargado y se encuentran almacenadas en el dispositivo.

Si las imágenes se encuentran activadas, no se descargarán automáticamente, si no que el usuario debe solicitarlo y confirmar en caso necesario la conexión a Internet (Ilustración 44).

6.2.6.3. Servidor

El servidor donde se encuentran disponibles los archivos necesarios para descargar ciudades y mapas puede ser cambiado. El cambio a un servidor que no contenga la estructura adecuada de directorios puede provocar errores en la aplicación.

6.2.6.4. Mapas

Se puede cambiar la configuración de los mapas para que se utilice la aplicación externa “Google Maps” en vez de los mapas offline.

También se puede habilitar la descarga de mapas a través de la red de datos, aunque esta opción está desaconsejada debido a el tamaño de los mismos, si esta opción no está habilitada la aplicación solo permitirá la descarga de mapas mediante una red wifi.

6.2.6.5. Directorio de configuración

El directorio donde la aplicación guarda los archivos de configuración, así como los ficheros HDT de las ciudades, la caché de imágenes y los mapas offline puede ser cambiado mediante esta opción.

6.2.6.6. Restaurar valores por defecto

Desde esta opción se permite restaurar los valores por defecto:

- Preguntar antes de conectar a Internet.
- No descargar imágenes.
- Servidor proporcionado por los tutores para la descarga de archivos.
- Uso de mapas offline y descarga de los mismos mediante la red wifi.
- Almacenamiento de archivos de configuración en el directorio por defecto dado por el sistema operativo Android.

Capítulo 7. Conclusiones

7.1. Objetivos alcanzados

Con el desarrollo de este Trabajo Fin de Grado, se han conseguido todos los objetivos propuestos al inicio del proceso.

El objetivo principal consistía en realizar una primera aproximación hacia una interfaz gráfica de consulta móvil de información semántica en el formato comprimido HDT, centrándonos en el caso del turismo offline. Este objetivo se ha alcanzado de forma satisfactoria.

Por otro lado, personalmente este proyecto me ha permitido profundizar en la filosofía de la Web Semántica y sus tecnologías asociadas. Así como las posibilidades que puede brindar en el desarrollo de nuevas aplicaciones.

Personalmente también he aprendido a desarrollar aplicaciones para dispositivos móviles con plataforma Android. Teniendo en cuenta la gran expansión que han tenido estos dispositivos y su actual cuota de mercado, considero que aprender a desarrollar en esta plataforma ha sido una experiencia de gran valor y posiblemente de gran utilidad para el futuro.

7.2. Dificultades

A continuación se detallan las principales dificultades que han surgido durante la elaboración y desarrollo del presente Trabajo Fin de Grado:

- El desarrollo de la aplicación en una plataforma en la que se carecía de experiencia ha sido una de las dificultades de este proyecto. Si se tenía experiencia en la programación en Java, que al ser el lenguaje base de Android ha facilitado en gran medida el proceso de aprendizaje. La gran cantidad de tutoriales e información encontrada en Internet, así como el curso que se realizó al inicio del proyecto [34] ha ayudado en la etapa de elaboración de la aplicación.
- Otro de los problemas que se han encontrado al desarrollar la aplicación ha sido la fragmentación. Se ha querido que la aplicación estuviera disponible para el mayor número de dispositivos posibles, por lo que la versión mínima de Android elegida ha sido baja, *Gingerbread v2.3 API 9*. Programar teniendo en cuenta la retrocompatibilidad ha sido mucho más complicado que programar directamente para la versión más actual, ya que ha sido necesario invertir más tiempo en adaptar el contenido para lidiar con versiones antiguas del sistema operativo. Una de las soluciones a este problema ha sido añadir una librería de soporte desarrollada por Google para paliar este problema. Esta librería ha permitido implementar características de las últimas versiones haciéndolas compatibles para todos los dispositivos.
- La gestión de memoria ha sido otro problema, Android automatiza la gestión de memoria y esto hace que cuando se encuentra un problema de fuga de memoria su solución no sea sencilla, una de las causas de estas fugas de memoria es la carga de imágenes. Al cargar varias veces una imagen (por ejemplo al girar el dispositivo), no se libera correctamente la memoria. Esto se ha solucionado utilizando una librería externa que gestiona la liberación de la

memoria de las imágenes y a mayores nos ha proporcionado una caché para que no sea necesario descargar más de una vez la misma imagen [1].

- La búsqueda de una herramienta que permitiera gestionar la visualización de mapas sin conexión ha sido otra dificultad. No todas las herramientas encontradas permitían gestionar los mapas desde la propia aplicación o incluso muchas de ellas requerían la instalación de otras aplicaciones en el dispositivo. Finalmente se ha encontrado una librería que se adaptaba a nuestros requerimientos y se ha podido configurar con éxito.
- Otro problema ha sido generar consultas HDT de información que obtuvieran información concreta de ciudades diferentes. Debido a que en DBPedia distintas ciudades han utilizado distintas ontologías. Esto se ha solucionado añadiendo la posibilidad de hacer conjunciones entre consultas.

7.3. Trabajo futuro

Una vez terminado este proyecto, existen varios puntos en los que se podría seguir trabajando:

- Una de las mejoras, sería añadir más ciudades a la aplicación. Para que HdTourist pueda considerarse realmente una aplicación de turismo offline debería proporcionar al usuario información sobre aquellas ciudades que va a visitar, por lo que aumentar el número de ciudades permitiría ampliar su funcionalidad.
- De cara a la información semántica y dada la gran información contenida en los ficheros HDT, se podrían añadir categorías de información y ampliar las existentes modificando los ficheros de configuración para ampliar esta información en las pantallas.
- En cuanto a los mapas, estos podrían ser mejorados añadiendo funcionalidades. Por ejemplo,
 - Calculando la distancia desde el punto actual hasta cualquier punto geolocalizado dentro de una ciudad o añadiendo el cálculo de rutas en modo offline. La librería utilizada permite el enrutamiento offline basado en la herramienta Graphhopper [35].
 - Posibilidad de guardar puntos geolocalizados personalizados por el usuario. Es decir, en una aplicación de turismo puede ser interesante permitir al usuario guardar la dirección del hotel donde se aloja y mostrársela en el mapa, e incluso permitir calcular rutas desde el hotel a puntos de interés o al revés.
- Finalmente, aunque no sea una mejora propiamente dicha, se podría registrar y subir la aplicación a Google Play. Google Play es la plataforma de distribución de aplicaciones de Android, permite a los usuarios buscar y descargar aplicaciones, libros, música... etc. Añadir HdTourist a Google Play sería el paso necesario para lanzar la aplicación al público general.

Capítulo 8. Bibliografía

- [1] AmbySoft.
«The Agile Unified Process (AUP)» [En línea].
Disponible en: <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.
Último acceso en: Marzo 2013.
- [2] IBM.
«IBM Rational Unified Process (RUP)» [En línea].
Disponible en: <http://www-01.ibm.com/software/rational/rup/>.
Último acceso en: 25 Febrero 2014.
- [3] P. Torrecilla.
«El proceso unificado ágil: Modelos y documentación» [En línea].
Disponible en: <http://nosolopau.com/2012/06/08/mas-sobre-el-proceso-unificado-agil-modelos-y-documentacion/>.
Publicado en: Junio 2012-
Último acceso en: Mayo 2013.
- [4] W3C.
«Guía Breve de Linked Data» [En línea].
Disponible en: <http://www.w3c.es/Divulgacion/GuiasBreves/LinkedData>.
Último acceso en: Febrero 2014.
- [5] C. Gutierrez.
Modeling the Web of Data (introductory overview).
RW'11 Proceedings of the 7th international conference on Reasoning web: semantic technologies for the web of data. pp. 416-444
Galway, Ireland, 2011.
- [6] C. Bizer, T. Heath y T. Berners-Lee.
«Linked data-the story so far»
International journal on semantic web and information systems. Vol. 5, nº 3, pp. 1-22.
2009.
- [7] W3C.
«Semantic Web» [En línea].
Disponible en: <http://www.w3.org/standards/semanticweb/>.
Último acceso en: Febrero 2014.
- [8] Kioskea.net.
«Web semántica: las aplicaciones actuales» [En línea].
Disponible en: <http://es.kioskea.net/faq/7082-web-semantica-las-aplicaciones-actuales>
Publicado en: Mayo 2014.
Último acceso en: Mayo 2014.

- [9] M. Alvarez.
«Uso de Lenguajes Documentales en la Web Semántica» [En línea].
Disponible en: http://www.sedic.es/p_boletinclip44_sedicabierto3.asp.
Último acceso en: Febrero 2014.
- [10] Dave Beckett.
«RDF/XML Syntax Specification», W3C Recommendation, 2004.
Disponible en: <http://www.w3.org/TR/REC-rdf-syntax/>.
- [11] Tim Berners-Lee, Dan Connolly.
«Notation3 (N3): A readable RDF syntax», W3C Recommendation, 2011.
Disponible en: <http://www.w3.org/TeamSubmission/n3/>.
- [12] David Beckett, Tim Berners-Lee.
«Turtle - Terse RDF Triple Language», W3C Recommendation, 2011.
Disponible en: <http://www.w3.org/TeamSubmission/turtle/>.
- [13] JSON-LD.
«JSON for Linking Data» [En línea].
Disponible en: <http://json-ld.org/>.
Último acceso en: Mayo 2014.
- [14] Frank Manola, Eric Miller.
«RDF Primer», W3C Recommendation, 2004.
Disponible en: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>.
- [15] Gavin Carothers, Andy Seaborne.
«RDF 1.1 N-Triples», W3C Recommendation, 2014.
Disponible en: <http://www.w3.org/TR/n-triples/>.
- [16] Eric Prud'hommeaux, Andy Seaborne.
«SPARQL Lenguaje de consulta para RDF», W3C Recommendation, 2008.
Disponible en: <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>.
- [17] J.D. Fernández, M.A. Martínez-Prieto, C. Gutiérrez, A. Polleres, M. Arias.
«Binary RDF Representation for Publication and Exchange».
Journal of Web Semantics, vol. 19, pp. 22-41, 2013.
- [18] J.D. Fernández, M.A. Martínez-Prieto, C. Gutiérrez, A. Polleres
«Binary RDF Representation for Publication and Exchange (HDT) Submission»,
W3C Recommendation, 2011
Disponible en: <http://www.w3.org/Submission/2011/03/>.
- [19] DBPedia Blog.
«About DBPedia» [En línea].
Disponible en: <http://wiki.dbpedia.org/About>.
Último acceso en: Marzo 2014.

- [20] Datahub.
«DBPedia» [En línea].
Disponible en: <http://datahub.io/es/dataset/dbpedia>.
Último acceso en: Marzo 2014.
- [21] H. Dediu.
«Asymco» [En línea].
Disponible en: <http://www.asymco.com/2011/02/19/the-lives-and-deaths-of-mobileplatforms/>.
Publicado en: Febrero 2011.
Último acceso en: Febrero 2014.
- [22] J. Tomás Gironés.
«El Gran Libro de Android».
Publicado en: Febrero de 2013.
Editorial Marcombo, 3ª edición.
- [23] Android Developers [En línea].
Disponible en: <http://developer.android.com/design/index.html>.
Último acceso en: Abril 2014.
- [24] J. Maturana.
«Android consolidando sus nuevas versiones» [En línea].
Disponible en: <http://www.xataka.com/moviles/android-consolidando-sus-nuevasversiones-acabando-con-la-fragmentacion>.
Publicado en: Enero 2014.
Último acceso en: Febrero 2014.
- [25] DBpedia Blog.
«DBpedia Mobile» [En línea].
Disponible en: <http://wiki.dbpedia.org/DBpediaMobile>.
Último acceso en: Marzo 2014.
- [26] A. Llaves, A. Fernández y O. Corcho.
«Map4RDFiOS» [En línea].
Linking Geospatial Data, London, W3C/OGC. Marzo 2014.
Disponible en: http://www.w3.org/2014/03/lgd/papers/lgd14_submission_46.
- [27] D. Le-Phuoc, J. X. Parreira, V. Reynolds y M. Hauswirth.
«RDF On the Go» [En línea].
The 9th International Semantic Web Conference (ISWC2010), Shanghai, China. 2010.
Disponible en: <http://ceur-ws.org/Vol-658/paper503.pdf>.

- [28] D. Jérôme , E. Jérôme Euzenat y M.-E. Roçoiu.
«Linked data from your pocket» [En línea].
The 9th International Semantic Web Conference (ISWC2010), Shanghai, China.
2010.
Disponible en: http://hal.univ-grenoble-alpes.fr/docs/00/76/84/18/PDF/paper_3.pdf.
- [29] «DataWeb» [En línea].
Disponible en: <http://dataweb.infor.uva.es/>.
Último acceso en: Marzo 2014.
- [30] Nutiteq [En línea].
Disponible en: <http://www.nutiteq.com/>.
Último acceso en: Abril 2014.
- [31] S. Tarasevich.
«Universal Image Loader» [En línea].
Disponible en: <https://github.com/nostra13/Android-Universal-Image-Loader/wiki>.
Último acceso en: Abril 2014.
- [32] MapsWithMe [En línea].
Disponible en: <https://github.com/mapswithme/api-android>.
Último acceso en: Abril 2014.
- [33] gMapMaker [En línea].
Disponible en: <http://archives.damiendebin.net/gMapMaker/>.
Último acceso en: Abril 2014.
- [34] Miriadax.
«Android: Programación de Aplicaciones» [En línea].
Disponible en: https://www.miriadax.net/web/android_programacion.
Último acceso en: Mayo 2013.
- [35] Nutiteq.
«Offline routing» [En línea].
Disponible en: <https://github.com/nutiteq/hellomap3d/wiki/Offline-routing>.
Último acceso en: Abril 2014.

Capítulo 9. Anexos

9.1. Consultas sparql para generar ficheros HDT

9.1.1. Madrid.hdt

```

CONSTRUCT { <http://dbpedia.org/resource/Madrid> ?a ?b .
?pb <http://www.w3.org/2000/01/rdf-schema#label> ?pc .
?ha ?hc ?hd .
?ma2 ?mc2 ?md2 .
?ea ?ec ?ed .
?pa ?pc ?pd .
?pla ?plc ?pld .
?pl2a ?pl2c ?pl2d .
?la ?lc ?ld .
?p3a ?p3c ?p3d .
?moa ?moc ?mod .
?pea ?pec ?ped .
?pe2a ?pe2c ?pe2d .
}
WHERE{
{<http://dbpedia.org/resource/Madrid> ?a ?b .
} UNION {
<http://dbpedia.org/resource/Madrid>
<http://dbpedia.org/ontology/country> ?pb .
?pb <http://www.w3.org/2000/01/rdf-schema#label> ?pc .
} UNION {
?ha ?hb <http://dbpedia.org/resource/Madrid> .
?ha rdf:type <http://dbpedia.org/ontology/HistoricPlace> .
?ha ?hc ?hd .
} UNION {
?ma2 ?mb2 <http://dbpedia.org/resource/Madrid> .
?ma2 rdf:type <http://dbpedia.org/class/yago/MuseumsInMadrid> .
?ma2 ?mc2 ?md2 .
}

```

```

} UNION {
?ea ?eb <http://dbpedia.org/resource/Madrid> .
?ea rdf:type
<http://dbpedia.org/class/yago/BuildingsAndStructuresInMadrid> .
?ea ?ec ?ed .
} UNION {
?pa ?pb <http://dbpedia.org/resource/Madrid> .
?pa rdf:type <http://dbpedia.org/class/yago/PalacesInMadrid> .
?pa ?pc ?pd .
} UNION {
?pla ?plb <http://dbpedia.org/resource/Madrid> .
?pla rdf:type <http://dbpedia.org/class/yago/PlazasInMadrid> .
?pla ?plc ?pld .
} UNION {
?pl2a ?pl2b <http://dbpedia.org/resource/Madrid> .
?pl2a rdf:type
<http://dbpedia.org/class/yago/RedevelopedPortsAndWaterfronts> .
?pl2a ?pl2c ?pl2d .
} UNION {
?la ?lb <http://dbpedia.org/resource/Madrid> .
?la rdf:type <http://schema.org/LandmarksOrHistoricalBuildings>
.
?la ?lc ?ld .
} UNION {
?p3a ?p3b <http://dbpedia.org/resource/Madrid> .
?p3a rdf:type <http://dbpedia.org/ontology/Place> .
?p3a ?p3c ?p3d .
} UNION {
?moa ?mob <http://dbpedia.org/resource/Madrid> .
?moa rdf:type
<http://es.dbpedia.org/resource/Categor%C3%ADa:Arquitectura_de_la_provincia_de_Madrid> .
?moa ?moc ?mod .
} UNION {

```

```

?pea ?peb <http://dbpedia.org/resource/Madrid> .
?pea rdf:type <http://dbpedia.org/ontology/Person> .
?pea ?pec ?ped .
} UNION {
<http://dbpedia.org/resource/Madrid> ?pe2b ?pe2a .
?pe2a rdf:type <http://dbpedia.org/ontology/Person> .
?pe2a ?pe2c ?pe2d .
}}

```

9.1.2. Barcelona.hdt

```

CONSTRUCT { <http://dbpedia.org/resource/Barcelona> ?a ?b .
?pb <http://www.w3.org/2000/01/rdf-schema#label> ?pc .
?ha ?hc ?hd .
?ma2 ?mc2 ?md2 .
?ea ?ec ?ed .
?pa ?pc ?pd .
?pla ?plc ?pld .
?pl2a ?pl2c ?pl2d .
?la ?lc ?ld .
?p3a ?p3c ?p3d .
?moa ?moc ?mod .
?pea ?pec ?ped .
?pe2a ?pe2c ?pe2d .
}
WHERE{
{<http://dbpedia.org/resource/Barcelona> ?a ?b .
} UNION {
<http://dbpedia.org/resource/Barcelona>
<http://dbpedia.org/ontology/country> ?pb .
?pb <http://www.w3.org/2000/01/rdf-schema#label> ?pc .
} UNION {
?ha ?hb <http://dbpedia.org/resource/Barcelona> .

```

```

?ha rdf:type <http://dbpedia.org/ontology/HistoricPlace> .
?ha ?hc ?hd .
} UNION {
?ma2 ?mb2 <http://dbpedia.org/resource/Barcelona> .
?ma2 rdf:type <http://dbpedia.org/class/yago/MuseumsInBarcelona>
.
?ma2 ?mc2 ?md2 .
} UNION {
?ea ?eb <http://dbpedia.org/resource/Barcelona> .
?ea rdf:type
<http://dbpedia.org/class/yago/BuildingsAndStructuresInBarcelona
> .
?ea ?ec ?ed .
} UNION {
?pa ?pb <http://dbpedia.org/resource/Barcelona> .
?pa rdf:type <http://dbpedia.org/class/yago/PalacesInBarcelona>
.
?pa ?pc ?pd .
} UNION {
?pla ?plb <http://dbpedia.org/resource/Barcelona> .
?pla rdf:type <http://dbpedia.org/class/yago/PlazasInBarcelona>
.
?pla ?plc ?pld .
} UNION {
?pl2a ?pl2b <http://dbpedia.org/resource/Barcelona> .
?pl2a rdf:type
<http://dbpedia.org/class/yago/RedevelopedPortsAndWaterfronts> .
?pl2a ?pl2c ?pl2d .
} UNION {
?la ?lb <http://dbpedia.org/resource/Barcelona> .
?la rdf:type <http://schema.org/LandmarksOrHistoricalBuildings>
.
?la ?lc ?ld .
} UNION {
?p3a ?p3b <http://dbpedia.org/resource/Barcelona> .

```



```

?p3a rdf:type <http://dbpedia.org/ontology/Place> .
?p3a ?p3c ?p3d .
} UNION {
?moa ?mob <http://dbpedia.org/resource/Barcelona> .
?moa rdf:type
<http://es.dbpedia.org/resource/Categor%C3%ADa:Arquitectura_de_la_provincia_de_Barcelona> .
?moa ?moc ?mod .
} UNION {
?pea ?peb <http://dbpedia.org/resource/Barcelona> .
?pea rdf:type <http://dbpedia.org/ontology/Person> .
?pea ?pec ?ped .
} UNION {
<http://dbpedia.org/resource/Barcelona> ?pe2b ?pe2a .
?pe2a rdf:type <http://dbpedia.org/ontology/Person> .
?pe2a ?pe2c ?pe2d .
}}

```

9.1.3. Valladolid.hdt

```

CONSTRUCT { <http://dbpedia.org/resource/Valladolid> ?a ?b .
?pb <http://www.w3.org/2000/01/rdf-schema#label> ?pc .
?ha ?hc ?hd .
?ma2 ?mc2 ?md2 .
?ea ?ec ?ed .
?pa ?pc ?pd .
?pla ?plc ?pld .
?pl2a ?pl2c ?pl2d .
?la ?lc ?ld .
?p3a ?p3c ?p3d .
?moa ?moc ?mod .
?pea ?pec ?ped .
?pe2a ?pe2c ?pe2d .

```

```

}
WHERE{
<http://dbpedia.org/resource/Valladolid> ?a ?b .
} UNION {
<http://dbpedia.org/resource/Valladolid>
<http://dbpedia.org/ontology/country> ?pb .
?pb <http://www.w3.org/2000/01/rdf-schema#label> ?pc .
} UNION {
?ha ?hb <http://dbpedia.org/resource/Valladolid> .
?ha rdf:type <http://dbpedia.org/ontology/HistoricPlace> .
?ha ?hc ?hd .
} UNION {
?ma2 ?mb2 <http://dbpedia.org/resource/Valladolid> .
?ma2 rdf:type
<http://dbpedia.org/class/yago/MuseumsInValladolid> .
?ma2 ?mc2 ?md2 .
} UNION {
?ea ?eb <http://dbpedia.org/resource/Valladolid> .
?ea rdf:type
<http://dbpedia.org/class/yago/BuildingsAndStructuresInValladolli
d> .
?ea ?ec ?ed .
} UNION {
?pa ?pb <http://dbpedia.org/resource/Valladolid> .
?pa rdf:type <http://dbpedia.org/class/yago/PalacesInValladolid>
.
?pa ?pc ?pd .
} UNION {
?pla ?plb <http://dbpedia.org/resource/Valladolid> .
?pla rdf:type <http://dbpedia.org/class/yago/PlazasInValladolid>
.
?pla ?plc ?pld .
} UNION {
?pl2a ?pl2b <http://dbpedia.org/resource/Valladolid> .

```

```

?p12a rdf:type
<http://dbpedia.org/class/yago/RedevelopedPortsAndWaterfronts> .
?p12a ?p12c ?p12d .
} UNION {
?p1a ?p1b <http://dbpedia.org/resource/Valladolid> .
?p1a rdf:type <http://schema.org/LandmarksOrHistoricalBuildings>
.
?p1a ?p1c ?p1d .
} UNION {
?p3a ?p3b <http://dbpedia.org/resource/Valladolid> .
?p3a rdf:type <http://dbpedia.org/ontology/Place> .
?p3a ?p3c ?p3d .
} UNION {
?moa ?mob <http://dbpedia.org/resource/Valladolid> .
?moa rdf:type
<http://es.dbpedia.org/resource/Categor%C3%ADa:Arquitectura_de_la
a_provincia_de_Valladolid> .
?moa ?moc ?mod .
} UNION {
?pea ?peb <http://dbpedia.org/resource/Valladolid> .
?pea rdf:type <http://dbpedia.org/ontology/Person> .
?pea ?pec ?ped .
} UNION {
<http://dbpedia.org/resource/Valladolid> ?pe2b ?pe2a .
?pe2a rdf:type <http://dbpedia.org/ontology/Person> .
?pe2a ?pe2c ?pe2d .
}}

```

9.1.4. Salamanca.hdt

```

CONSTRUCT { <http://dbpedia.org/resource/Salamanca> ?a ?b .
?pb <http://www.w3.org/2000/01/rdf-schema#label> ?pc .
?ha ?hc ?hd .
?ma2 ?mc2 ?md2 .

```

```

?ea ?ec ?ed .
?pa ?pc ?pd .
?pla ?plc ?pld .
?pl2a ?pl2c ?pl2d .
?la ?lc ?ld .
?p3a ?p3c ?p3d .
?moa ?moc ?mod .
?pea ?pec ?ped .
?pe2a ?pe2c ?pe2d .
}
WHERE{
{<http://dbpedia.org/resource/Salamanca> ?a ?b .
} UNION {
<http://dbpedia.org/resource/Salamanca>
<http://dbpedia.org/ontology/country> ?pb .
?pb <http://www.w3.org/2000/01/rdf-schema#label> ?pc .
} UNION {
?ha ?hb <http://dbpedia.org/resource/Salamanca> .
?ha rdf:type <http://dbpedia.org/ontology/HistoricPlace> .
?ha ?hc ?hd .
} UNION {
?ma2 ?mb2 <http://dbpedia.org/resource/Salamanca> .
?ma2 rdf:type <http://dbpedia.org/class/yago/MuseumsInSalamanca>
.
?ma2 ?mc2 ?md2 .
} UNION {
?ea ?eb <http://dbpedia.org/resource/Salamanca> .
?ea rdf:type
<http://dbpedia.org/class/yago/BuildingsAndStructuresInSalamanca
> .
?ea ?ec ?ed .
} UNION {
?pa ?pb <http://dbpedia.org/resource/Salamanca> .

```

```

?pa rdf:type <http://dbpedia.org/class/yago/PalacesInSalamanca>
.
?pa ?pc ?pd .
} UNION {
?pla ?plb <http://dbpedia.org/resource/Salamanca> .
?pla rdf:type <http://dbpedia.org/class/yago/PlazasInSalamanca>
.
?pla ?plc ?pld .
} UNION {
?pl2a ?pl2b <http://dbpedia.org/resource/Salamanca> .
?pl2a rdf:type
<http://dbpedia.org/class/yago/RedevelopedPortsAndWaterfronts> .
?pl2a ?pl2c ?pl2d .
} UNION {
?la ?lb <http://dbpedia.org/resource/Salamanca> .
?la rdf:type <http://schema.org/LandmarksOrHistoricalBuildings>
.
?la ?lc ?ld .
} UNION {
?p3a ?p3b <http://dbpedia.org/resource/Salamanca> .
?p3a rdf:type <http://dbpedia.org/ontology/Place> .
?p3a ?p3c ?p3d .
} UNION {
?moa ?mob <http://dbpedia.org/resource/Salamanca> .
?moa rdf:type
<http://es.dbpedia.org/resource/Categor%C3%ADa:Arquitectura_de_l
a_provincia_de_Salamanca> .
?moa ?moc ?mod .
} UNION {
?pea ?peb <http://dbpedia.org/resource/Salamanca> .
?pea rdf:type <http://dbpedia.org/ontology/Person> .
?pea ?pec ?ped .
} UNION {
<http://dbpedia.org/resource/Salamanca> ?pe2b ?pe2a .

```

```
?pe2a rdf:type <http://dbpedia.org/ontology/Person> .
?pe2a ?pe2c ?pe2d .
}}
```

9.1.5. London.hdt

```
CONSTRUCT { <http://dbpedia.org/resource/London> ?a ?b .
?pb <http://www.w3.org/2000/01/rdf-schema#label> ?pc .
?ha ?hc ?hd .
?ma2 ?mc2 ?md2 .
?ea ?ec ?ed .
?pa ?pc ?pd .
?pla ?plc ?pld .
?pl2a ?pl2c ?pl2d .
?la ?lc ?ld .
?p3a ?p3c ?p3d .
?moa ?moc ?mod .
?pea ?pec ?ped .
?pe2a ?pe2c ?pe2d .
}
WHERE{
{<http://dbpedia.org/resource/London> ?a ?b .
} UNION {
<http://dbpedia.org/resource/London>
<http://dbpedia.org/ontology/country> ?pb .
?pb <http://www.w3.org/2000/01/rdf-schema#label> ?pc .
} UNION {
?ha ?hb <http://dbpedia.org/resource/London> .
?ha rdf:type <http://dbpedia.org/ontology/HistoricPlace> .
?ha ?hc ?hd .
} UNION {
?ma2 ?mb2 <http://dbpedia.org/resource/London> .
?ma2 rdf:type <http://dbpedia.org/class/yago/MuseumsInLondon> .
```

```

?ma2 ?mc2 ?md2 .
} UNION {
?ea ?eb <http://dbpedia.org/resource/London> .
?ea rdf:type
<http://dbpedia.org/class/yago/BuildingsAndStructuresInLondon> .
?ea ?ec ?ed .
} UNION {
?pa ?pb <http://dbpedia.org/resource/London> .
?pa rdf:type <http://dbpedia.org/class/yago/PalacesInLondon> .
?pa ?pc ?pd .
} UNION {
?pla ?plb <http://dbpedia.org/resource/London> .
?pla rdf:type <http://dbpedia.org/class/yago/PlazasInLondon> .
?pla ?plc ?pld .
} UNION {
?pl2a ?pl2b <http://dbpedia.org/resource/London> .
?pl2a rdf:type
<http://dbpedia.org/class/yago/RedevelopedPortsAndWaterfronts> .
?pl2a ?pl2c ?pl2d .
} UNION {
?la ?lb <http://dbpedia.org/resource/London> .
?la rdf:type <http://schema.org/LandmarksOrHistoricalBuildings>
.
?la ?lc ?ld .
} UNION {
?p3a ?p3b <http://dbpedia.org/resource/London> .
?p3a rdf:type <http://dbpedia.org/ontology/Place> .
?p3a ?p3c ?p3d .
} UNION {
?moa ?mob <http://dbpedia.org/resource/London> .
?moa rdf:type
<http://es.dbpedia.org/resource/Categor%C3%ADa:Arquitectura_de_la_provincia_de_London> .
?moa ?moc ?mod .

```

```
} UNION {  
?pea ?peb <http://dbpedia.org/resource/London> .  
?pea rdf:type <http://dbpedia.org/ontology/Person> .  
?pea ?pec ?ped .  
}  
UNION {  
<http://dbpedia.org/resource/London> ?pe2b ?pe2a .  
?pe2a rdf:type <http://dbpedia.org/ontology/Person> .  
?pe2a ?pe2c ?pe2d .  
}}
```


9.2. Glosario de términos

API

Interfaz de programación de aplicaciones (Application programming interface), es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por otro software como una capa de abstracción.

Back-end

Es la parte del software que procesa la entrada desde el front-end.

Código abierto

Es la expresión con la que se conoce al software distribuido y desarrollado libremente.

Crawler

Programa que inspecciona contenidos de forma metódica y automatizada.

Endpoint

Es una interfaz expuesta por un emisor o por un canal de comunicación. Facilita un estándar programable donde los sistemas y subsistemas de software pueden comunicarse entre sí.

Framework

Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

Front-end

Es la parte del software que interactúa con el usuario.

GPS

Sistema global de posicionamiento (Global Positioning System) es el sistema global de navegación por satélite que permite determinar en todo el mundo la posición de un objeto, una persona o un vehículo.

HTML

Lenguaje de marcas de hipertexto (HyperText Markup Language), hace referencia al lenguaje de marcado para la elaboración de páginas web.

HTTP

Protocolo de transferencia de hipertexto (Hypertext Transfer Protocol) es el protocolo usado en cada transacción de la World Wide Web.

iOS

Es un sistema operativo móvil de la empresa Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV.

Java

Lenguaje de programación originalmente desarrollado por Sun Microsystems, adquirida por Oracle, para aplicaciones software independientes de la plataforma.

SDK

Kit de desarrollo de software (Software development kit), es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto.

URI

Identificador uniforme de recursos (Uniform Resource Identifier), es una cadena de caracteres corta que identifica inequívocamente un recurso.

W3C

World Wide Web Consortium, es un consorcio internacional que produce recomendaciones para la World Wide Web.

WIFI

(Wireless Fidelity) Es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica.

World Wide Web

Sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet.

XML

Lenguaje de marcas extensible (eXtensible Markup Language), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.