



---

# Universidad de Valladolid

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE  
TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN

## Diseño e implementación de una aplicación distribuida semántica para el apoyo del aprendizaje ubicuo relacionado con el patrimonio cultural

Autor

**Pablo García Zarza**

Tutores

**Dr. Miguel Luis Bote Lorenzo**

**Dr. Guillermo Vega Gorgojo**

Valladolid, 22 de septiembre de 2021

---

TÍTULO: **Diseño e implementación de una aplicación distribuida semántica para el apoyo del aprendizaje ubicuo relacionado con el patrimonio cultural**

AUTOR: **Pablo García Zarza**

TUTOR: **Dr. Miguel Luis Bote Lorenzo  
Dr. Guillermo Vega Gorgojo**

DEPARTAMENTO: **Tratamiento de la Señal y Comunicaciones e Ingeniería Telemática**

---

## TRIBUNAL

---

PRESIDENTE: **Dr. Juan Ignacio Asensio Pérez**

VOCAL: **Dr. Manuel Rodríguez Cayetano**

SECRETARIO: **Dr. Juan Pablo de Castro Fernández**

---

FECHA: **29 de septiembre de 2021**

CALIFICACIÓN:

---

## RESUMEN

Múltiples organizaciones publican datos abiertos de patrimonio cultural. Normalmente estos datos, aunque son una excelente fuente de información, no están preparados para ser utilizados directamente en la educación. Por este motivo, en trabajos previos del grupo de investigación GSIC/EMIC se generó de manera semiautomática un conjunto de tareas educativas basándose en diferentes fuentes de datos abiertos de patrimonio y en plantillas validadas por docentes. El conjunto de datos generado –ideado para ser utilizado por estudiantes de los últimos cursos de Educación Secundaria y bachillerato– fue publicado como un repositorio de triplas RDF.

Los estudiantes de los cursos a los que les podría resultar interesante la información del repositorio no suelen conocer las tecnologías relacionadas con la Web Semántica por lo que no podrán acceder a los datos generados. Para solventar este problema en este Trabajo Fin de Máster se propone la aplicación distribuida *Casual Learn* que permite a los estudiantes aprovechar la información que está contenida en el almacén de triplas de forma transparente. Esta aplicación se ha dividido en un servidor y un cliente. El servidor trabajará como una pasarela entre el cliente y Casual Learn SPARQL (almacén de triplas) traduciendo las peticiones que se realicen desde el cliente a solicitudes SPARQL y además almacenará la información de la aplicación que sea privada. Para que el cliente pueda solicitar información al servidor este expone una API RESTful. El cliente se puede instalar en dispositivos móviles Android y cuenta con dos modos de funcionamiento: el primero de ellos permite a los estudiantes la visualización de la información del repositorio a través de un mapa interactivo; en el segundo el cliente busca de forma pasiva lugares de interés cercanos a la posición del usuario para notificarle de su existencia. Independientemente del modo de funcionamiento utilizado, el cliente permite al estudiante realizar tareas educativas de manera contextualizada. Cuando el estudiante finalice una tarea el cliente envía la respuesta al servidor. De esta forma, y si el estudiante lo permite, sus respuestas pueden ser consultadas por cualquier persona a través de la API del servidor mediante un portafolio público.

Casual Learn se puede utilizar para apoyar el aprendizaje ubicuo de patrimonio cultural. Prueba de ello son las diferentes experiencias educativas realizadas con esta aplicación, habiendo sido utilizada en estas experiencias por más de 100 estudiantes que cursaban el último curso de Educación Secundaria Obligatoria y primero de bachillerato. El cliente de la aplicación se encuentra disponible en Google Play para quien quiera aprender sobre el patrimonio cultural de Castilla y León, España.

## PALABRAS CLAVE

Patrimonio cultural, Web Semántica, aplicación distribuida, aprendizaje ubicuo, Android

## ABSTRACT

Many organizations publish open cultural heritage data. Usually these data, although they are an excellent source of information, are not ready to be used directly in education. For this reason, in previous work by GSIC/EMIC research group, a set of learning tasks was generated semiautomatically using different open information sources and templates validated by teachers. The generated dataset –designed to be used in education by high school students– was published as a repository of RDF store.

Students of courses that could be interested in this information do not usually know the technologies related with Semantic Web, so they will not be able to access to the dataset generated. To solve this problem, this Master’s Degree proposes the distributed application *Casual Learn* that allows students to take advantage of the information stored in the triplestore in a transparent way. This application has been split into a server and a client. Server works as a gateway between client and Casual Learn SPARQL (RDF store) by translating client requests into SPARQL requests and stores private application information. For the client to request information from the server, it exposes a RESTful API. Client can be installed on Android mobile devices and has two operation modes: the first allows students to view the repository information through an interactive map; in the second, the client passively looks for point of interest near to the user’s position to notify them. Regardless of the operating mode used, client allows the student to do contextualized learning tasks. When a user completes a task, client sends the response to the server. In this way, and if the student allows it, its answers can be consulted by anyone using the server’s API through a public portfolio.

Casual Learn can be used to support the ubiquitous learning of cultural heritage. Proof of this are the learning experiences carried out with this application, having been used by more than 100 high school students. The application client is available in Google Play for anyone who wants to learn about the cultural heritage of Castile and Leon, Spain.

## KEYWORDS

Cultural heritage, Semantic Web, distributed application, ubiquitous learning, Android

# Agradecimientos

Quiero expresar mi gratitud a todos los miembros del grupo de investigación GSIC/EMIC por brindarme la oportunidad de realizar este trabajo. Sin vuestros consejos, apoyo y comprensión habría sido mucho más complicado conseguir un resultado como el alcanzo. En especial quiero destacar a mis tutores por haberme guiando en cada etapa del proceso. Muchas gracias.

También quiero agradecer a mi familia y amigos por todo el apoyo que me han ofrecido y me ofrecen diariamente. Gracias a vosotros siento que cada día puedo llegar un poquito más lejos.



# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Objetivos . . . . .	2
1.2. Metodología . . . . .	3
1.3. Estructura del documento . . . . .	4
<b>2. Estado del arte</b>	<b>7</b>
2.1. Introducción . . . . .	7
2.2. Aplicaciones para apoyar el aprendizaje ubicuo de patrimonio cultural . . . . .	8
2.3. La Web Semántica y su utilización para el apoyo de aprendizaje ubicuo de patrimonio cultural . . . . .	11
2.3.1. Casual Learn SPARQL . . . . .	14
2.4. Conclusiones . . . . .	17
<b>3. Análisis</b>	<b>19</b>
3.1. Introducción . . . . .	19
3.2. Ejemplo de uso de la aplicación . . . . .	20
3.3. Historias de usuario . . . . .	21
3.4. Casos de uso del sistema . . . . .	23
3.5. Requisitos no funcionales . . . . .	28
3.6. Discusión de la visión general del sistema . . . . .	29
3.7. Conclusiones . . . . .	32
<b>4. Diseño</b>	<b>33</b>
4.1. Introducción . . . . .	33
4.2. Arquitectura del sistema . . . . .	33
4.3. Diseño del servidor . . . . .	36
4.3.1. Diseño de la API RESTful . . . . .	36
4.3.2. Peticiones SPARQL . . . . .	39
4.3.3. El portafolio de la aplicación . . . . .	41
4.4. Diseño del cliente . . . . .	43
4.4.1. Búsqueda activa de POI . . . . .	43
4.4.2. Búsqueda pasiva de contextos . . . . .	46
4.4.3. Realización de tareas educativas y edición de respuestas . . . . .	46
4.5. Conclusiones . . . . .	47
<b>5. Implementación</b>	<b>49</b>
5.1. Introducción . . . . .	49
5.2. Implementación del servidor . . . . .	50
5.2.1. Persistencia de datos . . . . .	54
5.2.2. Servicio de recomendación: Recombee . . . . .	55

5.3. Implementación del cliente . . . . .	57
5.3.1. Identificación del usuario y configuración del portafolio . . . . .	59
5.3.2. Búsqueda activa de POI . . . . .	61
5.3.3. Búsqueda pasiva de POI . . . . .	63
5.3.4. Realización y edición de una tarea . . . . .	64
5.3.5. Otras pantallas del cliente . . . . .	67
5.3.6. Evolución de la interfaz gráfica del cliente . . . . .	68
5.4. Comunicaciones entre los servicios que forman la aplicación . . . . .	70
5.5. Conclusiones . . . . .	73
<b>6. Evaluación del sistema</b>	<b>75</b>
6.1. Introducción . . . . .	75
6.2. Rendimiento del servidor . . . . .	75
6.3. Evaluación de la aplicación . . . . .	78
6.3.1. Pilotos realizados con la aplicación . . . . .	83
6.4. Conclusiones . . . . .	84
<b>7. Conclusiones y líneas de trabajo futuro</b>	<b>87</b>
7.1. Conclusiones del trabajo realizado . . . . .	87
7.2. Limitaciones y líneas de trabajo futuro . . . . .	90
<b>Referencias</b>	<b>96</b>
<b>A. Documentación de la API del servidor</b>	<b>97</b>
<b>B. Plantillas del portafolio</b>	<b>107</b>



# Índice de figuras

2.1.	Capturas de pantalla de aplicaciones relacionadas con el patrimonio cultural. (a) Captura de la aplicación « <i>Google Arts &amp; Culture</i> ». (b)(c) Capturas de pantalla de la aplicación « <i>World Heritage – Lista del Patrimonio de la UNESCO</i> ». En (a) y (b) se muestran dos mapas donde cada aplicación ha colocado una serie de marcadores para resaltar lugares de interés. En (c) se resaltan las distintas listas en las que se puede caracterizar el contenido de la aplicación. . . . .	9
2.2.	Capturas de pantalla de aplicaciones con las que se puede apoyar el aprendizaje ubicuo del patrimonio cultural. . . . .	10
2.3.	Representación de la ontología de Casual Learn SPARQL . . . . .	15
3.1.	Casos de uso de la aplicación que se va a desarrollar. Con la silueta de una persona se representan a los actores de la aplicación. La línea discontinua separa a estos actores del sistema que se va a crear. . . . .	24
3.2.	Arquitectura física básica de la aplicación distribuida que se va a desarrollar (color de fondo azul). El cliente utilizará la API REST del servidor para obtener información y almacenar las respuestas de los estudiantes. El servidor realizará solicitudes SPARQL a Casual Learn SPARQL. . . . .	31
4.1.	Arquitectura lógica de la aplicación Casual Learn. En este TFM se desarrollan el cliente y el servidor de Casual Learn (en fondo verde). La descripción de las operaciones de la API REST se muestran en la Tabla 4.2. Los servicios externos que se utilizarán se representan con un rectángulo con el fondo de color gris. Casual Learn SPARQL representa el repositorio de triplas donde está almacenada la información de POI y tareas educativas. . . . .	34
4.2.	Operaciones soportadas por el servidor de Casual Learn. Se utilizan URL relativas [BFM05] para indicar la ruta al recurso. Los valores entre corchetes significan que son parámetros opcionales. Los valores entre llaves son obligatorios y deberán sustituirse por el dato que corresponda. . . . .	37
4.3.	Maqueta del portafolio de la aplicación Casual Learn. Lista de respuestas públicas realizadas por un usuario. En esta lista se podrá ver el número de tareas completadas por tipo esperado de respuesta. . . . .	41
4.4.	Maqueta del portafolio de la aplicación Casual Learn. Vista individual de una respuesta. En la vista individual se podrá ver el POI donde se inició la tarea y su imagen descriptiva. . . . .	42

- 4.5. Maquetas de las pantallas más relevantes del cliente Android de Casual Learn. Para su diseño se ha decidido utilizar la escala de grises para destacar el formato general de cada una de las pantallas. En el Capítulo 5 se mostrarán los colores que se han elegido para el cliente en producción. (a) Pantalla de mapas donde los usuarios podrán consultar los POI existentes en una zona. Al pulsar sobre un POI aparecerá la lista de tareas educativas asociadas a este. El mapa mostrado ha sido generado mediante `city-roads` (<https://github.com/anvaka/city-roads>) utilizando datos de OpenStreetMap [OSM]. (b) Pantalla donde el usuario realizará la tarea de aprendizaje seleccionada. La interfaz de esta pantalla se adaptará al tipo esperado de respuesta para cada tarea educativa. (c) Pantalla donde el usuario podrá valorar la tarea educativa (si lo desea) y compartir la respuesta proporcionada en las distintas redes sociales (RRSS) y entornos educativos soportados por la aplicación. (d) Pantalla con la lista de tareas completada por el usuario. Se proporciona información suficiente (tipo esperado de respuesta, nombre del POI e instante de realización) para que el usuario pueda diferenciar las distintas respuestas. . . . . 44
- 4.6. Ejemplificación de la simplificación que realiza el cliente de Casual Learn para dividir el mapamundi en teselas cuadradas de tamaño constante. El origen desde donde se inicia la división del mapa en teselas será fijo para el cliente desarrollado en este TFM. En el caso del ejemplo representado, el cliente deberá solicitar al servidor como máximo la información de los POI de 12 teselas. La totalidad o parte de estas teselas se estarían mostrando en la pantalla del dispositivo móvil. Si el cliente dispone de la información de alguna de las teselas en la memoria local del dispositivo del usuario –y esta información continúa siendo válida– no tiene que realizar la solicitud de los POI de esta tesela al servidor. . . . . 45
- 5.1. Capturas de pantalla de paginas de un portafolio de ejemplo. Las capturas han sido realizadas con una tableta que dispone de una resolución de pantalla de  $768 \times 1024$  px. . . . . 53
- 5.2. (a) Pantalla de identificación y (b) selección de cuenta. (c) Mensaje explicativo del motivo por el que se solicita el permiso para obtener la posición del usuario en todo momento. (d) Configuración de la visibilidad del portafolio y el retardo en la publicación. . . . . 60
- 5.3. (a) Pantalla que muestra el mapa de la zona consultada por el usuario. Se están mostrando marcadores porque existen POI en la zona que se está consultando y porque el nivel de zum es lo suficientemente alto (se muestra menos región del mapa). (b) Información de un POI con la lista de tareas educativas asociadas a este. (c) Búsqueda de un municipio de Castilla y León para centrar el mapa en su posición. . . . . 62
- 5.4. Notificación expandida (a) y compacta (b) de un POI cercano. (c) Información del POI cercano junto con la lista de tareas en el orden personalizado para el usuario. . . . . 64
- 5.5. (a) Pantalla con toda la información disponible de la tarea educativa. (b) Navegador interno mostrando la descripción de uno de los términos que aparecen en el enunciado de la tarea. (c) Posición del usuario (representada por un triángulo marrón) frente a la ubicación del POI. . . . . 65

5.6.	(a) Pantalla de realización de una tarea en la que el estudiante ha contestado a una pregunta y ha realizado una fotografía. (b) Diálogo para reportar algún error en la tarea que esté realizando el estudiante. (c) Pantalla que se le muestra a un estudiante para que valore la tarea educativa que acaba de realizar. (d) Pantalla para compartir la respuesta en diferentes redes sociales y entornos educativos externos a la aplicación. . . . .	66
5.7.	(a) El usuario ha completado todas las tareas educativas de un POI por lo que se modifica su icono. (b) Lista de respuestas que ha generado el usuario. (c) Pantalla para la visualización y edición de la respuesta. . . . .	67
5.8.	(a) Menú de la aplicación accesible desde la pantalla de búsqueda activa de POI. (b) Lista de ejemplo con los POI que se le han notificado al estudiante. (c) Pantalla para cambiar las preferencias de la aplicación. (d) Pantalla de más información de Casual Learn. . . . .	68
5.9.	Evolución de la interfaz gráfica de la aplicación. . . . .	69
5.10.	Arquitectura física de la aplicación Casual Learn. Interacción entre las partes desarrolladas y con los servicios de terceros . . . . .	70
5.11.	Interacciones entre las distintas partes que forman Casual Learn. (a) Obtención de POI o tareas educativas. (b) Obtención de tareas personalizadas asociadas a un POI. (c) Un usuario completa por primera vez una tarea. (d) Identificación de un usuario (azul) y obtención del portafolio de un usuario (negro). . . . .	73
6.1.	Representación de la consistencia del servidor a distintas peticiones. . . . .	77
6.2.	Representación de la consistencia del servidor a las peticiones de solicitud de las tareas asociadas a un POI y personalizadas para un usuario. . . . .	78
6.3.	Gráfico en la que se indica el número de usuarios con la aplicación instalada, el número de adquisiciones de usuario y el número de desinstalaciones. Las series de adquisiciones y desinstalaciones muestran el número de eventos diarios. La serie de usuarios con la aplicación instalada muestra en cada día el número total de usuarios que tienen la aplicación instalada. El periodo temporal con el fondo amarillo corresponde con las fechas de la aparición de Casual Learn en prensa y en los canales oficiales de la Universidad de Valladolid. El periodo con el fondo naranja corresponde con el periodo de desarrollo de los distintos pilotos donde destacan las instalaciones y desinstalaciones que se producen al inicio y al final de estos. . . . .	80
6.4.	Gráficos de eventos producidos en Casual Learn recogidos con Google Analytics. En cada uno de los gráficos se muestra el número de veces por día que se produjo el evento. . . . .	81
6.5.	Gráfico con los fallos diarios que se han producido con Casual Learn desde que se encuentra en producción junto con el día en el que se lanzó un nueva versión de la aplicación a través de Google Play. . . . .	82



# Índice de tablas

2.1. Prefijos y namespaces utilizados en el documento . . . . .	13
2.2. Propiedades presentes en los POI almacenados en Casual Learn SPARQL. . . . .	15
2.3. Propiedades presentes en las tareas almacenadas en Casual Learn SPARQL. . . . .	16
2.4. Propiedades presentes en los temas almacenados en Casual Learn SPARQL. . . . .	16
3.1. Narrativa de las historias de usuario de Casual Learn junto con un identificador (ID) y prioridad que tiene para el usuario. Se omite la parte inicial de la estructura que debe seguir una historia de usuario puesto que todas empiezan por « <i>Como un estudiante, quiero...</i> ». . . . .	21
3.2. Requisitos no funcionales de Casual Learn. . . . .	28
4.1. Descripción de los elementos que forman parte de la arquitectura representada en la Fig. 4.1 . . . . .	35
4.2. Descripción de las operaciones que se pueden realizar con el servidor de Casual Learn. . . . .	38
6.1. Resumen de las métricas analizadas en el servidor de Casual Learn, siendo: <i>POI</i> la solicitud de los POI dentro de una tesela; <i>Tareas</i> la solicitud de las tareas asociadas a un POI; <i>TareasP</i> la solicitud de las tareas personalizadas de un POI para un usuario (interviene el servicio de recomendación). . . . .	76
6.2. Resumen de usuarios que han utilizado el cliente. . . . .	79
6.3. Eventos que se han registrado con Casual Learn. . . . .	80
6.4. Resumen de las experiencias educativas que han utilizado Casual Learn en la que que indica el tema del que trabaja el piloto, el número de estudiantes que lo componían, las fechas de inicio y fin del piloto y el número de tareas que completaron. Estos datos se han obtenido de [Rui+21a; Rui+21d]. . . . .	83



# Índice de listados

2.1.	Ejemplo de dos triplas escritas en formato N-Triples. Es un extracto de la información almacenada en [GSICa]. . . . .	12
2.2.	Ejemplo de dos triplas escritas en formato N3. Es un extracto de la información almacenada en [GSICa]. . . . .	13
2.3.	Ejemplo de consulta SPARQL. . . . .	13
2.4.	Petición SPARQL para obtener el nombre y posición de norte a sur de las catedrales almacenadas en Casual Learn SPARQL. . . . .	16
3.1.	Estructura que debe seguir una historia de usuario. . . . .	21
4.1.	Plantilla de la consulta SPARQL para obtener los POI contenidos en una zona. Con los parámetros utilizados se recupera el identificador del POI (?contexto), la ubicación del POI mediante su latitud (?lat) y longitud (?lng), el autor de los datos (?creadoPor), el número de tareas educativas asociadas al POI (?nTareas), la descripción del lugar (?com) y el enlace de DBpedia (?wiki). . . . .	39
4.2.	Plantilla SPARQL para solicitar las tareas educativas asociadas a un POI. Los parámetros que se obtendrán en la respuesta son el IRI de la tarea (?task), el tipo esperado de respuesta (?tR), la URL de la imagen en baja calidad (?thumb), la URL de la imagen en alta calidad (?imag), el autor (?imgA) y la licencia (?imgL) de la imagen, el nombre del POI donde se realiza la tarea (?com), el enunciado de la tarea (?rAT), la respuesta correcta (?rE) y el autor de la tarea (?cP). . . . .	40
4.3.	Solicitud SPARQL para obtener todas las tareas de Casual Learn SPARQL. Los parámetros de la respuesta serán el IRI de la tarea educativa (?task), la ubicación donde se encuentra el POI con el que está relacionado (?lat y ?lng) y el tipo esperado de respuesta (?tR). . . . .	40
5.1.	Estructura de paquetes y ficheros del código del servidor de Casual Learn. . . . .	51
5.2.	Código de ejemplo para mostrar la construcción de bucles y las sentencias if-else en Apache Velocity. . . . .	53
5.3.	Se activa la característica por la que solo los dispositivos que puedan utilizar un sistema hardware de navegación por satélite sean capaces de ejecutar el cliente de Casual Learn. . . . .	57
5.4.	Estructura del proyecto del cliente de Casual Learn. . . . .	58
5.5.	Extracto del código para el envío de la solicitud de creación de un portafolio al servidor. . . . .	60
A.1.	Fichero en formato YAML con el que se genera la documentación de la API RESTful del servidor . . . . .	97
B.1.	Plantilla para la respuesta de un portafolio en el que el usuario todavía no haya registrado ninguna respuesta . . . . .	107
B.2.	Plantilla para la respuesta de un portafolio en el que el usuario tenga al menos una respuesta en el servidor . . . . .	109
B.3.	Plantilla para la respuesta de una solicitud de respuesta individual de un usuario . . . . .	111





# Capítulo 1

## Introducción

La UNESCO<sup>1</sup> (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura) define el patrimonio cultural [UNE] como «el legado cultural que recibimos del pasado», que se disfruta en el presente y que se debe transmitir a las futuras generaciones. Para disfrutar de este legado y poder transmitirlo es necesario que las personas lo conozcan. El conocimiento del patrimonio cultural está incluido como un objetivo dentro del currículo de enseñanza obligatoria español [BOE15, Artículo 11j]. En la asignatura de Historia del Arte –que es impartida a estudiantes de segundo de bachillerato– se propone que se comience el estudio de la asignatura con la cultura grecorromana y se llegue hasta el arte contemporáneo, destacando de manera general las características más importantes de cada periodo y analizando un conjunto limitado de obras de arte [BOE15]. Esta generalización puede provocar que los estudiantes tengan dificultades para relacionar los conceptos vistos de manera teórica en la asignatura con el patrimonio cultural que se encuentre en su entorno.

La omnipresencia de la tecnología en todos los aspectos del día a día ha permitido que se desarrolle una nueva metodología educativa llamado aprendizaje ubicuo [Gal16]. Este tipo de aprendizaje puede ayudar a que los estudiantes identifiquen con más facilidad las características de un determinado monumento al visitarlo de manera presencial en vez de aprender sus características de manera teórica a través de un libro [Pis+20]. Históricamente, desde los centros de enseñanza se han fomentado las visitas culturales, bien en formato de actividad grupal o simplemente como una recomendación. Las visitas grupales se suelen realizar durante las horas lectivas por lo que la disponibilidad de los estudiantes es máxima, pero tienen asociadas una serie de desventajas de carácter económico y logístico.

Los estudiantes que decidan realizar una visita cultural de manera individual, o en pequeños grupos, pueden necesitar ayuda para comprender los detalles del lugar visitado y conectarlos con los estudiados en clase. Esta conexión se llevaría a cabo a través del aprendizaje ubicuo apoyándose en algún dispositivo electrónico. En España, el 92,8 % de los niños de 14 años han dispuesto de un móvil durante los últimos tres meses y este porcentaje aumenta para los de 15 años hasta llegar al 95,7 % [INE20]. Incrementando el rango de edad, el 97 % de los jóvenes de entre 16 y 24 años han accedido a Internet diariamente [INE20]. Por este motivo, la utilización de las TIC (Tecnologías de la Información y la Comunicación) para apoyar una visita relacionada con el patrimonio cultural no debería suponer para los jóvenes españoles un problema.

Múltiples organismos estatales proporcionan información relacionada con el patrimonio cultural a través de datos abiertos expuestos en sus portales. Ejemplos de estos portales

---

<sup>1</sup><https://es.unesco.org>

podrían ser Datos Abiertos de Castilla y León<sup>2</sup>, Open Data Euskadi<sup>3</sup>, datos.gob.es<sup>4</sup> o EU Open Data Portal<sup>5</sup>. Estos sitios web exponen la información en formatos difíciles de manejar directamente por personas que no estén acostumbradas a utilizar estas tecnologías. Además, estas plataformas no suelen estar diseñadas para ser utilizadas como herramientas para apoyar el aprendizaje de alumnos de instituto. Por estos motivos los estudiantes pueden desistir de obtener la información directamente de estas plataformas oficiales y utilizar otras fuentes menos contrastadas.

Gracias a que diversos organismos publican los datos de manera abierta, el Grupo de Investigación GSIC/EMIC [GSICb] ha presentado en [Rui+21c] un conjunto de datos creados de manera semiautomática destinado a apoyar el aprendizaje del patrimonio cultural de la comunidad autónoma de Castilla y León, España. Se han combinado datos abiertos de distintas fuentes como DBpedia, Wikidata y la plataforma de Datos Abiertos de la Junta de Castilla y León. Los datos se dividen en dos grandes grupos: puntos de interés y tareas de aprendizaje. En Casual Learn SPARQL [GSICa], nombre que se le ha dado al repositorio donde este almacena la información, se pueden encontrar más de 10 000 tareas de aprendizaje asociadas a más de 2000 puntos de interés. Los puntos de interés están geolocalizados en lugares relevantes del patrimonio cultural de Castilla y León. Las tareas de aprendizaje resaltan elementos específicos del lugar para facilitar su aprendizaje. Estas tareas educativas han sido diseñadas considerando el currículo educativo de Castilla y León y siendo avaladas por profesores de Educación Secundaria de institutos de esta región española [Rui+21c].

La forma de consultar la información almacenada en Casual Learn SPARQL es mediante consultas de tipo SPARQL (*SPARQL Protocol and RDF Query Language*), un lenguaje estandarizado para realizar consultas a grafos RDF (*Resource Description Framework*), que describe la forma de representar información en la Web [HS13; CWL14]. Estas consultas se realizan sobre un punto SPARQL, un sitio Web que permite realizar consultas a un repositorio tanto a personas como a máquinas utilizando el lenguaje SPARQL [EUOD17]. La interacción con el punto SPARQL para obtener los datos de patrimonio puede ser poco agradable y de un grado de complejidad elevado para usuarios que no sean expertos en tecnologías semánticas, como les pasará a la mayoría de estudiantes que cursen los últimos cursos de Educación Secundaria Obligatoria. Este es el problema que motiva el desarrollo del Trabajo Fin de Máster (TFM) que se va a presentar. Con él se pretende facilitar la obtención de los datos de este repositorio para que sean utilizados por los últimos cursos de Educación Secundaria y bachillerato. Al conjunto de servicios que compongan la solución que se va a desarrollar se le denominará **Casual Learn**.

### 1.1. Objetivos

El principal objetivo de este TFM es la **creación de un sistema que disponga de una interfaz con la que los usuarios puedan utilizar los datos almacenados en el repositorio [GSICa] para apoyar el aprendizaje ubicuo sobre el patrimonio cultural de su entorno**. Para ello el sistema, además de proporcionar una interfaz gráfica con la que consultar la información almacenada en el punto SPARQL, deberá ofrecer información al usuario basándose en su localización para motivar un aprendizaje informal del patrimonio. Las respuestas que proporcionen los usuarios a las tareas de aprendizaje podrán ser públicas en un portafolio [CL07] si los usuarios lo desean. Debido a todas las funciones que debe realizar el sistema es necesario

---

<sup>2</sup><https://datosabiertos.jcyl.es/web/es/datos-abiertos-castilla-leon.html>

<sup>3</sup><https://opendata.euskadi.eus/inicio/>

<sup>4</sup><https://datos.gob.es/>

<sup>5</sup><https://data.europa.eu/euodp/en/home>

dividirlo en dos partes: una aplicación móvil que actúe de interfaz del sistema y un servidor que responda a las peticiones que se realicen desde la aplicación móvil con la información que obtenga del punto SPARQL. El servidor también deberá almacenar las respuestas de los usuarios para poder publicarlas en cada portafolio. El objetivo general se puede dividir en una serie de subobjetivos para facilitar su desarrollo e implementación.

1. Se debe **desarrollar un cliente que permita consultar la información almacenada en el repositorio de forma transparente para los alumnos**. La aplicación desarrollada deberá utilizar esta información en dos modos de funcionamiento diferentes: un **modo de búsqueda activa de información** y otro de **búsqueda pasiva**. Con estos dos modos de funcionamiento se pretende establecer una conexión entre lo que los alumnos aprendan en clase (aprendizaje formal) y el patrimonio de su entorno.
2. Se debe desarrollar un **servidor que permita responder a las peticiones de obtención de la información de puntos de interés, tareas de aprendizaje y portafolios**. La información de los puntos y las tareas las obtiene del punto SPARQL mientras que los portafolios los crea dinámicamente con las respuestas que el usuario haya proporcionado por lo que también debe permitir almacenar las respuestas. Se crean identificadores únicos para el portafolio de cada usuario además de para cada respuesta de cada usuario.
3. Se debe **comprobar el correcto funcionamiento y la usabilidad del sistema**. Por parte del servidor se crean unos ficheros de registros para comprobar que se responde correctamente a las peticiones de los usuarios donde también se almacena el tiempo de procesado de cada petición. Además, en la aplicación se registra un evento cada vez que el usuario interactúe con una tarea de aprendizaje o se le informe de un punto de interés cercano. La *usabilidad* del sistema se reportará a través de la realización de un cuestionario SUS [Bro96] (*System Usability Scala*) a los usuarios que hayan utilizado el sistema y quieran realizarlo.

## 1.2. Metodología

Para el desarrollo de este TFM se ha seguido el desarrollo ágil de *software*. Con esta forma de desarrollo se pretende obtener un producto finalizado de manera rápida que cumpla con los requisitos solicitados por el cliente [SS05]. Este tipo de desarrollo se basa en un manifiesto [Bec+01] formado por cuatro axiomas: los individuos y las interacciones sobre los procesos y las herramientas; el *software* en producción sobre la creación de documentación, con el que se establece que un producto acabado es el objetivo prioritario y si el código es lo suficientemente claro es autodescriptivo; mantener una comunicación constante con el cliente que realimente el producto frente a establecer un contrato fijo; y el último, adaptar el producto a los cambios de los requisitos y los problemas que surjan frente a seguir un plan fijo.

En las metodologías ágiles se pueden distinguir tres tipos de actores [SS05]: el **cliente**, que establece, de forma directa o indirecta, los requisitos del producto y valida cada una de las versiones del producto que se desarrolle; los **desarrolladores** que extraen los requisitos funcionales y no funcionales del cliente, además de recoger la realimentación del cliente de cada una de las versiones que se le muestre y desarrollar cada versión del producto; los **directores** se encargan de mantener unas interacciones productivas entre clientes y desarrolladores. Aplicando estos roles al desarrollo de este TFM, los clientes y directores serían los miembros del Grupo de Investigación donde se ha realizado este trabajo y el desarrollador sería el autor de este documento. Además, debido a los comentarios que han proporcionado los alumnos y profesores

de cada uno de los pilotos que han ido utilizando el sistema desarrollado en distintas fases de madurez, también pueden ser considerados como clientes. Gracias a estos comentarios se ha conseguido mejorar el sistema en cada fase de su desarrollo.

En las metodologías ágiles se suele programar ciclos de desarrollo de entre dos semanas y dos meses [SS05]. Al final el ciclo de desarrollo se realiza una reunión entre el cliente y los miembros del equipo de desarrollo para que se le haga una demostración de la versión finalizada y pueda aportar su realimentación. En lo que se refiere al desarrollo de este proyecto, los **ciclos de desarrollo** se establecían en **dos semanas finalizando con una reunión**. En estas reuniones se realizaba una pequeña demostración de las funcionalidades agregadas durante las últimas semanas y se definían los requisitos que tenían que desarrollarse para próximas iteraciones, además de aportar ideas para mejorar el resultado de la versión que se acababa de mostrar. Se podían establecer más comunicaciones entre el equipo de desarrollo y el cliente si se estaba próximo a publicar una versión en producción. Previamente a las reuniones se entregaba un pequeño informe donde se explicaban los cambios realizados y se intentaban justificar cada una de las decisiones que se habían tomado durante el desarrollo.

Durante las semanas de creación de una nueva versión el equipo de desarrollo siempre se siguen cuatro fases para obtener una nueva versión que presentarle al cliente. La primera de estas fases es el **análisis de los requisitos** solicitados por el cliente y el **estudio de soluciones existentes** en el mercado. La siguiente fase sería el **diseño** de la solución a presentar. A continuación se pasaría a la fase de desarrollo o **codificación del diseño realizado**. Antes de presentar la solución al cliente se realizan una serie de pruebas en la fase de **verificación** para comprobar que la solución es lo suficientemente robusta y cumple con lo solicitado por el cliente. Durante cualquiera de estas etapas se puede consultar al cliente para que ofrezca realimentación sobre algún punto concreto. En lo referente al desarrollo del TFM se han seguido las cuatro etapas agregando una pequeña fase a mayores para documentar tanto el código como las decisiones tomadas. Si en algún momento ocurría algún problema o se tenía alguna duda se contactaba con los miembros del GSIC para obtener su realimentación.

### 1.3. Estructura del documento

La documentación para estructurar este TFM se basa en cada una de las fases de la metodología empleada. Por este motivo, en el **capítulo 2 – Estado del arte** se define qué es el aprendizaje ubicuo y se estudia algunas de las aplicaciones relacionadas con el patrimonio cultural que existan en el mercado. También en este capítulo se indica cómo es el modelo de datos que utiliza Casual Learn SPARQL para estructurar la información de los puntos de interés y tareas educativas que se utilizan en el sistema que se va a desarrollar.

En el **capítulo 3 – Análisis** se transforman las ideas que tienen los clientes de lo que les gustaría que fuera Casual Learn a una serie de requisitos que se pueden implementar. En este capítulo se realiza una primera aproximación de cómo será la arquitectura de Casual Learn como consecuencia de los requisitos que hayan solicitado los clientes.

El siguiente capítulo que forma parte de este documento es el **capítulo 4 – Diseño**, donde se presenta la arquitectura lógica de Casual Learn. A partir de esta arquitectura se describen los dos componentes que se van a desarrollar: un servidor que se encargue de conseguir la información del punto SPARQL y de almacenar las respuestas de los usuarios y un cliente, que podrá instalarse en los dispositivos móviles de los usuarios, donde se representará la información de Casual Learn SPARQL y permitirá la realización de las tareas educativas. En este capítulo

también se muestran las maquetas de la interfaz gráfica que se desea obtener.

En **capítulo 5 – Implementación** se describe qué tecnologías se han utilizado para el desarrollo de cada parte de la aplicación y cómo es la organización de directorios y ficheros de cada uno de los proyectos. Además, en este capítulo se muestra la arquitectura física de Casual Learn y se describen una serie de ejemplos con posibles intercambios de información entre cada uno de los servicios.

En el **capítulo 6 – Evaluación del sistema** se comparten los datos actuales de la aplicación: los relacionados con el cliente (número de usuarios, número de tareas completadas, etc.) y las del servidor (tiempo que necesita el servidor en obtener la información sobre los POI de una zona y las tareas de uno POI).

El último capítulo que compondrá este TFM será el **capítulo 7 – Conclusiones y líneas de trabajo futuro** donde se explican los motivos por los que se han cumplido con los objetivos descritos en la Sección 1.1. Además, en este capítulo también se indican las limitaciones que tiene la versión del sistema presentado y las líneas que se podrían seguir para solucionar estos problemas.

Por último, a este documento le acompañaran dos anexos: **anexo A – Documentación de la API del servidor** y **Anexo B – Plantillas del portafolios**. En el primero de ellos se adjunta el fichero de texto necesario para poder generar el sitio web<sup>6</sup> de la documentación de las operaciones soportadas por el servidor. En el segundo anexo se han agregado las plantillas utilizadas para generar las páginas web del portafolio de Casual Learn. Las plantillas se completan con la información de las respuestas de los usuarios.

---

<sup>6</sup>[https://casuallearnapp.gsic.uva.es/server\\_api](https://casuallearnapp.gsic.uva.es/server_api)



## Capítulo 2

# Estado del arte

Conocer qué soluciones existen en el mercado de aplicaciones relacionadas con el aprendizaje del patrimonio cultural servirá para poder determinar qué características debe satisfacer Casual Learn y cuáles serán sus aspectos diferenciadores. En este capítulo se define qué se entiende por aprendizaje ubicuo y se realiza una pequeña exposición de aplicaciones que podrían servir para apoyar escenarios educativos relacionados con el patrimonio cultural. Además, el objetivo principal de este TFM indicaba que se tenían que utilizar datos del repositorio generado en trabajos anteriores del grupo de investigación donde se está llevando a cabo este trabajo. Por ello, en este capítulo se explica qué información se encuentra en este repositorio y cuál es su estructura organizativa. Al finalizar el capítulo se conocen posibles alternativas de presentar la información a los usuarios y qué información –y en qué formato– se encuentran los datos almacenados en Casual Learn SPARQL.

### 2.1. Introducción

La aplicación propuesta en este TFM estará diseñada como herramienta de apoyo al aprendizaje ubicuo (UL, *ubiquitous learning*). En [Gal16] se define UL como «aquél conocimiento que se da en cualquier lugar y en cualquier momento a través de herramientas tecnológicas». La utilización de la tecnología permite establecer una conexión entre el mundo físico y virtual [Gal16]. Esta conexión es la que permitirá a los estudiantes que utilicen Casual Learn a respetar y conocer el patrimonio cultural que les rodea.

La principal ventaja del aprendizaje ubicuo a otros modelos está en su omnipresencia [Can15]. Los estudiantes pueden estar aprendiendo nueva información en cualquier lugar y sin ser plenamente conscientes de que están utilizando un recurso educativo. Además, los dispositivos móviles utilizados en el aprendizaje ubicuo no se limitan a la utilización de recursos modernos sino que se pueden utilizar para acceder a recursos tradicionales (por ejemplo, un libro de Historia del Arte en formato electrónico que puede ser utilizado para que el estudiante conozca los fundamentos teóricos de algún estilo arquitectónico).

En trabajos previos del grupo de investigación GSIC/EMIC se propuso un repositorio de triplas con tareas educativas relacionadas con el patrimonio cultural que pueden ser utilizadas para el UL [Rui+21c]. La información contenida en el repositorio de triplas es un conjunto valioso de datos para los estudiantes, pero también puede ser útil para las máquinas. El motivo por el que las máquinas pueden utilizar este conjunto de datos es porque la forma de publicar esta información ha sido como Datos Enlazados [Ber06] (*Linked Data*, LD) en un formato abierto. Para que un conjunto de datos se considere enlazado se tienen que seguir una serie de princi-

pios [Ber06] con el propósito de que, al seguir un formato estándar, tanto ordenadores como personas puedan utilizarlos.

Integramente relacionado con los LD se encuentra la Web Semántica [W3CSW]. Esta versión de la web está concebida como una evolución de las anteriores y con ella se hace la transformación para que el contenido en vez de documentos sean datos. La Web Clásica estaba ideada para que su contenido –y las relaciones de este contenido– fueran entendidas preferentemente por las personas. Con la Web Semántica, utilizando los principios de LD, la información pasa a estructurarse para que pueda ser usada por ordenadores. La mayoría de personas necesitarán de una interfaz gráfica que les facilite el acceso a la información en cualquiera de los dos casos. Por ejemplo, cualquier persona que pueda acceder al portal de Wikipedia podrá obtener información sobre lo que es una exedra, pero puede que les resulte más complicado obtener esta información a través del punto SPARQL de DBpedia.

En la Sección 2.2 se darán a conocer algunas aplicaciones que se puedan utilizar para apoyar el aprendizaje ubicuo de patrimonio cultural. Con ello se espera tener una primera aproximación de las características más importantes que pudiera tener Casual Learn. Además, en la Sección 2.3 se proporcionará más información de las tecnologías relacionadas con la Web Semántica que se van a utilizar en este TFM y qué datos y la estructura de estos en Casual Learn SPARQL.

## 2.2. Aplicaciones para apoyar el aprendizaje ubicuo de patrimonio cultural

Como se ha indicado previamente, en el aprendizaje ubicuo se consigue acceder a los conocimientos a través de dispositivos electrónicos siendo los teléfonos móviles un buen ejemplo de este tipo de dispositivos [Gal16; Can15]. Para que los móviles puedan ser utilizados en el aprendizaje ubicuo se deben desarrollar una serie de herramientas que ofrezcan al usuario esta información simplificando el proceso que sea necesario para la obtención de los datos.

Debido al auge de los teléfonos inteligentes existe una constante creación de aplicaciones –tanto nativas para los sistemas operativos móviles como páginas web, que están diseñadas para adaptarse al tamaño, forma y resolución de las pantallas– entre las que se encuentran las relacionadas con el patrimonio cultural. Por ejemplo, en la Fig. 2.1 se muestran capturas de pantalla de dos aplicaciones: *World Heritage – Lista del Patrimonio de la UNESCO*<sup>1</sup> y *Google Arts & Culture*<sup>2</sup>. Estas aplicaciones están disponibles para dispositivos móviles que ejecuten los sistemas operativos Android o iOS. Además, a *Google Arts & Culture* también se puede acceder a través de su página web (Web App). Ambas aplicaciones muestran información sobre el patrimonio cultural mundial. Mientras que *World Heritage* usa la información de la lista de lugares y monumentos que componen la lista del Patrimonio Mundial de la UNESCO<sup>3</sup>, *Google Arts & Culture* no indica de dónde extrae la información que aparece en la aplicación.

En la Fig. 2.1a y en la Fig. 2.1b se muestra una forma muy popular de representar información geolocalizada: a través de una serie de marcadores sobre un mapa interactivo. Tanto *World Heritage* como *Google Arts & Culture* disponen de otro modo para visualizar la información. En este modo de funcionamiento se muestran los lugares de interés cercanos al usuario a través de una lista de tarjetas ordenadas por distancia (las más cercanas a la posición del usuario las colocan en primer lugar). La última característica a destacar de *World Heritage* es que permite

---

<sup>1</sup><https://www.appworldheritage.com/>

<sup>2</sup><https://artsandculture.google.com/>

<sup>3</sup><https://whc.unesco.org/es/list/>



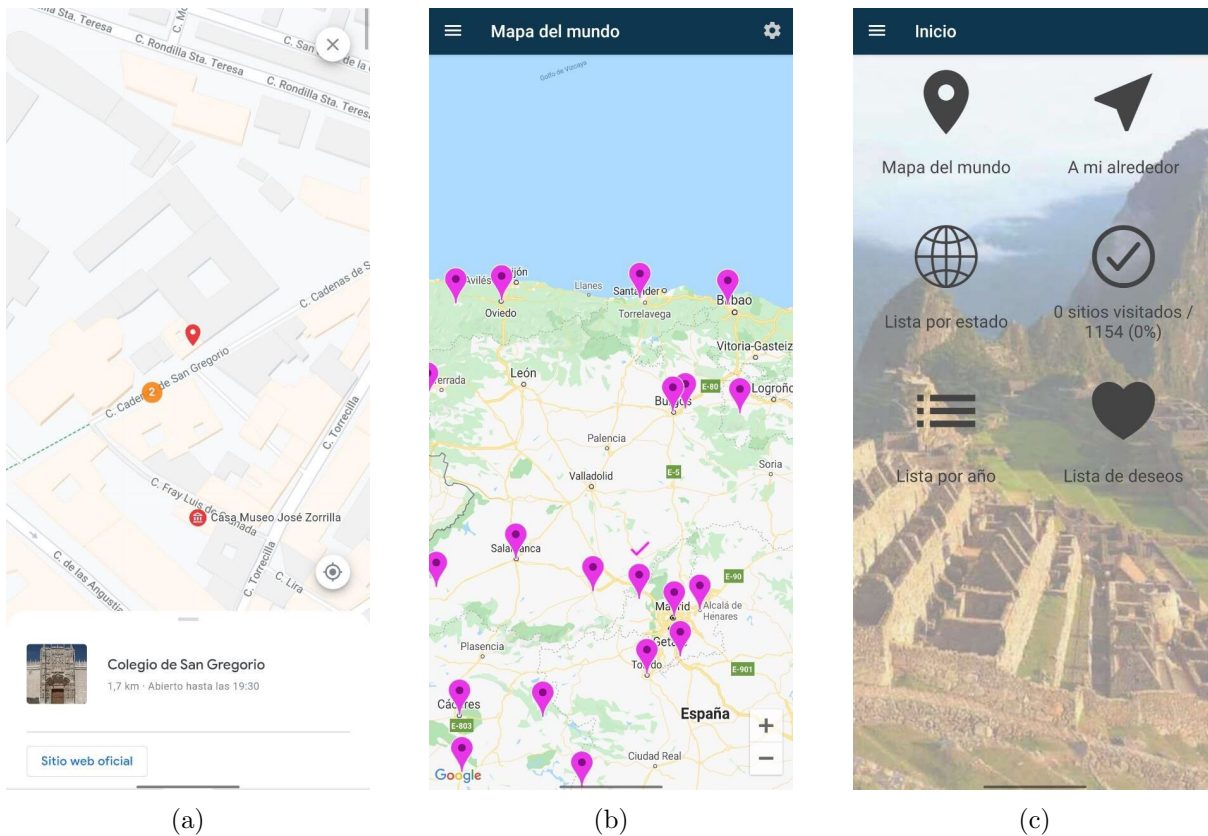


Fig. 2.1: Capturas de pantalla de aplicaciones relacionadas con el patrimonio cultural. (a) Captura de la aplicación «Google Arts & Culture». (b)(c) Capturas de pantalla de la aplicación «World Heritage – Lista del Patrimonio de la UNESCO». En (a) y (b) se muestran dos mapas donde cada aplicación ha colocado una serie de marcadores para resaltar lugares de interés. En (c) se resaltan las distintas listas en las que se puede caracterizar el contenido de la aplicación.

registrar los lugares que el usuario de la aplicación ha visitado. Al realizar esta acción se cambia el marcador con el que se identificaba el monumento por un icono de un *check*. Estas aplicaciones, aunque pueden utilizarse para obtener información de lugares relevantes del patrimonio cultural mundial, no cuentan con tareas educativas que se puedan utilizar en UL para que los estudiantes visiten y aprendan sobre el patrimonio de su entorno.

Existen otras aplicaciones con las que se puede obtener el patrimonio cultural de una determinada zona. Como ejemplo se puede resaltar a **Monumentos de Castilla y León**<sup>4</sup>. En esta página web se representan los monumentos de la comunidad autónoma de Castilla y León. La página permite filtrar los elementos que se muestran en el mapa por tipo de categoría, periodo histórico y estilo predominante del monumento (o lugar). La característica por la que se destaca esta aplicación es que los datos mostrados en esta página se recuperan de distintas fuentes de información que publican de manera abierta. Este sitio web se limita a proporcionar el lugar y la descripción en el que se encuentra el monumento, pero no solicita realizar ninguna acción al estudiante que le facilite a fijarse en los detalles.

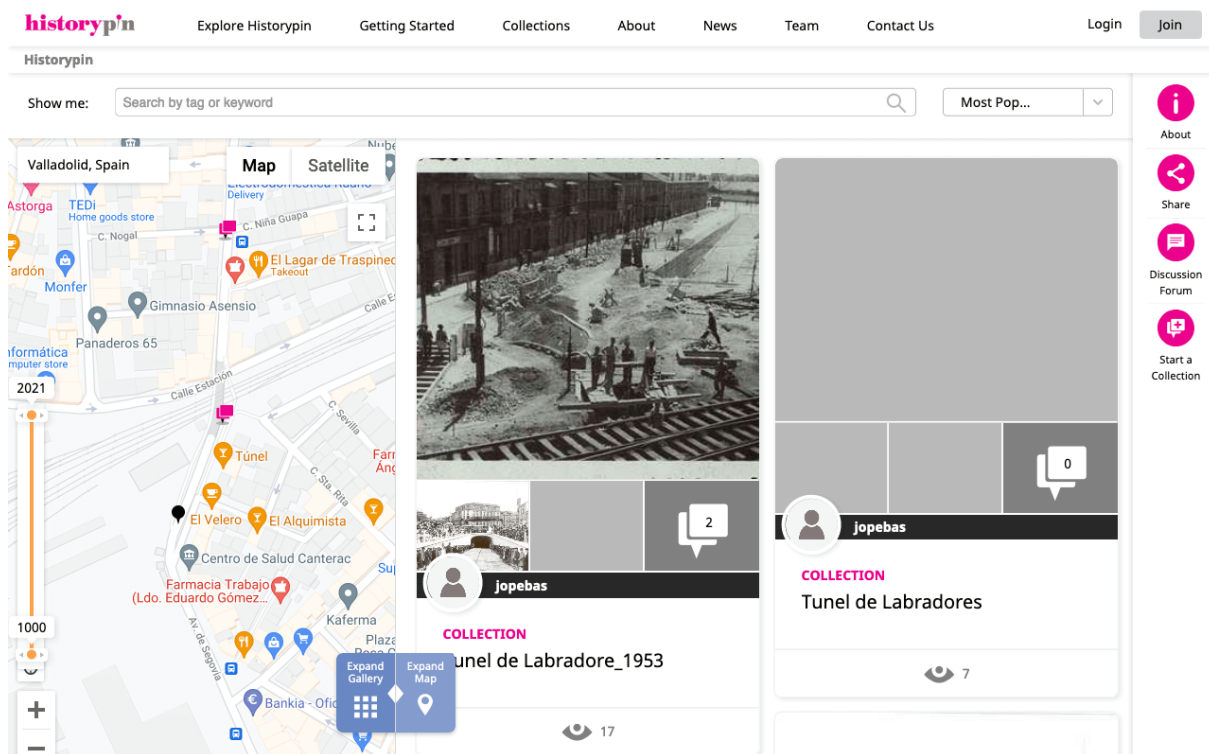
Otras aplicaciones como *historypin*<sup>5</sup> y *EduLoc*<sup>6</sup> (Fig. 2.2), aunque su tema principal no sea el patrimonio cultural, se pueden utilizar para apoyar experiencias educativas que estén centra-

<sup>4</sup><https://monumentoscyl.netlify.app/>

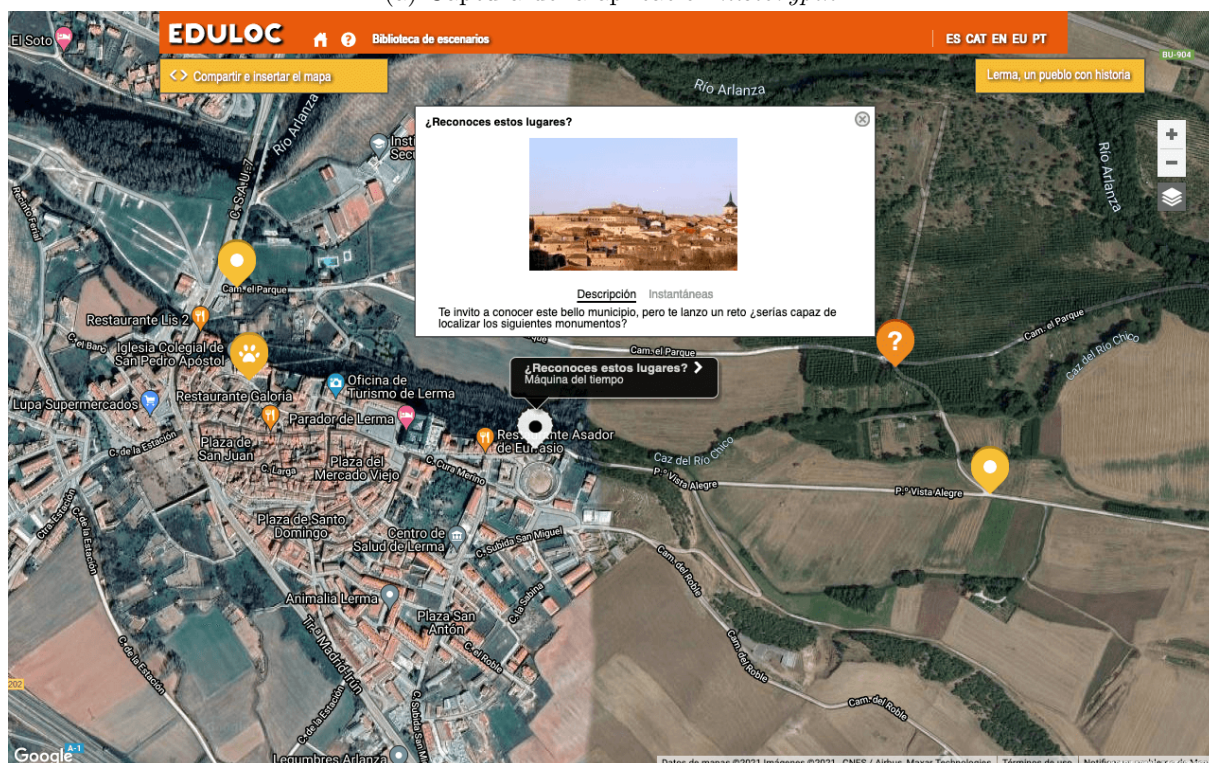
<sup>5</sup><https://www.historypin.org>

<sup>6</sup><http://www.eduloc.net>

## 2. ESTADO DEL ARTE



(a) Captura de la aplicación *historypin*.



(b) Captura de *Eduloc* con la ruta «Lerma, un pueblo con historia».

Fig. 2.2: Capturas de pantalla de aplicaciones con las que se puede apoyar el aprendizaje ubicuo del patrimonio cultural.

das en el patrimonio cultural. Las dos páginas buscan que su comunidad de usuarios generen el contenido que posteriormente se mostrará en ellas. *historypin* es una página para compartir

elementos multimedia e historias de manera geolocalizada y orientada al ámbito local (cada comunidad local genera el contenido de la zona que conoce o de la que dispone información). Los usuarios pueden realizar comentarios en las creaciones de otros usuarios. En la Fig. 2.2a se pueden ver fotografías de la construcción de uno de los túneles realizados en la ciudad de Valladolid para que sus habitantes tuvieran una forma segura de cruzar las vías del tren. Su interfaz muestra la información en dos zonas: un mapa donde el usuario puede ver los puntos donde otro usuario haya compartido contenido; una zona con la información (en formato tarjeta) de los puntos que se estén mostrando en el mapa. Un profesor podría utilizar esta página cargando nueva información en los puntos más relevantes del municipio donde esté ejerciendo la docencia para que sus estudiantes puedan conocer el patrimonio histórico mediante fotografías y texto. *historypin* permite utilizar la información que tenga disponible a través de una API, proporcionando atribuciones de manera adecuada<sup>7</sup> a su autor. La limitación de esta aplicación es que se necesita tener una comunidad activa de usuarios para que el sitio web tenga contenido.

*Eduloc* también busca que su comunidad de usuarios genere y comparta información geolocalizada con otros usuarios. El autor de contenido que utilice esta página web podrá generar un escenario (proyecto) donde establecer una serie de marcadores sobre una área concreta del mapa. A estos marcadores –al igual que en *historypin*– se les podrá asignar elementos multimedia y texto como información para el usuarios. Los marcadores también se pueden utilizar para generar otros contenidos como una serie de preguntas geolocalizadas como la que se puede ver en la Fig. 2.2b. Los escenarios creados con esta página se podrán encontrar a través de un listado<sup>8</sup>. Una profesora puede utilizar esta herramienta para generar un escenario educativo en el que sus estudiantes tienen que contestar preguntas situadas en los puntos del patrimonio cultural de su entorno. Tanto *Eduloc* como *historypin* tendrían la misma limitación: su contenido se basa en lo que su comunidad de usuarios pueda o quiera aportar. Esta herramienta puede resultar interesante para usarla en entornos académicos puesto que el profesor puede utilizarla para generar una serie de tareas educativas para sus estudiantes. El problema es precisamente que el profesor tiene que dedicar parte de su tiempo en generar estas tareas educativas.

### 2.3. La Web Semántica y su utilización para el apoyo de aprendizaje ubicuo de patrimonio cultural

A la Web Clásica W3C<sup>9</sup> (*Word Wide Web Consortium*, organización que se encarga de desarrollar estándares para la Web) también la denomina como Web de documentos [W3CSW]. Las personas pueden navegar entre los documentos que forman la Web gracias a los hipervínculos creados para establecer relaciones entre ellos. Estos dos elementos, los documentos y su relación, están pensadas para ser interpretadas por humanos: por ejemplo, una persona que quiera comparar un televisor accederá al portal (documento) de una tienda virtual que ofrezca productos electrónicos. Una vez dentro de la página web de la tienda accederá (hipervínculo) al apartado de televisores (documento) donde podrá comparar los distintos modelos. Siguiendo con el ejemplo, si un programa tuviera que buscar ofertas de televisores le sería complicado seguir el razonamiento de la persona del ejemplo. El desarrollador de la aplicación de ejemplo debería haber analizado el código de los portales donde el programa va a buscar ofertas y configurar un algoritmo que permitiera realizar las búsquedas (si es que no se ofrece un *endpoint* en cada una de las páginas donde consultar directamente estos datos). Este algoritmo podría quedar desfasado si el código de uno de los portales fuera modificado. Por este motivo parece lógico

<sup>7</sup><https://about.historypin.org/policies/api-policy/>

<sup>8</sup><http://www.eduloc.net/es/escenari/lista>

<sup>9</sup><https://www.w3.org>

pensar que la Web Clásica no está pensada para ser utilizada por las máquinas [DuC13].

Para solventar este problema se desarrolló una evolución de la Web Clásica denominada Web Semántica. En esta nueva Web se realiza la transformación de trabajar con documentos a operar con datos. Su principal objetivo es que las máquinas realicen un trabajo más eficiente que el que podían realizar operando con la Web Clásica [W3CSW]. La definición que se propuso para la Web Semántica fue «una extensión de la web actual donde la información tiene una estructura bien definida, lo que permite una mejor cooperación entre máquinas y personas» [BHL01]. Con ella se resalta que este nuevo tipo de web está diseñada no solo para que las personas puedan utilizarlas, también está estructurada para que su información pueda ser usada por ordenadores. Otra definición de la Web Semántica es la que se ofrece en [DuC13] que define a la Web Semántica como «un conjunto de estándares y mejores prácticas para compartir datos y su semántica sobre la Web para su utilización por aplicaciones». Con el extracto «[...] para su utilización por aplicaciones» el autor quiere destacar que la Web Semántica está ideada para que pueda ser utilizadas por máquinas.

La información que se proporciona en la Web Semántica está estructurada. Para proporcionar de una estructura a esta información se tiene que seguir una serie de buenas prácticas. Al conjunto de estas buenas prácticas se las denomina Datos Enlazados [Ber06](*Linked Data*). Estas buenas prácticas se basan en cuatro reglas: identificación de cada recurso a través de un Identificador de Recurso Único (URI, *Uniform Resource Identifier*); utilizar URIs HTTP (*Hypertext Transfer Protocol*) para que las personas puedan buscar por los nombres; proporcionar información relevante a través del uso de un estándar cuando alguien busque una URI; enlazar URIs con otras URIs con el objetivo de que al realizar una consulta de un recurso se pueda encontrar información relacionada con esta.

Existen múltiples tecnologías relacionadas con la Web Semántica como RDF [CWL14] (*Resource Description Framework*) y SPARQL [HS13] (*SPARQL Protocol and RDF Query Language*). En RDF se define la forma de representar información en la Web mientras que SPARQL es un conjunto de especificaciones con las que consultar y editar contenido de la Web almacenado mediante RDF [CWL14; HS13].

RDF se convirtió en estándar en el año 1999 [DuC13]. Su estructura básica son las triplas, compuestas por tres elementos: sujeto, predicado y objeto. El objeto puede ser una IRI [DS05], (*Internationalized Resource Identifier*), que es una generalización de las URIs que extiende los caracteres que se pueden utilizar para componerlas [DuC13], un literal, que sería una cadena texto, un número o una fecha, o un *blank node* (identificadores cuya estructura no es definida por RDF). Por otro lado, los sujetos pueden ser IRIs o nodos blancos y los predicados únicamente IRIs. Al conjunto de triplas se le denomina grafo RDF. En el Listado 2.1 y en 2.2 se muestra un ejemplo con dos triplas que indican el tipo de elemento (tripla 1) y la etiqueta o nombre del elemento (tripla 2). Para este ejemplo se han utilizando los formatos N-Triples y N3. Estos formatos se utilizan para almacenar la información en ficheros. En N-Triples cada tripla termina con un punto, las cadena de texto se escribe entre comillas, y las URI entre los signos de menor y mayor que. Por otro lado, con el formato N3 se permite la utilización de espacios extras para mejorar la legibilidad del grafo, la utilización de prefijos [DuC13] y atajos (como el de utilizar el carácter ; para indicar que tanto la tripla de antes del punto y coma como la de después comparten el sujeto). Los prefijos utilizados en este documento hacen referencia a los *namespace*, conjunto de nombres de una única autoridad [AH11], de la Tabla 2.1.

Listado 2.1: Ejemplo de dos triplas escritas en formato N-Triples. Es un extracto de la información almacenada en [GSICa].

```
1 <https://casuallearn.gsic.uva.es/context/Iglesia_de_San_Pablo_(  
  Valladolid)_47245_41657> <http://www.w3.org/1999/02/22-rdf-syntax-ns#
```

```

1 type> <https://casuallearn.gsic.uva.es/ontology/physicalSpace> .
2 <https://casuallearn.gsic.uva.es/context/Iglesia_de_San_Pablo_(
  Valladolid)_47245_41657> <http://www.w3.org/2000/01/rdf-schema#label>
  "Iglesia de San Pablo (Valladolid)" .

```

Listado 2.2: Ejemplo de dos triplas escritas en formato N3. Es un extracto de la información almacenada en [GSICa].

```

1 <https://casuallearn.gsic.uva.es/context/Iglesia_de_San_Pablo_(
  Valladolid)_47245_41657>
2   a clo:physicalSpace ;
3   rdfs:label "Iglesia de San Pablo (Valladolid)" .

```

Tabla 2.1: Prefijos y namespaces utilizados en el documento

Prefijo	Namespace
clo:	https://casuallearn.gsic.uva.es/ontology/
clp:	https://casuallearn.gsic.uva.es/property/
cls:	https://casuallearn.gsic.uva.es/space/
dc:	http://purl.org/dc/elements/1.1/
edbo:	http://es.dbpedia.org/ontology/
geo:	http://www.w3.org/2003/01/geo/wgs84_pos#
owl:	http://www.w3.org/2002/07/owl#
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
skos:	http://www.w3.org/2004/02/skos/core#
xsd:	http://www.w3.org/2001/XMLSchema#

SPARQL es el lenguaje de consultas que se encuentra dentro de la especificación W3C desde el año 2008 [DuC13]. Este lenguaje permite al usuarios definir consultas de alto nivel. El usuario indicará qué información desea obtener y dónde se encuentra esa información, siendo el procesador de la petición el encargado de obtener estos datos a través de la realización de distintas etapas como la construcción, agrupación y filtrado. El procesador de la consulta intentará optimizar el proceso de cada una de las etapas [CB15]. La versión actual de este lenguaje de consultas es SPARQL 1.1, que se convirtió en *W3C Recommendation* en el año 2013.

SPARQL, como el resto de lenguajes, tiene una sintaxis definida en la Recomendación de la especificación [HS13]. En el ejemplo del Listado 2.3 se puede observar el formato de una consulta SPARQL para obtener información. En primer lugar, se indica que es una consulta para obtener información con la palabra reservada **SELECT**. SPARQL ofrece otras tres formas de realizar consultas de información mediante las palabras reservadas como **CONSTRUCT** (utilizado para obtener triplas), **ASK** (para consultar si existe o no un patron dado) y **DESCRIBE** (para obtener las triplas que describen un recurso) [DuC13]. Además, la especificación SPARQL 1.1 Update<sup>10</sup> permite realizar inserciones, eliminaciones y reemplazos en la información del grafo (suele ser necesario que el usuario se autentifique antes de permitirle realizar cambios en los datos). Las consultas que se implementarán en este TFM se realizarán usando **SELECT** por lo que la explicación del resto de formatos y posibles configuraciones se omite.

Listado 2.3: Ejemplo de consulta SPARQL.

```

1 SELECT ?etiqueta WHERE {
2   <https://casuallearn.gsic.uva.es/context/Iglesia_de_San_Pablo_(
     Valladolid)_47245_41657> rdfs:label ?etiqueta .

```

<sup>10</sup><https://www.w3.org/TR/sparql11-update/>

3 } LIMIT 5

**SELECT** se utiliza para indicar qué datos se le van a mostrar al usuario. La información que se le muestra al usuario se asigna a las variables que se indiquen después de **SELECT**. Las variables en SPARQL comienzan con el carácter de cierre de interrogación (?) o con el símbolo de dólar (\$). En el caso del Listado 2.3 solamente se ha establecido una variable (**etiqueta**). **WHERE** se usa para especificar qué información se va a extraer del conjunto de datos donde se esté realizando la consulta. En el Listado 2.3 se indica que se desea extraer la información de las triplas en las que el sujeto sea la IRI [https://casuallearn.gsic\[...\].41657](https://casuallearn.gsic[...].41657) y el predicado `rdfs:label`. Mediante la sentencia **LIMIT** el usuario puede indicar el máximo de elementos que devolverá la consulta.

### 2.3.1. Casual Learn SPARQL

Casual Learn SPARQL<sup>11</sup> es un repositorio de triplas que contiene datos abiertos enlazados sobre **tareas educativas**. Las tareas de aprendizaje almacenada en este punto están relacionadas con el **patrimonio cultural** de Castilla y León. El contenido del repositorio fue generado de manera semiautomática combinando información proveniente de distintas fuentes de datos abiertos en trabajos previos del grupo de investigación GSIC/EMIC [Rui+21c]. El objetivo de combinar estas fuentes era tener una base de conocimiento lo suficientemente rica como para generar tareas educativas en sitios relevantes del patrimonio cultural de la mencionada comunidad autónoma española [Rui+21c]. Por ejemplo, de estas fuentes de información se podía saber a qué estilo arquitectónico pertenece los edificios de una zona y, utilizando una serie de plantillas diseñadas por expertos educadores, generar tareas de aprendizaje que solicitaran comparar dos edificios de estilos diferentes o pedir realizar una reflexión del motivo por el cual en una zona existen varios edificios del mismo estilo. Al final del proceso de creación semiautomática se obtuvo una serie de ficheros `.ttl` que tienen como contenido el grafo, con las triplas expresadas utilizando el formato `N-triples`, con el contenido educativo que se va a utilizar en este TFM.

En la Fig. 2.3 se ha representado la vista general de la ontología de Casual Learn. Es una ontología formada por tres clases: **temas** (`clo:themes`), **clo:tareas** (`clo:tasks`) y **espacios físicos** (`physicalSpace`). Cada **tarea** está asociada a **una entidad física** y a **uno o varios temas**. Cada `physicalSpace` está geotiquetado en una ubicación física de especial interés por su valor cultural. Casual Learn SPARQL dispondrá de las coordenadas de estos puntos de interés (POI, *Point of Interest*). Por otro lado, el concepto de tema es más amplio que el del espacio físico. Una tarea educativa puede estar relacionado con más de un tema. Por ejemplo, los temas que aparecerán en una tarea de las asociadas a la Iglesia de San Pablo de Valladolid (POI utilizado en los Listados 2.1, 2.2 y 2.3) estarán relacionados tanto por el elemento arquitectónico que se resalte en la tarea (por ejemplo, bóveda, rosetón o imposta) como con el tipo de edificio que es (iglesia, edificio religioso, etc.).

La aplicación que se va a desarrollar tiene que utilizar la información disponible en este repositorio y presentársela de manera adecuada a los usuarios que la utilicen, ocultando el proceso de recuperación de los datos. Es por ello que el sistema deberá conocer qué información puede recuperar de cada una de las clases que se han representado en la Fig. 2.3. En primer lugar, la aplicación podrá determinar el lugar donde se encuentran las tareas de aprendizaje a partir de la información del POI. Se podrá conocer tanto la **latitud** como la **longitud** del punto geográfico donde se encuentre. Como se puede ver en la Tabla 2.2, además de la información de la ubicación del POI, la aplicación podrá obtener su **nombre**, **descripción**, **imagen**, y **autor**

<sup>11</sup><https://casuallearn.gsci.uva.es/sparql>

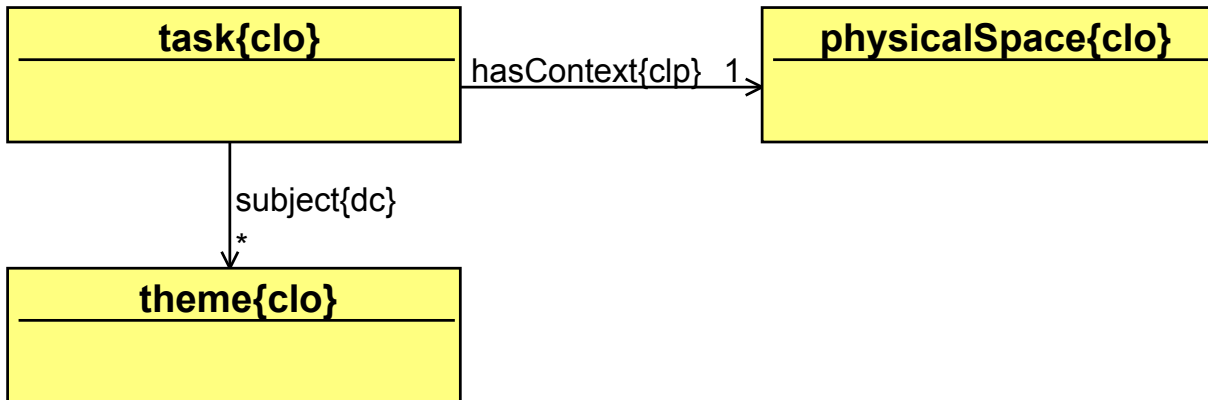


Fig. 2.3: Representación de la ontología de Casual Learn SPARQL

del POI. Las tres primeras se podrían utilizar para la descripción del lugar mientras que con la última los usuarios podrían saber quién ha generado la información.

Tabla 2.2: Propiedades presentes en los POI almacenados en Casual Learn SPARQL.

Propiedad	Descripción
<code>edbo:thumbnail</code>	Imagen en baja resolución del POI.
<code>dc:creator</code>	Autor del POI.
<code>rdf:type</code>	Clase a la que pertenece la información.
<code>rdfs:comment</code>	Descripción del POI.
<code>rdfs:label</code>	Cadena de texto con el nombre del POI.
<code>rdfs:seeAlso</code>	Identifica a las fuentes de información necesarias para la creación del POI.
<code>geo:lat</code>	Coordenada que indica la latitud donde se encuentra el POI.
<code>geo:long</code>	Coordenada que indica la longitud donde se encuentra el POI.
<code>clp:image</code>	Imagen del POI.

Una vez que la aplicación pueda identificar dónde se encuentran los puntos donde deben ir los usuarios deberá determinar qué actividades van a realizarse en estos puntos. Estas actividades serán las tareas educativas pertenecientes a la clase `tasks` cuyo listado de propiedades se pueden encontrar en la Tabla 2.3. Con el **contexto**<sup>12</sup> se enlazan las triplas de la tarea con las de un POI, por lo que no es necesario que la información que se almacena en el POI (por ejemplo las coordenadas geográficas) se almacene también en la tarea. Con la propiedad **espacio** el cliente determinará si la tarea está diseñada para ser realizada cerca del lugar donde se encuentre el POI (espacio **físico**), en cualquier lugar a través de un mapa virtual (espacio mapa web) o a través de un entorno educativo como Moodle (espacio virtual). Una propiedad que va a ser fundamental para el funcionamiento de la aplicación es el **tipo esperado de respuesta** con el que se indica qué acción se le está solicitando realizar al usuario en el **enunciado** de la tarea. Con los datos actuales de Casual Learn SPARQL estas acciones pueden ser: contestar a una o varias preguntas de manera escrita, realizar una o varias fotografías, responder a una pregunta y realizar una o varias fotografías, hacer un vídeo o contestar a una pregunta y realizar un vídeo. Por último, como en el caso del POI, el sistema que se va a desarrollar podría utilizar la **imagen** y el **nombre** para contextualizar la tarea al usuario.

Las propiedades de la última clase presente en Casual Learn SPARQL se pueden encontrar

<sup>12</sup>Debido a que la propiedad se llama **tiene contexto** en algunos lugares de este documento se puede hablar utilizando la palabra *contexto* como sinónimo de POI.

Tabla 2.3: Propiedades presentes en las tareas almacenadas en Casual Learn SPARQL.

Propiedad	Descripción
<code>edbo:thumbnail</code>	Imagen en baja resolución representativa de la tarea.
<code>dc:creator</code>	Autor de la tarea.
<code>dc:subject</code>	Propiedad que relaciona a la tarea con un tema.
<code>rdf:type</code>	Clase a la que pertenece la información.
<code>rdfs:comment</code>	Nombre del lugar donde se va a realizar la tarea.
<code>rdfs:seeAlso</code>	Identifica las fuentes de información necesarias para la creación de la tarea.
<code>clp:answerType</code>	Tipo esperado de la tarea.
<code>clp:associatedTextResource</code>	Enunciado de la tarea.
<code>clp:expectedAnswer</code>	Respuesta que se espera que se proporcione a la tarea.
<code>clp:hasContext</code>	Propiedad que relaciona a la tarea con un POI.
<code>clp:image</code>	Imagen en calidad completa.
<code>clp:jsonVersion</code>	Información en formato JSON.
<code>clp:space</code>	Espacio donde se puede realizar la tarea.

en la Tabla 2.4. Cada tarea tendrá asociado un tema o más. Las propiedades más interesantes para una aplicación como Casual Learn serían las **relaciones ascendentes y descendentes** que pudieran existir entre cada tema. De esta forma la aplicación que se va desarrollar podría, por ejemplo, saber qué tareas tienen una temática similar a la que acaba de realizar un usuario. Además, la información de los temas podría utilizarse como si fueran categorías para ofrecerle al usuario otra alternativa de ver la información contenida en el repositorio de triplas. Por ejemplo, se podría realizar una consulta como la del Listado 2.4 para informar dónde se encuentran las catedrales de Castilla y León y qué nombre tienen. Para ello, se ha tenido que recuperar previamente el IRI del tema relacionado con las catedrales.

Tabla 2.4: Propiedades presentes en los temas almacenados en Casual Learn SPARQL.

Propiedad	Descripción
<code>rdf:type</code>	Clase a la que pertenece la información ( <b>theme</b> ).
<code>rdfs:label</code>	Cadena de texto con el nombre del tema.
<code>owl:sameAs</code>	Indica que el tema descrito es igual al que se indique en el objeto.
<code>skos:broader</code>	Indica relaciones descendientes (de edificio civil se tendría molino, puente, museo, etc.)
<code>skos:narrower</code>	Indica relaciones ascendientes (de molino solo se tendría edificio civil).
<code>edbo:abstract</code>	Descripción del tema.

Listado 2.4: Petición SPARQL para obtener el nombre y posición de norte a sur de las catedrales almacenadas en Casual Learn SPARQL.

```

1 SELECT DISTINCT ?nombreCatedral ?lat ?lng WHERE {
2   ?tarea
3     dc:subject <https://casuallearn.gsic.uva.es/taxonomy/Catedral> ;
4     clp:hasContext ?poi .
5   ?poi
6     geo:lat ?lat ;
7     geo:long ?lng ;
8     rdfs:label ?nombreCatedral .
9 } ORDER BY DESC(?lat)

```



## 2.4. Conclusiones

El aprendizaje ubicuo permite a los estudiantes acceder a información relacionada con el patrimonio cultural en cualquier momento y lugar a través de un dispositivo electrónico. La aplicación instalada en un dispositivo de este tipo, como pudiera ser un móvil, debe ocultar al usuario la forma con la que se esté obteniendo información ya que, en lo que respecta a la mayoría de estudiantes, le puede resultar indiferente que la información sobre el patrimonio se encuentre almacenada en un repositorio público o privado o la forma de acceder a estos datos (basadas en tecnologías semánticas o no).

En este capítulo se han estudiado aplicaciones relacionadas con el patrimonio cultural en las que se destaca cómo representan la información relacionada con este patrimonio y qué se les pide que realicen los usuarios en esos puntos. En primer lugar, este tipo de aplicaciones suelen utilizar marcadores sobre un mapa para indicar dónde se encuentran los puntos que desean resaltar. Además, también se pueden representar los lugares destacados como una lista de tarjetas ordenadas por la distancia a la que se encuentren estas ubicaciones del lugar donde se encuentre el usuario. Ambas representaciones pueden ser válidas, pero con la que utiliza un mapa le puede proporcionar al usuario información sobre el entorno del punto a destacar. Por ejemplo, con el mapa el estudiante podrá saber a que lado de un río está el punto de interés (se tiene que desplazar hasta un puente para poder atravesarlo). Otra característica importante a la hora de representar la información es permitir al usuario la creación de listas para que pueda llevar un control sobre la que ha accedido y a las que no. En segundo lugar, se ha visto cómo hay aplicaciones puramente informativas y otras en las que los usuarios pueden interactuar con la información que existe (a través de la realización de tareas o comentando los datos de la información existente). Las que permiten interactuar con los elementos suelen ser aplicaciones colaborativas.

En las aplicaciones estudiadas se pueden encontrar una serie de limitaciones. En primer lugar, algunas de ellas eran meramente informativas. Estas aplicaciones pueden estar diseñadas pensando más en un público turístico que en uno educativo. Por este motivo carecen de tareas de aprendizaje que hagan reflexionar al estudiante de los elementos más representativos de cada lugar. En segundo lugar se encuentran las aplicaciones colaborativas. Este tipo de aplicaciones sí que se podrán utilizar para que los docentes creen escenarios educativos en los que sus estudiantes visiten una serie de lugares y/o realicen unas tareas, pero necesita que el profesor dedique parte de su tiempo en diseñar las tareas educativas y colocarlas en los lugares donde se encuentran los puntos que desea resaltar. Si una aplicación utilizara la información disponible en el repositorio Casual Learn SPARQL los docentes no necesitan crear nuevo contenido puesto que no se parte de una base de datos vacía como ocurría con las aplicaciones colaborativas. Este repositorio de triplas cuenta con más de 10 000 tareas educativas repartidas en más de 2000 puntos de interés de Castilla y León, España y casi 700 categorías relacionadas con las tareas de aprendizaje. Casual Learn deberá permitir que los usuarios utilicen esta información (objetivo de la Sección 1.1) ocultando el proceso de obtención de los datos, ya que podría suponer un problema añadido para usuarios que no sean expertos en la utilización de tecnologías relacionadas con la Web Semántica.



# Capítulo 3

## Análisis

En este capítulo se realiza la búsqueda de los requisitos para determinar qué desean los estudiantes que vayan a utilizar Casual Learn. Por este motivo, la primera fase que se realiza es describir un posible uso de la aplicación donde un usuario quiere conocer el patrimonio cultural de su ciudad. A partir de ese ejemplo se describen las historias de usuario para determinar qué funcionalidades tiene que cumplir el sistema. Además de las historias de usuario, en este capítulo se indican los casos de uso de la aplicación y los requisitos no funcionales. En la última parte del capítulo se realiza una discusión de cómo podría ser la arquitectura de la aplicación que se va a desarrollar y los motivos por los que se ha elegido el modelo que finalmente se ha implementado.

### 3.1. Introducción

Siguiendo el modelo de la metodología ágil presentado en la Sección 1.2, modelo que permitía ofrecer un producto finalizado de forma rápida al cliente [SS05], en este capítulo se va a realizar el estudio de la fase de análisis. En esta fase se deberán detectar los requisitos que solicita el cliente al producto. En el caso del sistema planteado no se dispondrá de un cliente típico puesto que los requisitos vendrán determinados por un grupo de expertos formado por miembros del grupo de investigación GSIC/EMIC [GSICb]. Las opiniones de estos expertos estarán basadas en su propia experiencia como usuarios de sistemas similares y en las distintas entrevistas que han realizado con profesores de varios institutos. Siempre se tendrá en cuenta que los usuarios finales del sistema son estudiantes de los últimos cursos de Enseñanza Secundaria Obligatoria y bachillerato. Además, se deberá deliberar sobre la arquitectura general de la aplicación, que estará indirectamente determinada por los requisitos que los expertos hayan solicitado y con los que la aplicación debe cumplir. El funcionamiento de este sistema se basará en los datos disponibles en el repositorio de triplas mantenido por el grupo GSIC/EMIC. En él se encuentran disponibles un conjunto de puntos de interés (POI) ubicados en la comunidad autónoma de Castilla y León relacionados con una serie de tareas de aprendizaje sobre el patrimonio cultural de esta región española.

Con el objetivo de obtener los requisitos que el sistema debe cumplir, se van a exponer una serie de posibles casos de utilización de la aplicación. Se formularán desde el punto de vista de utilizar la aplicación como una herramienta que posibilite la conexión entre lo aprendido en clase y el patrimonio cultural del contexto local del aprendiz. Además, se tendrá en cuenta el caso de un usuario que utilice la aplicación de una forma más casual, considerando únicamente a la aplicación como una fuente fiable de información en vez de verla como un apoyo al aprendizaje

ubicuo. Una vez expuestos estos ejemplos, se enumerará una serie de historias de usuario [Koe16] que la aplicación deberá satisfacer. Las historias de usuario exponen de manera clara y concisa alguna característica que solicita un usuario del sistema. Esta forma de escribir las acciones que debe cumplir la aplicación debería ser suficiente para extraer los requisitos del sistema. De esta forma se evitaría realizar la clásica redacción de requisitos en la que, en algunos casos, su coste de creación supera a sus beneficios [Koe16]. La redacción de las historias de usuario les facilitará la comprobación de si el sistema desarrollado satisface las necesidades que han planteado.

Una vez que mediante las historias de usuario se han identificado las características que la aplicación debería tener se pasa a estudiar cuáles podrían ser las distintas opciones para implementarlo. De todas las alternativas planteadas se expondrá cuál se ha elegido y se proporcionará una primera aproximación de la arquitectura de la aplicación. Con cada una de las partes del sistema bien definido ya se podrán plantear sus casos de uso. Un caso de uso es una explicación de cómo los usuarios pueden llevar a cabo una acción en el sistema [Koe16]. Los casos de uso expuestos en este capítulo serán los más representativos de la aplicación. Se obvian los casos de uso más simples como el que se podría plantear para llevar a cabo una modificación en la configuración de la aplicación o el de mostrar al usuario una lista de tareas completadas o lugares notificados.

## 3.2. Ejemplo de uso de la aplicación

La información aglutinada en el repositorio de Casual Learn, compuesta mayoritariamente por tareas educativas y POI creados de manera semiautomática [Rui+21c], está pensada para ser utilizar como apoyo a la educación. Debido a ello, un estudiante de Historia del Arte podría querer utilizar esta información para reforzar alguno de los conceptos que su profesor haya explicado.

En las últimas clases a las que ha asistido este estudiante se le han explicado las principales características del estilo arquitectónico gótico. En el desarrollo de estas clases, el docente que le imparte la asignatura ha indicado una serie de ejemplos representativos de este estilo como la catedral de Notre-Dame de Francia o la de Canterbury de Reino Unido. Este estudiante es de Valladolid (España), por lo que se decide a buscar construcciones de este estilo arquitectónico en su entorno. Para ello acude a los datos almacenados en Casual Learn SPARQL ya que su profesor les habló de este sistema, pero se encuentra con el problema de que no conoce, ni quiere conocer, cómo es la sintaxis necesaria para obtener la información que necesita conocer. Además, la interfaz gráfica del punto SPARQL no le resulta cómoda, por lo que decide utilizar Casual Learn para utilizar la información almacenada en el repositorio.

La aplicación la puede utilizar en su dispositivo móvil y le muestra los lugares donde se encuentran los POI de su entorno. Además de estos puntos el estudiante se localiza en el mapa gracias a que la interfaz también muestra su ubicación. Con ello, después de encontrar varios edificios góticos en Valladolid se decide visitar uno de ellos puesto que es el que más cerca se encuentra de su posición. Con todo ello, el usuario inicia su paseo sorprendiéndole gratamente que la aplicación le vaya notificando de lugares de interés cercanos a su posición de manera periódica. En el trayecto que necesitaba recorrer para llegar al edificio gótico le aparece una notificación de un lugar que no conocía, pero decide visitarlo más tarde.

Una vez que el estudiante se encuentra próximo a la ubicación del lugar que desea visitar observa que en la aplicación se han desbloqueado una serie de tareas asociadas a este edificio. El estudiante observa que en estas tareas se le pide realizar distintas acciones como realizar

fotografías a algún elemento del lugar de interés o escribir un pequeño texto comparando el estilo de este edificio con otro cercano. El estudiante decide realizar una tarea donde se le solicita realizar una fotografía y contestar a una pregunta. Al finalizar la tarea al estudiante se le muestran una serie de redes sociales y entornos educativos para compartir la respuesta. El estudiante decide compartir esta respuesta en un grupo de Microsoft Teams que su profesor había creado para cuando uno de ellos tuviera algo que contar (relacionado con la asignatura). Los compañeros de este estudiante le preguntan de por qué ha realizado esa tarea y el estudiante comparte el enlace de la aplicación para que sean ellos los que aprendan a utilizarla.

### 3.3. Historias de usuario

Las historias de usuario son una forma de describir las características que un sistema tiene que cumplir siendo fáciles de entender por todas las partes que participen en la creación del sistema (desde los desarrolladores hasta los clientes) [Koe16]. Debido a esta facilidad de escritura y lectura las historias de usuario son más apropiadas para la metodología ágil empleada en este TFM [Koe16]. Se escriben siguiendo una estructura bien definida como se puede ver en el Listado 3.1. En esta estructura se puede observar que existen tres lugares que se deberán personalizar para cada historia de usuario: el **rol** indicará la función que desempeña un usuario; la **característica o función** es lo que el usuario representado mediante su rol necesita; el **beneficio** es lo que el usuario consigue al implementar la característica en la aplicación. Además, las historias de usuario deben cumplir que sean independientes unas de otras, negociables, que proporcionen valor al usuario, que se pueda hacer una estimación del coste que tendrá su desarrollo, acotadas de funciones para que puedan implementarse en ciclos de horas o días (para que encajen con la metodología ágil) y que se pueda comprobar su correcto funcionamiento [Koe16].

Listado 3.1: Estructura que debe seguir una historia de usuario.

Como un {rol}, quiero {característica o función} para {beneficio}.

En la Tabla 3.1 se muestran las historias de usuario para Casual Learn. Estas historias de usuario representan funcionalidades que se necesitan implementar para que los clientes tengan un sistema completo. Independientemente de que todas son fundamentales, se ha podido darle un grado de prioridad u otro. Esto es debido a que por ejemplo si el sistema no permite resolver tareas de aprendizaje (US03) difícilmente el usuario va a poder compartir las respuestas en redes sociales (US05).

Tabla 3.1: Narrativa de las historias de usuario de Casual Learn junto con un identificador (ID) y prioridad que tiene para el usuario. Se omite la parte inicial de la estructura que debe seguir una historia de usuario puesto que todas empiezan por «*Como un estudiante, quiero...*».

ID	Narrativa de la historia de usuario	Prioridad
US00	visualizar sobre un mapa los POI de una zona para poder ubicarlos	Alta
US01	conocer mi posición sobre el mapa para saber qué POI existen cerca de donde me encuentro	Alta
US02	que se me notifique de POI cercanos mientras voy paseando para no tener que llevar la pantalla del móvil encendida	Alta
US03	que la aplicación me permita resolver los tipos tareas educativas incluidas en Casual Learn SPARQL para desarrollar distintas habilidades	Alta
US04	compartir las respuestas con mi profesor para que pueda evaluarme	Media

Tabla 3.1 continuación de la página anterior

ID	Narrativa de la historia de usuario	Prioridad
US05	compartir las respuestas en mis redes sociales para que mis contactos sepan dónde he estado	Baja
US06	compartir las respuestas en un sistema propio de la aplicación para no tener que compartir mis RRSS con mi profesor	Media
US07	encontrar los POI de un municipio para poder planificar mis viajes	Baja
US08	que el sistema tenga en cuenta mis preferencias a la hora de proponerme tareas para que me muestre primero las tareas que más me gusta realizarlas	Alta
US09	poder diferenciar las tareas propuestas por mi docente para poder realizarlas	Media
US10	ver las tareas/POI que me ha propuesto mi profesor para realizarlas	Media
US11	poder desactivar la notificación de POI para que la aplicación no me moleste	Media
US12	poder controlar la visualización de mis respuestas en el portafolio para que se respete mi privacidad	Alta
US13	que se me permita utilizar la aplicación sin identificarme para conocer su funcionamiento	Baja
US14	que la información de inicio de sesión sea la menor posible para no tener que escribir el usuario y la contraseña	Baja

Las historias de usuario US00, US02, US03, US09, US10, US14 tienen que ver con la información que se extrae de Casual Learn SPARQL. Por ejemplo, para que poder realizar US00 será necesario obtener de alguna forma cuáles son las posiciones de los POI que contiene el repositorio de triplas y para US02 se necesitará conocer qué POI se encuentran alrededor del usuario. Por este motivo, y aunque ya se indicaba en el objetivo principal del TFM, la aplicación deberá obtener de Casual Learn SPARQL tanto la información de las tareas educativas como la de los POI.

US01 es necesario para que el usuario sepa donde se encuentra con respecto a las ubicaciones de los POI. Es necesario que conozca tanto la ubicación del POI como su posición para saber si va a tener que superar alguna barrera arquitectónica para llegar hasta el POI. Por ejemplo, en la ciudad de Valladolid las vías del tren delimitan unas zonas de otras y solamente se pueden atravesar por túneles y puentes que se encuentran separados a cierta distancia. Por ello, aunque la distancia entre un usuario y un POI en línea recta es baja si tiene que superar este tipo de barreras el usuario, posiblemente, vaya a visitar otro POI.

Las historias de usuario US03, US04, US05, US06 y US12 están relacionadas con la realización de las tareas educativas y su publicación en las diferentes plataformas que estarán soportadas por Casual Learn. En primer lugar, como en el repositorio existen varios tipos esperados de respuesta US03 solicita que la aplicación sea capaz de adaptar su interfaz para cada uno de estos tipos. Por otro lado, US04, US06 y US12 son historias de usuario que establecen las primeras características del portafolio. El usuario podrá controlar qué información es publica y podrá compartir el identificador del portafolio para que su profesor pueda evaluarle. Además US05 establece que se pueda compartir información en plataformas de terceros.

US02, US08 y US11 tienen que ver con la notificación o no de puntos de interés cercanos al usuario. También con que se deberá pedir opinión a los usuarios para conocer sus preferencias sobre qué tipo de tarea educativa prefieren realizar y recomendarle realizar unas u otras en función de lo que haya respondido.

US00, US01 y US07 estarán relacionados con la búsqueda por parte del usuario de nuevos POI. Con US07 se utilizará para llegar rápidamente a municipios lejanos al usuario. Esta búsqueda sería tediosa de realizar si solo se pudiera realizar utilizando un mapa virtual (es más rápido buscar por el nombre del municipio que tenerlo que localizar en un mapa).

Por último US13 y US14 están relacionadas con la identificación del usuario que utiliza la aplicación. Los usuarios podrán probar la aplicación para ver si es de su grado antes de identificarse. Además, esta identificación se deberá realizar sin tener que crear una nueva identificación de usuario para este servicio.

### 3.4. Casos de uso del sistema

Mientras que las historias de usuario describen funcionalidades finales del sistema, los casos de uso muestran paso a paso cómo se produce todo el proceso: desde que un actor lo inicia, las interacciones entre el sistema y los actores y la acción que completa el proceso. Por ello, los casos de uso descritos en esta sección estarán basados en las historias de usuario de la Sección 3.3.

Para describir los casos de uso se tomará Casual Learn como si fuera un único elemento. Por este motivo no se entrará en detalle del intercambio de información interno que se pueda producir entre las partes. De este modo se dejarán estas decisiones para las etapas de diseño e implementación de los capítulos 4 y 5. La descripción de estos casos tiene una estructura concreta compuesta por una serie de campos [Koe16] como son: un **identificador único**; un **título** lo más corto posible; una **descripción** que explique de forma breve su propósito; los **actores** que intervienen en él; las **precondiciones** que se tienen que cumplir para que se produzca; las **postcondiciones** que suceden cuando finaliza; el **disparador** que provoca su inicio; el **camino básico** que se sigue cuando no sucede nada incorrecto; los **caminos alternativos** que se provocan cuando se produce un evento que impide que se siga por el camino básico. Además se agrega un campo para **información extra** por si fuera necesario aclarar alguna información indicada en el caso de uso. En la Fig. 3.1 se puede ver el diagrama de casos de uso de la aplicación que se va a crear. Con este diagrama se puede entender qué actor interviene en cada uno de ellos.

CU00 Ver puntos de interés

<b>Descripción</b>	Un estudiante navega por la pantalla de mapas de la aplicación.
<b>Actor</b>	Estudiante y repositorio de triplas.
<b>Precondición</b>	El estudiante ha dado permiso para compartir su ubicación y está interactuado con el mapa.
<b>Postcondición</b>	En la pantalla del estudiante se representan marcadores con los puntos de interés existentes en la zona representada.
<b>Disparador</b>	La diagonal del mapa mostrado en la pantalla del usuario es inferior a 4km, el usuario realiza un desplazamiento en el mapa y han transcurrido más de 250ms desde que se produjo el último inicio de este caso de uso.
<b>Camino básico</b>	1. El sistema divide el mapa en cuadrículas fijas desde el punto prefijado. Comprueba la información que disponga en la memoria local. La que no disponga se la solicita al repositorio de triplas.

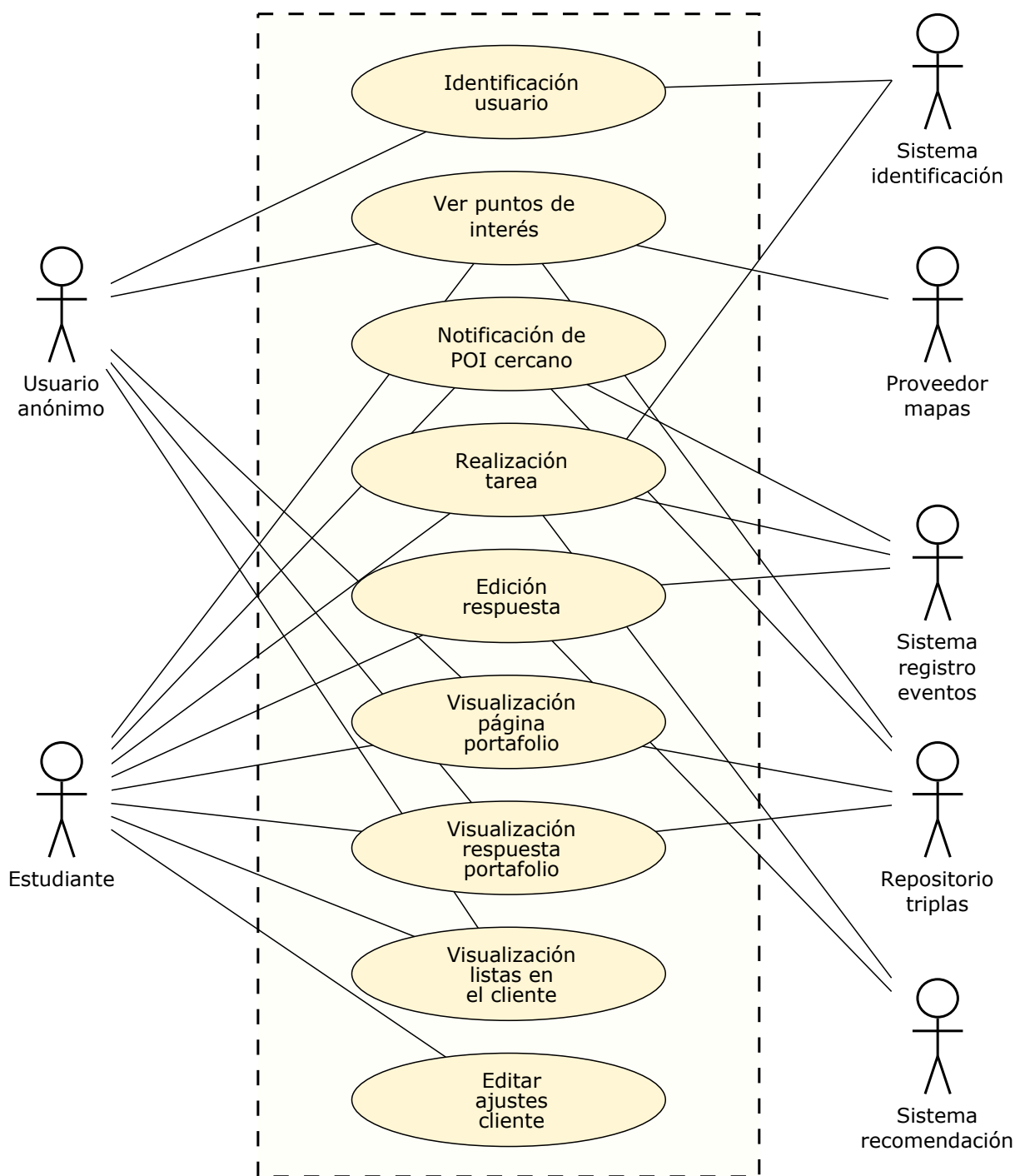


Fig. 3.1: Casos de uso de la aplicación que se va a desarrollar. Con la silueta de una persona se representan a los actores de la aplicación. La línea discontinua separa a estos actores del sistema que se va a crear.

2. El repositorio de triplas recibe una consulta solicitando los POI en una zona. De todo el conjunto de datos solo devuelve la información de los puntos que hayan pasado el filtrado.
3. Cuando el sistema recibe toda la información del servidor, agrupa los POI dependiendo del nivel de zum y de la distancia que exista entre ellos y representa la información por la pantalla del dispositivo del usuario.



- Cam. alternativo** 1. Si la aplicación distribuida no es capaz de obtener respuesta del repositorio de triplas representa solamente la información que tenga almacenada en memoria local.
- Info. extra**
- Para pasar de grados a kilómetros en toda la aplicación se utiliza la fórmula del semiverseno [Rob57] en la que se efectúa una simplificación en la forma de la Tierra.
  - Con la limitación de los 4km se pretende que el número de objetos que tenga que representar el dispositivo del usuario sea limitado.
  - El cliente de la aplicación distribuida proporciona una validez de un día a los datos que obtenga del repositorio de triplas.

## CU01 Notificación de POI cercano

- Descripción** Se comprueba en intervalos discretos de dos minutos la posición del estudiante para proporcionarle una notificación de que se encuentra cerca de un punto de interés.
- Actor** Estudiante, repositorio de triplas y el sistema de registro de eventos.
- Precondición** El estudiante se ha identificado y ha concedido el permisos sobre posicionamiento en segundo plano.
- Postcondición** El estudiante observa una notificación en el centro de notificaciones de su sistema.
- Disparador** Han transcurrido más de `timn-2` minutos y más de 2 minutos desde la última iteración.
- Camino básico**
1. El sistema obtiene la posición del estudiante. Con ella comprueba si la información disponible sobre los puntos cercanos sigue siendo válida. Si es válida estima la distancia que el estudiante ha recorrido para saber si está caminando. El sistema comprueba si existe algún POI a menos de 150m de la posición del estudiante. Si existe, el usuario no ha realizado ninguna tarea educativa en él y no se ha notificado en la última semana se crea una notificación.
  2. El sistema de registro de eventos registra que un usuario ha recibido una notificación de POI cercano.
  3. El usuario recibe la notificación del sistema.
- Cam. alternativo**
1. Si no se dispone de información válida de los puntos de interés o el usuario se ha salida de la zona que se tenía descargada se solicita al repositorio de triplas la información de la nueva zona con una diagonal aproximada de 1000m. El sistema solicita la información de esa zona al repositorio de triplas.
  2. El repositorio de triplas recupera la información de los POI según el criterio que haya determinado el sistema y se la devuelve en la respuesta.
  3. El sistema recibe los datos de Casual Learn SPARQL y vuelve a iniciar las comprobaciones del camino básico.
- Info. extra**
- `timn` pueden ser configurado por el usuario.
  - La información se considera válida cuando ha transcurrido menos de un día desde que se obtuvo del punto SPARQL.

## CU02 Realización una tarea educativa

Descripción	El cliente de la aplicación le permite al estudiante realizar una tarea de aprendizaje cuya respuesta será almacenada en el la infraestructura del servidor de la aplicación.
Actor	Estudiante, sistema de recomendación, sistema de identificación y el sistema de registro de eventos.
Precondición	El estudiante ha completado una tarea educativa y ha proporcionado una valoración.
Postcondición	La respuesta del estudiante se encuentra almacenada tanto en el almacenamiento local donde se ejecuta el cliente de la aplicación como en el servidor. El servicio de recomendación tiene la valoración de la tarea.
Disparador	El estudiante completa una tarea y la valora.
Camino básico	<ol style="list-style-type: none"><li>1. El sistema comprueba que el estudiante esté identificado a través del sistema de identificación.</li><li>2. El sistema de identificación comprueba que el estudiante se encuentre identificado. Envía el estado del estudiante al sistema.</li><li>3. Si el usuario está identificado el sistema almacena la respuesta del estudiante. El sistema envía al servicio de recomendación la valoración del estudiante a la tarea educativa.</li><li>4. El sistema de recomendación almacena la valoración.</li><li>5. El sistema de registro de eventos registra que un usuario ha completado una tarea.</li><li>6. El sistema muestra al usuario que la respuesta se ha almacenado y que puede editarla si lo desea.</li></ol>
Cam. alternativo	<ol style="list-style-type: none"><li>1. Si el sistema no tiene conexión a Internet espera a realizar los envíos hasta que vuelva a estar conectado. Cuando vuelva a tener acceso seguirá los pasos del camino básico.</li><li>3. Si el usuario no está identificado mediante el servicio de identificación la respuesta se descarta y no se almacena.</li></ol>
Info. extra	—

**CU03 Edición respuesta**

Descripción	El estudiante ha completado una tarea y desea editar su respuesta o su valoración.
Actor	Estudiante, sistema de recomendación y el sistema de registro de eventos.
Precondición	El estudiante ha editado la respuesta en local.
Postcondición	La respuesta modificada por el usuario se encuentra almacenada en el sistema. El servicio de recomendación tiene la nueva valoración.
Disparador	El usuario pulsa sobre la opción de editar una respuesta desde su pantalla de visualización.
Camino básico	<ol style="list-style-type: none"><li>1. El sistema almacena la nueva respuesta del estudiante. El sistema envía al servicio de recomendación la valoración del estudiante a la tarea educativa.</li><li>2. El sistema de recomendación almacena la valoración.</li></ol>

	3. El sistema de registro de eventos registra que un usuario ha modificado una respuesta.
	4. El sistema muestra al usuario que la respuesta ha sido modificada.
Cam. alternativo	1. Si el sistema no tiene conexión a Internet espera a realizar los envíos hasta que vuelva a estar conectado. Cuando vuelva a tener acceso seguirá los pasos del camino básico.
Info. extra	–

## CU04 Visualización página portafolio

Descripción	Un estudiante desea consultar una página de un portafolio en un navegador web.
Actor	Estudiante y repositorio de triplas.
Precondición	El estudiante propietario del portafolio ha compartido con alguien el identificador único de su portafolio y lo mantiene como público.
Postcondición	Se le envía al usuario la información solicitada sobre la página del portafolio.
Disparador	Un usuario realiza una petición de acceso a un portafolio.
Camino básico	<ol style="list-style-type: none"> <li>1. El sistema recupera la información de las respuestas del estudiante que sea el propietario del portafolio. El sistema se comunica con el repositorio de triplas para solicitar la información que le falte de las tareas (como el enunciado).</li> <li>2. El repositorio de triplas obtiene la información que le haya solicitado el sistema.</li> <li>3. El sistema combina toda la información de las tareas educativas con las respuestas del estudiante y se las envía al usuario que las haya solicitado.</li> </ol>
Cam. alternativo	–
Info. extra	La información que se genera en este caso de uso está preparada para que se pueda adaptar a la resolución del dispositivo donde se vaya a visualizar.

## CU05 Visualización respuesta portafolio

Descripción	Se muestra la respuesta que ha proporcionado el estudiante a una tarea junto con su descripción.
Actor	Usuario y repositorio de triplas.
Precondición	El estudiante propietario del portafolio ha compartido con alguien el identificador único de su portafolio.
Postcondición	Se le muestra la información oportuna al usuario que desea visualizar la respuesta del portafolio.
Disparador	El usuario accede a la página realizando la operación HTTP GET a la URL de la respuesta del portafolio.
Camino básico	<ol style="list-style-type: none"> <li>1. El sistema recupera la información de la respuesta del estudiante. El sistema se comunica con el repositorio de triplas para solicitar la información que le falte de la tarea.</li> </ol>

2. El repositorio de triplas obtiene la información que le haya solicitado el sistema.
3. El sistema combina toda la información de la tarea educativa con la respuesta del estudiante y se la envía al usuario que las haya solicitado.

Cam. alternativo –

Info. extra La información que se genera en este caso de uso está preparada para que se pueda adaptar a la resolución del dispositivo donde se vaya a visualizar.

#### CU06 Identificación del usuario

Descripción El usuario desea realizar alguna de las acciones que solo están permitidas a estudiantes.

Actor Usuario sin identificar y sistema de identificación.

Precondición El usuario dispone de una cuenta válida para ser utilizada en el sistema.

Postcondición La aplicación obtiene un identificador único asociado al estudiante.

Disparador El usuario pulsa sobre la opción de «iniciar sesión» en alguna de las pantallas donde se muestre.

Camino básico

1. El sistema se comunica con el servicio de identificación para recuperar el identificador de usuario.
2. El servicio de identificación envía el identificador único de usuario para este sistema.
3. El sistema recibe un identificador único de usuario junto con información básica de este (nombre y dirección de correo).
4. Se crea un nuevo evento de nuevo usuario que es enviado al servicio de registro de eventos.

Cam. alternativo –

Info. extra El usuario identificado podrá configurar los ajustes de la aplicación, realizar tareas de aprendizaje y recibir notificaciones de puntos de interés cercanos.

### 3.5. Requisitos no funcionales

Los requisitos no funcionales son aquellos que definen cómo debería ser el sistema sin agregar nuevas funcionalidades [Koe16]. Los requisitos no funcionales estarán relacionado con la arquitectura, la eficiencia, la seguridad, el rendimiento, etc. En la Tabla 3.2 se recogen algunos de los requisitos no funcionales que solicitaron los clientes.

Tabla 3.2: Requisitos no funcionales de Casual Learn.

ID	Requisito no funcional
RNF00	Personalización de la frecuencia con la que van a notificar los POI cercanos.
RNF01	Retrasar la publicación de las tareas en el portafolio de la aplicación un tiempo pseudoaleatorio para respetar la privacidad el los usuarios y que no se pueda utilizar la aplicación para rastrearlos.

Tabla 3.2 continuación de la página anterior

ID	Requisito no funcional
RNF02	Reducir al mínimo posible la cantidad de datos utilizados de la tarifa de Internet.
RNF03	El cliente tiene que ser compatible con la mayor cantidad posible de dispositivos.
RNF04	Tamaño de descarga de la aplicación reducido para que los usuarios no descarten su uso antes de instalarla.
RNF05	Cliente descargable desde una tienda de aplicaciones que proporcione confianza.
RNF06	Personalización de la red que se desea utilizar para almacenar las respuestas en los servicios propios de Casual Learn.
RNF07	Los colores de la interfaz gráfica tiene que combinar.
RNF08	La interfaz gráfica tiene que intentar seguir alguna guía de diseño.
RNF09	Ver tanto las tareas propuestas por mi profesor como el resto para realizar alguna si la veo interesante
RNF10	Conocer el número de tareas educativas que no he contestado en un POI para saber si tengo que ir hasta ese POI

### 3.6. Discusión de la visión general del sistema

El sistema tiene como objetivo apoyar el aprendizaje ubicuo del patrimonio cultural. Para que se cumpla con este objetivo, y las características que se extraigan de las historias de usuario la Sección 3.3, se les debe proporcionar a los usuarios de una interfaz sencilla a través de la cual puedan acceder a los datos almacenados en el repositorio de triplas de manera transparente. Desde el punto de vista de la arquitectura lógica, este sistema –al igual que muchos otros– se puede estructurar siguiendo el modelo de tres capas, donde se diferencian las capas de presentación, negocio y datos. Los usuarios acceden y utilizan el sistema a través de la capa de presentación. Este acceso se suele conseguir a través de una interfaz gráfica como una aplicación, un navegador web o, directamente, con un intérprete de línea de comandos. Las operaciones que el sistema tenga que ejecutar se realizan dentro de la capa de negocio. Los resultados de estas operaciones se almacenan en la capa de datos. La división de los sistemas en tres capas bien diferenciadas tiene el beneficio de poder desarrollar sistemas modulares. Por ejemplo, se podría crear un sistema que tuviera diferentes interfaces para diferentes tipos de usuarios y que las capas de negocio y de datos fueran la mismas para todos. Otra ventaja del modelo lógico de tres capas es que se controla el acceso a los datos, impidiendo que se realice directamente desde la capa de presentación. Esto puede limitar problemas de seguridad en el acceso a los datos. Otra ventaja de esta arquitectura es la posibilidad de escalar el sistema [Re03].

La aplicación podría implementarse de diferentes formas. En la primera alternativa planteada las tres capas del modelo lógico se ejecutan en un único dispositivo físico (basándose en *thick client* [Re03]) que también tendría que comunicarse con el punto SPARQL para obtener la información de las tareas educativas y los POI. La ventaja de esta implementación es que gran parte del sistema se ejecutaría directamente en los terminales de los estudiantes con lo que se evitarían los tiempos de latencia que se podrían producir al distribuir el sistema en varias partes. Además, se reducirían los recursos propios utilizados para ejecutar el sistema. Que gran parte del sistema se ejecute directamente en los dispositivos de los usuarios también se podría ver como desventaja. Esto es debido a que estos dispositivos funcionan utilizando la energía que les proporciona una batería. Si la aplicación consume muchos recursos en realizar todas las operaciones del sistema es probable que los usuarios observen un gasto energético excesivo por parte de la aplicación y terminen desinstalándola. Además, si no se dispone de un lugar externo al dispositivo del usuario donde se almacenen sus respuestas resultaría más complicada la

creación de un portafolio que sea accesible de manera unívoca. Esta alternativa también tendría el inconveniente de que todas las operaciones se realizarían en el dispositivo del usuario por lo que sería necesario que dispusiera de una cantidad de recursos superior a las otras alternativas que se van a plantear.

Como segunda alternativa se propone un sistema cliente – servidor en el que la aplicación instalada en los dispositivos sea únicamente interfaz, es decir, necesite comunicarse constantemente con el servidor para poder mostrarle al usuario información. Esta opción tiene como ventaja la total separación de los componentes del sistema siguiendo la arquitectura del modelo de tres capas. Las desventajas de esta alternativa es la dependencia del sistema de conectividad a Internet, la variabilidad de los tiempos de respuesta y los posibles problemas de escalabilidad del servidor. Los posibles problemas en el servidor son debidos a que se necesitaría disponer de una cantidad elevada de recursos para poder atender a un número indeterminado de usuarios del sistema que estuvieran constantemente realizándole peticiones.

La tercera alternativa que se va a plantear está basada en el modelo de tres capas, pero permitiendo que el cliente pueda acceder a la capa de negocio y mantener una pequeña memoria caché. Fragmentando el sistema de esta forma se consigue utilizar de una forma eficiente los recursos de cada parte del sistema. En primer lugar, en los dispositivos móviles se representarían datos que fueran relevantes para los usuarios. La capacidad de decidir qué datos son relevantes para el usuario en cada instante se realizaría con la capa de la lógica de negocio que se estuviera ejecutando en el dispositivo del estudiante. Además, datos como las configuraciones de los usuarios se podrían almacenar en local, así como los datos que fueran generando los usuarios al utilizar el sistema y la información proveniente del punto SPARQL que se pudieran almacenar en local debida a la baja variabilidad. En segundo lugar, el servidor del sistema tendría una parte de capa de negocio y otra de datos. En la capa de negocio se tomarían las decisiones para ofrecer a los usuarios los datos que solicitaran del repositorio de triplas (contextos o tareas). Además, se almacenarían las respuestas de los usuarios (capa de datos) para que pudieran ser públicas a través de los portafolios. El portafolio de cada usuario podrá ser consultado a través de un navegador web (presentación), por lo que el servidor deberá generar los documentos que fueran necesarios. Como desventaja de utilizar esta alternativa es que se tendrían las capas del modelo distribuidas entre el cliente y el servidor de la aplicación.

Como se ha podido comprobar todas las alternativas tienen sus ventajas y desventajas. Para tomar la decisión de qué alternativa escoger para desarrollar la aplicación basta con observar cómo funcionan las aplicaciones móviles más populares. La gran mayoría necesitan de conectividad a Internet para poder desempeñar todas sus funciones, pero son capaces de funcionar, al menos de manera limitada, sin conectividad. Esto quiere decir que utilizan un modelo cliente – servidor en el que parte de la lógica del sistema se ejecuta directamente en el dispositivo móvil. Por este motivo se desarrollará la tercera alternativa presentada.

Como se acaba de indicar, la aplicación distribuida que se va a desarrollar estará formada por dos partes bien diferenciadas: una parte cliente que se ejecutará en los terminales móviles de los usuarios y una servidora que operará utilizando recursos del Grupo de Investigación GSIC/EMIC. Estos elementos se comunicarán con otros externos para conseguir que se puedan cumplir las características deseadas por los usuarios expuestas en la Sección 3.3. En la Fig. 3.2 se puede ver la arquitectura física básica de la aplicación que se va a desarrollar. En la figura previamente citada se observa cómo el servidor tendrá que comunicarse con el cliente del sistema y con el repositorio de triplas. Las comunicaciones entre el servidor y el cliente se llevará a cabo gracias al uso de una API REST que va a ser implementada en el servidor. Las comunicaciones que utilizan tecnologías de la Web Semántica se producirán entre el servidor y Casual Learn SPARQL.

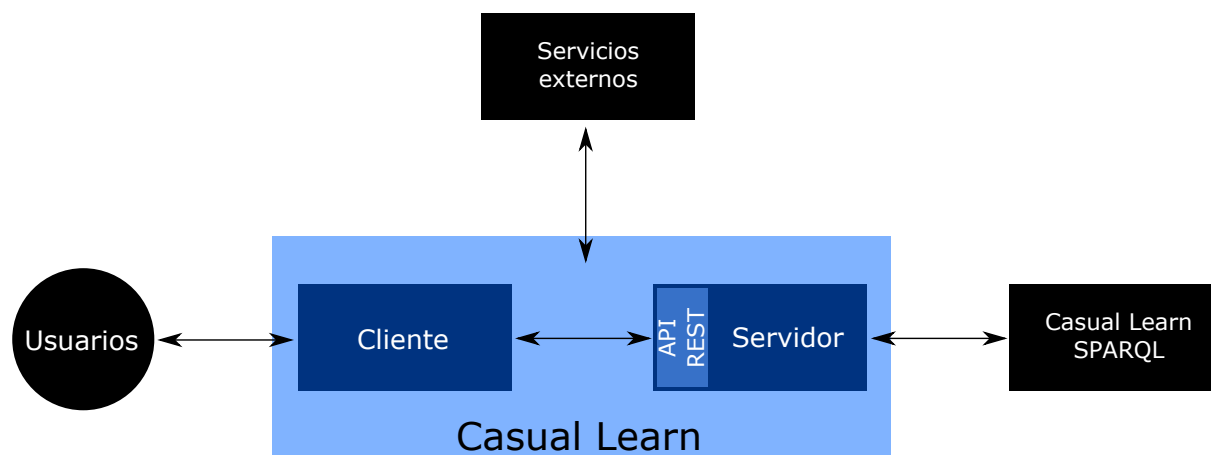


Fig. 3.2: Arquitectura física básica de la aplicación distribuida que se va a desarrollar (color de fondo azul). El cliente utilizará la API REST del servidor para obtener información y almacenar las respuestas de los estudiantes. El servidor realizará solicitudes SPARQL a Casual Learn SPARQL.

En la Fig. 3.2 se puede ver como Casual Learn necesitará comunicarse con distintos servicios para conseguir llevar a cabo todas sus funciones. Por ejemplo, para poder representar las ubicaciones de los POI sobre un mapa se necesitará un servidor de mapas que lo proporcione. Existen múltiples proveedores de mapas. Algunos de estos proveedores son grandes empresas tecnológicas como Google con **Google Maps Platform**<sup>1</sup> o Microsoft con **Bing Maps**<sup>2</sup>. Estas empresas disponen de planes gratuitos para proyectos que no vayan a utilizar una gran cantidad de recursos, pero los datos que utilizan son de su propiedad y son ellas las que se encargan de mantenerlos. Por otro lado, existen proveedores como **OpenStreetMap**<sup>3</sup> que proporcionan mapas cuyo contenido es creado por su propia comunidad de usuarios y dispone de una licencia que permite utilizar sus datos en otros proyectos de manera libre (proporcionando el adecuado reconocimiento).

Para que los usuarios puedan compartir las respuestas que realicen de las diferentes tareas el sistema ofrecerá diferentes alternativas. En primer lugar, como las respuestas se van a almacenar en el servidor de Casual Learn, se creará un portafolio personal con las respuestas que vaya proporcionando. El portafolio se podrá consultar a través de cualquier navegador web y lo podrá visualizar cualquier persona que disponga del identificador del portafolio (siempre y cuando el usuario no lo oculte). Por otro lado, si el estudiante prefiere compartir las respuestas en otros servicios independientes de la aplicación que se va a crear podrá hacerlo. Se le proporcionarán las herramientas para que pueda compartir sus avances en redes sociales populares (pensando en el usuario más casual) y en plataformas destinadas a apoyar el aprendizaje (para un ambiente educativo).

El cliente también deberá proporcionar un sistema para que los usuarios que lo deseen puedan identificarse. El mantenimiento de cuentas de usuario conlleva tener que gestionar información sensible como pueden ser las contraseñas o los datos personales de los usuarios. Para evitar tener que trabajar con este tipo de datos y tener que gestionar los problemas logísticos que conllevan se puede recurrir a un servicio de terceros especializado en este tipo de operaciones. Del mismo modo, para el registro de los eventos de acciones de los usuarios se podría haber optado por registrar cada evento en el servidor de la aplicación (lo que significaría aumentar la cantidad de recursos propios destinados a este fin) o utilizar un servicio de terceros que esté optimizado para

<sup>1</sup><https://cloud.google.com/maps-platform>

<sup>2</sup><https://www.microsoft.com/en-us/maps/>

<sup>3</sup><https://www.openstreetmap.org>

este propósito.

## 3.7. Conclusiones

La etapa de análisis ha permitido identificar las necesidades que tiene que cubrir la aplicación distribuida. Esta estará formada básicamente por dos elementos: un servidor y una aplicación que se pueda ejecutar en los dispositivos móviles de los usuarios. Con la aplicación móvil se pretende que los usuarios puedan utilizar la información almacenada en el repositorio de triplas [GSICa]. Debido a las características que han solicitado los usuarios esta aplicación móvil deberá tener dos modos de funcionamiento bien diferenciados que podrán funcionar al mismo tiempo. En primer lugar, se deberá generar un modo de funcionamiento en el que los usuarios puedan explorar un mapa para obtener la ubicación de los puntos de interés almacenados en el punto SPARQL. La existencia de estos marcadores se representará mediante una serie de marcadores que indicarán la autoría y el número de tareas que el usuario todavía no haya completado. Este modo de funcionamiento está diseñado para todo tipo de usuarios: desde los que utilicen la aplicación meramente con fines informativos hasta los estudiantes que deseen conocer qué puntos destacados tengan cerca. En segundo lugar, se tendrá un modo de funcionamiento que esté funcionando en segundo plano del dispositivo y en el que vaya monitorizando la posición del usuario para detectar cuándo este se encuentra cerca de la ubicación de un punto de interés. Este modo de funcionamiento estará consumiendo recursos del terminal de manera continua por lo que solo se activará cuando haya vencido un temporizador cuyo valor podrá ser personalizado por el usuario. Este modo de funcionamiento está pensado para usuarios que deseen completar tareas de aprendizaje ya que con las notificaciones se les recordará que hay puntos relevantes cercanos a su posición.

Por otro lado, el servidor del sistema deberá exponer una serie de recursos para que los distintos clientes puedan obtener la información de los puntos de interés de una zona y sus tareas de aprendizaje. También deberá proporcionar un punto para almacenar respuestas y obtener la información de los portafolios. Por lo tanto, el servidor tendrá dos tareas principales: traducir las peticiones sobre la información almacenada en el repositorio de triplas de peticiones HTTP a SPARQL y almacenar respuestas. La justificación de que los clientes realicen las peticiones al servidor en vez de directamente al punto SPARQL es para que un único elemento tenga que realizar esta transformación. Cuando se solicita información de un portafolio también será necesario obtener información almacenada en el punto SPARQL por lo que será más sencillo gestionar estas peticiones desde un único punto que desde varios. Además, de esta forma se realizarán comprobaciones adicionales de seguridad, como que la zona solicitada por el cliente tenga un tamaño inferior a  $0,1^\circ$  de lado y que la zona creada tenga sentido (que el valor de la posición norte sea superior a la del sur y que el oeste sea inferior a la del este).



# Capítulo 4

## Diseño

En este capítulo se describe qué decisiones se han tomado en la fase de diseño de Casual Learn. Se realiza una descripción de la arquitectura lógica de la aplicación analizando, a través de un ejemplo con un usuario, qué función tiene cada uno de los elementos y qué interacciones se producen entre ellos. A continuación se pasa a describir tanto el diseño del servidor y el cliente explicando con más detalle sus principales características y funciones. También se crean una serie de maquetas para indicar cómo será la interfaz gráfica de la aplicación.

### 4.1. Introducción

Una vez finalizada la fase de análisis de acuerdo con la metodología seguida en este TFM 1.2, en este capítulo se pasa a diseñar la aplicación sentando las bases necesarias para cumplir con los requisitos solicitados y registrados en la fase previa (Capítulo 3). La explicación del diseño se inicia desde la vista más general indicando la arquitectura de la aplicación hasta llegar a describir cada una de las partes a desarrollar. La arquitectura general del sistema se explica desde un punto de vista de arquitectura lógica, dando a conocer qué funciones se deberán realizar en cada una de las partes que compongan la aplicación distribuida y qué características se delegarán a servicios externos. Una vez realizada la vista general se pasa a diseñar cada una de las partes a desarrollar en este TFM: el servidor y el cliente de Casual Learn y sus interacciones. En este capítulo se darán a conocer cuáles son las características del servidor, siendo las más relevantes las relacionadas con la exposición de los recursos para la simplificación del acceso a los datos almacenados en el repositorio de triplas utilizado (el servidor ocultará las tecnologías relacionadas con la Web Semántica a los clientes) y en el almacenamiento persistente de las respuestas de los usuarios. En último lugar se habla del cliente de la aplicación, explicando sus modos de funcionamiento (búsqueda activa y pasiva de POI) y de cómo se lleva a cabo la realización de las tareas educativas.

### 4.2. Arquitectura del sistema

La aplicación Casual Learn se ha diseñado como una aplicación distribuida para poder satisfacer los requisitos indicados en el Capítulo 3. En la arquitectura se diferencia el **cliente**, que puede ejecutarse en los dispositivos móviles de los usuarios, y el **servidor**, que se encarga de recuperar la información de las tareas y POI a través de solicitudes al punto SPARQL, además

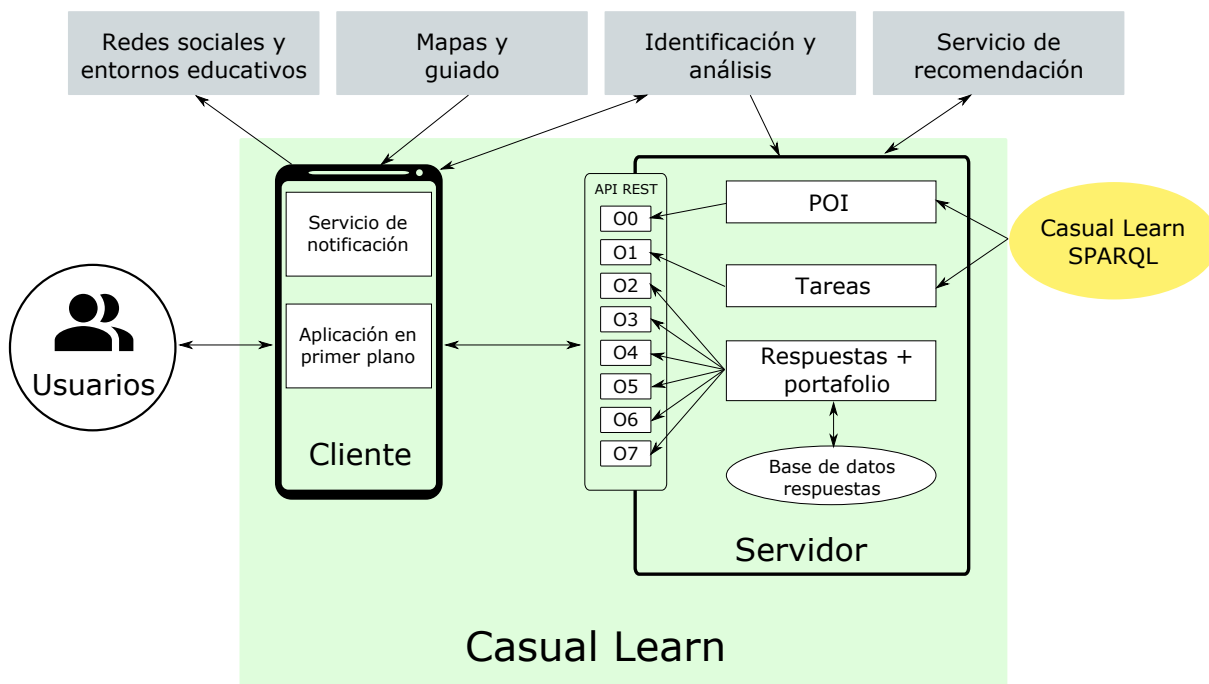


Fig. 4.1: Arquitectura lógica de la aplicación Casual Learn. En este TFM se desarrollan el cliente y el servidor de Casual Learn (en fondo verde). La descripción de las operaciones de la API REST se muestran en la Tabla 4.2. Los servicios externos que se utilizarán se representan con un rectángulo con el fondo de color gris. Casual Learn SPARQL representa el repositorio de triplas donde está almacenada la información de POI y tareas educativas.

de almacenar las respuestas de los usuarios. Los elementos que componen la arquitectura de Casual Learn se muestran en la Fig. 4.1, donde además se dan a conocer los servicios externos en los que se apoyará Casual Learn para su correcto funcionamiento. La descripción de cada uno de los elementos que forman la arquitectura de la Fig. 4.1 se incluyen en la Tabla 4.1. El funcionamiento general de la aplicación, y las interacciones que existen entre cada uno de los elementos de la arquitectura, se explicará mediante un ejemplo de uso:

1. En este ejemplo un estudiante desea conocer los monumentos y edificios relevantes (**POI**) de una ciudad que está visitando. Para ello el estudiante accede al **cliente de Casual Learn** utilizando su dispositivo móvil. El cliente solicita que el usuario se **identifique** para que pueda realizar tareas educativas y para que el sistema almacene sus respuestas.
2. Una vez identificado el usuario mediante el **servicio de identificación**, el estudiante observa a través de una serie de marcadores sobre el **mapa** los lugares de interés de la ciudad visitada que estén registrados en Casual Learn (**aplicación en primer plano**). El cliente obtendrá esta información del **servidor de Casual Learn** a través de la API REST mediante la solicitud de obtención de POI de una tesela<sup>1</sup> llevada a cabo con la petición O0. A su vez el servidor de Casual Learn la estará recuperando de Casual Learn SPARQL.
3. El usuario decide visitar uno de los POI mostrados sobre el mapa, pero como es la primera vez que visita la ciudad necesita ayuda para poder llegar hasta el lugar deseado. Por este motivo activa el **guiado** para que el móvil le indique la ruta que tiene que seguir para poder llegar hasta el POI.

<sup>1</sup>Tesela se define como «cada una de las piezas con que se forma un mosaico» [RAE]. Como se puede ver en la sección 4.4.1, en el cliente se realiza la simplificación de dividir el mapa mundial en teselas cuadradas.

Tabla 4.1: Descripción de los elementos que forman parte de la arquitectura representada en la Fig. 4.1

Elemento	Descripción
Cliente de Casual Learn	Interfaz gráfica de Casual Learn donde los usuarios podrán visualizar la información contenida en Casual Learn SPARQL y podrán realizar tareas educativas. Tendrá un servicio en segundo plano que se encargará de notificar al usuario de POI cercanos.
Servidor de Casual Learn	Expondrá una serie de recursos para que el cliente pueda obtener la información de los POI y las tareas. También permitirá el almacenamiento de las respuestas de los usuarios, su edición y visualización.
Redes sociales y entornos educativos	Elementos externos donde el usuario podrá compartir las respuestas.
Mapas y guiado	Los mapas permitirán al usuario hacerse una idea de su posición frente a la ubicación donde se encuentran los POI. Si tiene dificultades para llegar hasta la ubicación del POI podrá utilizar el guiado para obtener una ruta que le ayude a llegar al lugar deseado.
Identificación y análisis	Servicios para diferenciar a los usuarios que utilicen el sistema y llevar un control de los eventos que realicen en la aplicación.
Servicio de recomendación	Servicio para personalizar el orden de las tareas educativas ofrecidas a cada usuario.
Casual Learn SPARQL	Repositorio de triplas donde está almacenada la información de POI y tareas educativas.

- Una vez que se encuentra próximo al lugar de interés lee su descripción y decide realizar una de las **tareas de aprendizaje**. Del mismo modo que ocurría con la información de los POI, la información de las tareas se recupera del servidor de Casual Learn a través de una petición a la API REST (01). El servidor obtiene esta información mediante una petición al punto SPARQL. El cliente bloqueará la realización de la tarea educativa hasta que el usuario no se encuentre lo suficientemente cerca del POI (a menos de 150m).
- En la tarea que el usuario desea resolver se solicita la realización de una fotografía y contestar a una pregunta. Además, cuando el estudiante finaliza la tarea educativa, el cliente le muestra una pantalla al usuario para que valore la actividad que acaba de realizar. Esta valoración se utilizará para mejorar las **recomendaciones** que se le puedan ofrecer utilizando el **servicio de recomendación**.
- El cliente de Casual Learn envía la **respuesta** que el usuario ha dado junto con la posible valoración que haya podido proporcionar al servidor de Casual Learn para que estos valores sean almacenados en una **base de datos** privada. Para ello hace uso de la operación 03 de la API REST del servidor. La valoración se enviará al **servicio de recomendación** para que la tenga en cuenta en futuras recomendaciones.
- Una vez finalizada la tarea, el usuario decide compartir su respuesta en una **red social** de las permitidas por el cliente de Casual Learn para informar a sus seguidores de lo que está aprendiendo en la visita que está realizando. El cliente adaptará la respuesta para que pueda ser compartida en la red social seleccionada.

Una vez finalizada la visita al lugar que deseaba conocer, el usuario decide continuar dándose un paseo por la zona. Después de unos minutos observa que le ha aparecido una **notificación** de Casual Learn en el móvil (**servicio de notificación** del cliente de Casual Learn). Esta notificación le indica de la existencia de un POI cercano al lugar donde se encuentra. Para que

esta notificación se le muestre al usuario el cliente de Casual Learn ha tenido que monitorizar la ubicación del usuario y consultar al servidor de Casual Learn por los puntos de interés cercanos al lugar donde se encuentre (00). El usuario decide acceder a la notificación para obtener más información del POI. Al realizar esta acción, el cliente solicitará al servidor las tareas de aprendizaje del POI (01), indicando qué usuario es el que solicita esta información para mostrarle en primer lugar las tareas educativas que le puedan resultar más interesantes.

### 4.3. Diseño del servidor

El servidor de la aplicación Casual Learn tendrá que exponer una serie de recursos para facilitar la obtención de los POI de una zona, las tareas de aprendizaje y registrar las respuestas de los usuarios. El servidor deberá almacenar de forma persistente estas respuestas en una base de datos privada para que, si el usuario lo desea, puedan ser accesibles mediante un portafolio. El servidor deberá obtener la información de los POI y las tareas de Casual Learn SPARQL por lo que se deberán diseñar una serie de peticiones de este tipo para obtener los datos necesarios en cada momento. Además, el servidor también tendrá que comunicarse con el servicio de recomendación para almacenar las valoraciones que proporcione cada usuario a cada tarea educativa y obtener el orden en el que se le presentan las tareas educativas al usuario.

#### 4.3.1. Diseño de la API RESTful

El servidor tendrá que ser capaz de atender las peticiones de los clientes solicitando los POI que estén contenidos en un área delimitada y las tareas educativas asociadas a un POI. También deberá proporcionar un recurso donde el cliente pueda enviar las respuestas a las tareas de aprendizaje que el usuario complete y, si el usuario lo desea, permitir que cualquier persona pueda consultar las respuestas que ha generado a través de un portafolio accesible mediante un navegador web. Para conseguir estas características, el servidor se programa como un servicio web de tipo RESTful. Este tipo de servicios se basa en los principios REST [Fie00], destacando:

- **Arquitectura cliente-servidor:** realizando esta separación se consigue facilitar la posibilidad de reutilizar cada una de las partes que se desarrollen. Por ejemplo, el servidor que se implemente en este TFM podrá ser utilizado por clientes diferentes al que se va a presentar en la Sección 4.4.
- **Comunicación sin estado:** cada petición que se realice al servidor debe contener toda la información necesaria para que el servidor pueda procesarla. Por ejemplo, si un usuario está utilizando el cliente de Casual Learn para explorar el mapa de su ciudad, el cliente de la aplicación irá solicitando al servidor cada una de las zonas enviando sus límites. Si las comunicaciones fueran con estado, el cliente podría solicitar al servidor una de las zonas adyacentes a la anterior que hubiera solicitado sin necesidad de enviar los límites. El principal problema de la comunicación con estado es que el servidor tendría que almacenar el estado de los clientes (lo que llevaría a un sistema difícil de escalar por los recursos que estuvieran bloqueados y dificulta el cacheado de información, otro de los principios REST).
- **Interfaz uniforme:** el tipo y el número de operaciones que se puede realizar sobre cada recurso están predefinidas. El número y el tipo de operaciones vendrá determinado por el protocolo que se vaya a utilizar para las comunicaciones. Cada uno de los recursos tendrá asignado un identificador único.

00	GET	/contextos?norte={norte}&este={este}&sur={sur}&oeste={oeste}[&idUser={idUser}]
01	GET	/tareas?contextos={contexto}[&idUser={idUser}]
02	POST	/tareasCompletadas
03	PUT	/tareasCompletadas/{idTareaCompletada}
04	POST	/portafolio
05	GET	/portafolio/{idPortafolio}[?page={page}]
06	PUT	/portafolio/{idPortafolio}
07	GET	/portafolio/{idPortafolio}/{idTareaCompletada}

Fig. 4.2: Operaciones soportadas por el servidor de Casual Learn. Se utilizan URL relativas [BFM05] para indicar la ruta al recurso. Los valores entre corchetes significan que son parámetros opcionales. Los valores entre llaves son obligatorios y deberán sustituirse por el dato que corresponda.

Como se ha indicado previamente, el servidor será de tipo RESTful por lo que deberá cumplir con los principios REST. Además, este tipo de servicios utilizan tecnologías propias de la Web, con lo que se definirá el protocolo utilizado (HTTP [Nie+99]), la interfaz uniforme (sobre cada recurso se podrán utilizar un número limitado de operaciones soportadas por HTTP: GET, POST, etc.) y la manera de crear identificadores únicos para los recursos (mediante IRI [DS05], *Internationalized Resource Identifier*). En la Fig. 4.2 se indican los recursos que expone el servidor y qué operaciones se pueden realizar sobre cada uno de estos recursos (tareas educativas, POI y respuestas). En la Tabla 4.2 se describe cada uno de estos recursos. Además, en el Anexo A se puede consultar la documentación realizada de la API siguiendo el estándar OpenAPI en su versión 3.0.0 [OAS21].

En la Fig. 4.2 se puede ver que existen dos operaciones POST: una para almacenar respuestas (02) y otra para crear un portafolio (04). Para evitar que un atacante pueda comprometer el sistema, se comprueba que el identificador de usuario (que se tiene que enviar en la petición) está registrado en el servicio de identificación (mediante el SDK del servicio de identificación se estarán utilizando estándares como OAuth 2.0 [FIRAUT]). Todo usuario legítimo que envíe una operación POST deberá haberse identificado previamente con el mencionado servicio de identificación. El identificador de usuario se transmitirá entre el cliente de la aplicación y el servidor mediante un protocolo que garantice la integridad y confidencialidad (HTTPS). Del mismo modo, la comprobación entre el servidor y el servicio de identificación también se realiza mediante comunicaciones seguras a través de HTTPS.

Las operaciones 00 y 01 se utilizarán para recuperar los POI y las tareas educativas. Todas las tareas estarán asociadas a un POI y todos los POI tendrán asociado una posición geográfica. Por este motivo, el cliente deberá delimitar una zona de dónde quiera saber qué POI existen. Una vez que ha obtenido los POI podrá utilizar la operación 01 para recuperar las tareas educativas. Como se puede ver, las dos operaciones admiten `idUser`. En esta versión de la aplicación, el envío de este parámetro en la operación 01 tendrá como consecuencia que el servidor se comunique con el servicio de recomendación para devolver al cliente las tareas en el orden en el que les puedan resultar más interesantes. En el caso de la operación 00, para esta versión de la aplicación, el

Tabla 4.2: Descripción de las operaciones que se pueden realizar con el servidor de Casual Learn.

Operación	Descripción
00	Obtención de los POI de una zona. La zona queda descrita mediante su posición al norte, sur, este y oeste. El cliente deberá enviar estos valores como parámetros de la operación. Opcionalmente el cliente podrá enviar el identificador del usuario que esté solicitando la operación para ofrecerle los POI que se adapten más a sus intereses (sin uso en esta versión).
01	Obtención de las tareas de aprendizaje asociadas a un POI. Opcionalmente se podrá enviar el identificador del usuario que está solicitando la operación (el orden de las tareas educativas vendrá determinado por el sistema de recomendación).
02	Registro de una nueva respuesta en el sistema. Todos los datos necesarios (identificador del usuario y metadatos de la respuesta) deberán agregarse al cuerpo de la operación en formato JSON.
03	Edición de una respuesta.
04	Creación de un nuevo portafolio.
05	Obtención de una página del portafolio de un usuario ( <code>idPortafolio</code> ). La página solicitada vendrá determinada mediante un parámetro. Si no se indica una página concreta se envía la página inicial.
06	Edición del estado completo de un portafolio. El usuario puede cambiar su visibilidad (público para todo el mundo o para nadie).
07	Obtención de una respuesta concreta de un usuario.

envío de `idUser` del usuario no tendrá ningún efecto. En futuras versiones se podrá utilizar este identificador para ofrecer al usuario los POI que le puedan resultar de su interés teniendo en cuenta los POI en los que ha realizado tareas previamente.

Las operaciones 02 y 05 permitirán, respectivamente, crear nuevas respuestas y modificar las existentes. Para llevar a cabo las modificaciones, a cada respuesta que envíe el usuario el servidor le asignará un identificador único. Este identificador se devolverá en la respuesta de la operación 02.

Las operaciones de 04 a 07 están relacionadas con el portafolio del usuario. Este portafolio se generará en el servidor de Casual Learn con las respuestas que vayan proporcionando los usuarios. El usuario podrá modificar la visibilidad del portafolio en la creación (04) o en cualquier momento a través de una edición (06). Del mismo modo, el usuario también podrá decidir si las respuestas que envíe al servidor deben estar disponibles inmediatamente o después de un tiempo pseudoaleatorio. Esta segunda preferencia se pensó para que no se pudiera hacer un seguimiento de la ubicación en la que se encuentren los usuarios de la aplicación. Las operaciones 05 y 07 se utilizan para visualizar las respuestas de los usuarios. Si el portafolio de un usuario no está visible el servidor no devolverá la lista de respuestas cuando se realice la operación 05. Del mismo modo, si el usuario tiene oculta una respuesta específica 07 no se mostrará la respuesta del usuario.

Las operaciones 00, 01, 05 y 07 se utilizarán para recuperar información. El servidor responderá a las peticiones de 00 y 01 en formato JSON [Bra14] (*JavaScript Object Notation*). Este es uno de los formatos más populares para compartir información entre aplicaciones de la Web. Por otro lado tanto las peticiones a las operación 05 como a la operación 07 devolverán la información del portafolio en un formato HTML [BC95] (*Hypertext Markup Language*) para que un navegador web pueda mostrar la información de manera agradable a los usuarios.

Como se ha podido observar, algunas de las rutas de las operaciones soportadas por el servidor de Casual Learn en la Fig. 4.2 no corresponden con los elementos del modelo: `contextos` en vez de `POI` y `tareasCompletadas` en vez de `respuestas`. Esto es debido a que el modelo ha evolucionado a medida que se ha desarrollado este TFM. La nomenclatura utilizada en la API del servidor no se ha modificado desde la primera versión. De todas formas, se espera llevar a cabo una actualización en el servidor que utilice los nombres actuales del modelo (posiblemente traducidos al inglés). Por motivos de compatibilidad con las versiones más antiguas del cliente, se mantendrán estos recursos accesibles indicando que se está realizando una redirección hacia los nuevos [FR14, Sección 6.4.2].

### 4.3.2. Peticiones SPARQL

El servidor podrá obtener del punto SPARQL la información de los POI y tareas de aprendizaje que soliciten los usuarios. Esta información se utilizará cuando un cliente realice las llamadas 00, 01, 05 y 07 de la Tabla 4.2. Se necesitará consultar la información de las tareas cuando se solicite un portafolio de usuario ya que, en la base de datos del servidor solamente se almacenarán las respuestas del usuario junto con el identificador de la tarea, pero no se guardará ni el enunciado de la tarea de aprendizaje, ni el nombre del lugar o el lugar donde se encuentre. En esta subsección se van a explicar las consultas más relevantes para el funcionamiento del sistema. El resto de consultas diseñadas serán un extracto o combinación de las presentadas en esta sección.

Listado 4.1: Plantilla de la consulta SPARQL para obtener los POI contenidos en una zona. Con los parámetros utilizados se recupera el identificador del POI (`?contexto`), la ubicación del POI mediante su latitud (`?lat`) y longitud (`?lng`), el autor de los datos (`?creadoPor`), el número de tareas educativas asociadas al POI (`?nTareas`), la descripción del lugar (`?com`) y el enlace de DBpedia (`?wiki`).

```

1 SELECT DISTINCT ?contexto ?lat ?lng ?label ?creadoPor count (DISTINCT ?
2   id) as ?nTareas ?com ?wiki WHERE {
3   ?contexto
4     geo:lat ?lat ;
5     geo:long ?lng ;
6     rdfs:label ?label ;
7     dc:creator ?creadoPor .
8   ?id
9     clp:hasContext ?contexto ;
10    clp:space cls:physical .
11  FILTER (
12    (xsd:decimal (?lat) >= 41.655) &&
13    (xsd:decimal (?lat) < 41.66) &&
14    (xsd:decimal (?lng) >= -4.725) &&
15    (xsd:decimal (?lng) < -4.72)
16  )
17  OPTIONAL { ?contexto rdfs:comment ?com .}
18  OPTIONAL {
19    ?contexto rdfs:seeAlso ?wiki .
20    FILTER ( regex(?wiki , 'dbpedia')) .
21  }
}

```

En primer lugar, con la consulta del Listado 4.1 el servidor recoge del repositorio de triplas la información de los contextos. El servidor realizará esta consulta cuando el cliente solicite obtener los POI de una determinada zona. Por este motivo, el cliente deberá haber proporcionado las coordenadas que limitan el área donde el cliente quiera conocer qué POI existan. En esta

consulta se pueden distinguir los parámetros de la respuesta (indicados mediante las variables), que estará compuesta por el identificador del POI, su ubicación –mediante su latitud y longitud–, el nombre del lugar o monumento, el número de tareas que el usuario se podrá encontrar en el POI y, opcionalmente, la descripción del lugar o el monumento y el enlace para obtener más información de DBpedia. Se establece un filtro con las posiciones del área rectangular que ha enviado el usuario, habiendo realizado previamente las comprobaciones de formato y valor de estos datos. Para finalizar la consulta de obtención de la información de los POI de una zona se intenta obtener información extra que le pudiera resultar útil al cliente. Para ello se realiza una búsqueda del enlace de DBpedia que pudieran estar relacionado con cada lugar. Posteriormente, el servidor transformará este enlace de DBpedia a uno de Wikipedia.

Listado 4.2: Plantilla SPARQL para solicitar las tareas educativas asociadas a un POI. Los parámetros que se obtendrán en la respuesta son el IRI de la tarea (?task), el tipo esperado de respuesta (?tR), la URL de la imagen en baja calidad (?thumb), la URL de la imagen en alta calidad (?imag), el autor (?imgA) y la licencia (?imgL) de la imagen, el nombre del POI donde se realiza la tarea (?com), el enunciado de la tarea (?rAT), la respuesta correcta (?rE) y el autor de la tarea (?cP).

```

1 SELECT DISTINCT ?task ?tR ?thumb ?imag ?com ?rAT ?rE ?cP ?imgA ?imgL
  WHERE {
2   ?task
3     clp:hasContext <{IRIContexto}> ;
4     clp:answerType ?tR ;
5     clp:associatedTextResource ?rAT ;
6     clp:space cls:physical ;
7     dc:creator ?cP .
8   OPTIONAL {
9     ?task
10      edbo:thumbnail ?thumb .
11      OPTIONAL { ?thumb dc:creator ?imgA . }
12      OPTIONAL { ?thumb dc:license ?imgL . }
13    }
14   OPTIONAL { ?task clp:image ?imag . }
15   OPTIONAL { ?task clp:expectedAnswer ?rE . }
16   OPTIONAL { ?task rdfs:comment ?com . }
17 }

```

En segundo lugar, con el código presentado en el Listado 4.2 se construirá una consulta para obtener todas las tareas asociadas a un POI. Esta consulta se realizará cuando un usuario pulse sobre el marcador de un POI o acceda a una notificación en la que se indica la existencia de un POI cercano a la posición del usuario. En este caso, la información que se extrae de cada tarea educativa es el tipo esperado de respuesta de la tarea, su enunciado y el autor. Opcionalmente se obtiene la imagen descriptiva de la tarea, tanto en calidad original como en baja calidad, el autor y licencia de la imagen, la respuesta esperada y el nombre del lugar o el monumento. La información opcional de la consulta del Listado 4.2, aunque no sea necesaria para que el usuario entienda lo que se le solicita realizar en la tarea educativa, enriquecerá su contenido y lo hará más atractivo. Por ejemplo, la fotografía del lugar donde se vaya a realizar la tarea educativa puede que no sea muy relevante para su realización, pero puede favorecer que al usuario le resulte atractiva la fotografía y acceda a leer la descripción de la tarea.

Listado 4.3: Solicitud SPARQL para obtener todas las tareas de Casual Learn SPARQL. Los parámetros de la respuesta serán el IRI de la tarea educativa (?task), la ubicación donde se encuentra el POI con el que está relacionado (?lat y ?lng) y el tipo esperado de respuesta (?tR).

```

1 SELECT DISTINCT ?task ?lat ?lng ?tR WHERE {
2   ?ctx
3     geo:lat ?lat ;

```



```

4   geo:long ?lng .
5   ?task
6   clp:hasContext ?ctx ;
7   clp:space cls:physical ;
8   clp:answerType ?tR .
9 }

```

En último lugar, en el Listado 4.3 se encuentra la petición que se utilizará para recuperar la información necesaria para el servicio de recomendación. Esta información se reduce a los identificadores de las tareas educativas junto con la ubicación del punto de interés donde se debe realizar cada una de estas tareas y el tipo de respuesta que deberá proporcionar el usuario. Tanto en el código de este Listado como en el Listado 4.2 únicamente se obtienen las tareas educativas que se tengan que realizar en el espacio físico, es decir, las tareas que para poder realizarlas se necesite estar presente en la ubicación del POI. Como se indicó en la Sección 2.3, otros espacios que están presentes en Casual Learn SPARQL son el espacio mapa virtual, tareas educativas ideadas para realizarlas a través de un mapa virtual en cualquier lugar, y el espacio virtual, con tareas pensadas para ser realizadas a través de un entorno de aprendizaje virtual como pudiera ser Moodle.

### 4.3.3. El portafolio de la aplicación

El portafolio de la aplicación se utilizará para que los usuarios que lo deseen puedan compartir sus respuestas con cualquier persona. El portafolio se podrá visualizar desde cualquier dispositivo a través de un navegador web, por lo que será necesario que se adapte a las diferentes resoluciones de pantalla. La interfaz del portafolio será similar a las que se puede visualizar en las maquetas de la Fig. 4.3 y de la Fig. 4.4.

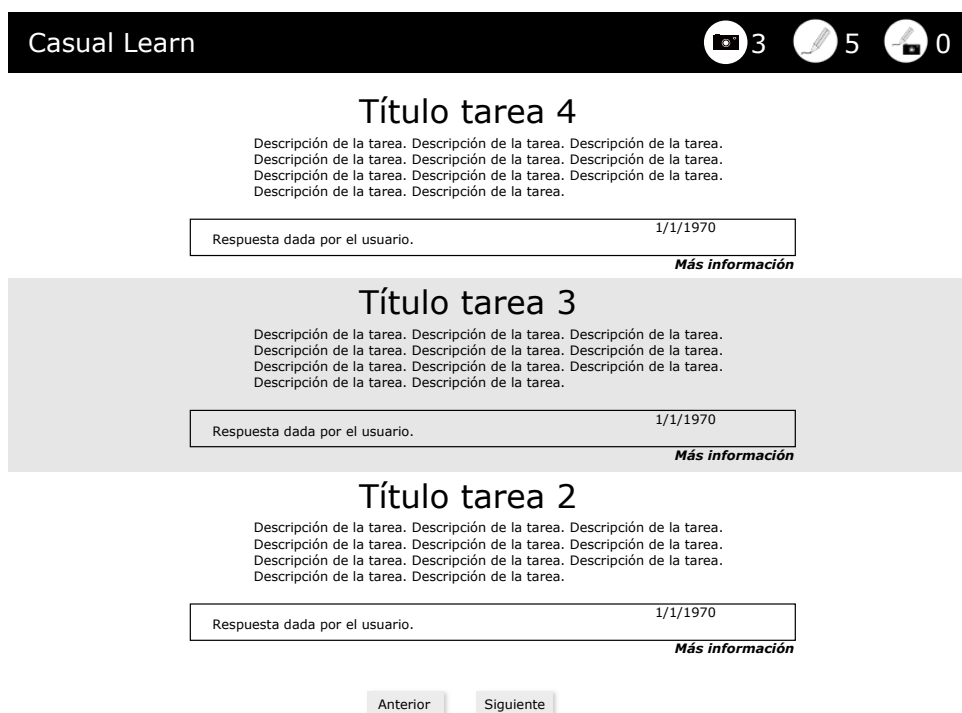


Fig. 4.3: Maqueta del portafolio de la aplicación Casual Learn. Lista de respuestas públicas realizadas por un usuario. En esta lista se podrá ver el número de tareas completadas por tipo esperado de respuesta.

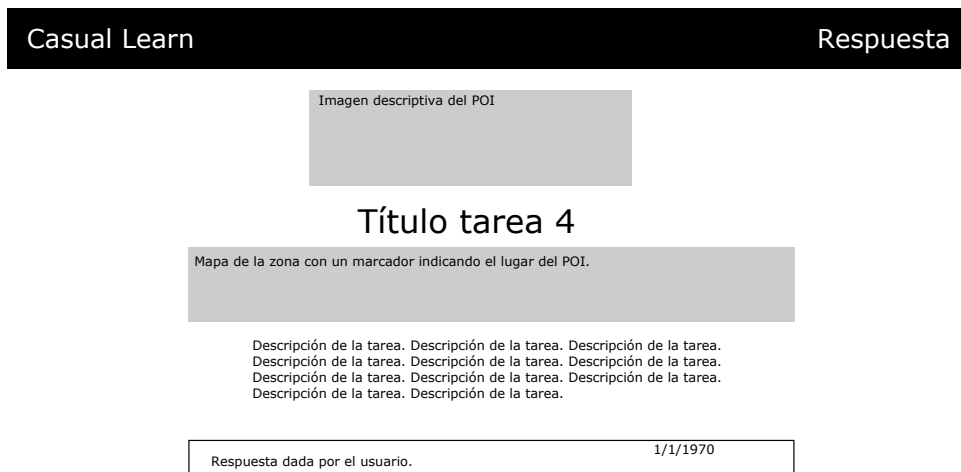


Fig. 4.4: Maqueta del portafolio de la aplicación Casual Learn. Vista individual de una respuesta. En la vista individual se podrá ver el POI donde se inició la tarea y su imagen descriptiva.

En la Fig. 4.3 se puede ver una página del portafolio donde se muestra una lista de tres respuestas y unos botones de navegación para cambiar de página. Esta página se habrá obtenido al realizar una petición a la operación 05 de la Tabla 4.2. Las respuestas se muestran de más recientes a más antiguas. En cada una de las secciones destinadas a cada respuesta se muestra el nombre del lugar en el que el usuario ha comenzado la tarea educativa, qué se le pedía realizar, cuál ha sido su respuesta, la fecha de realización de la tarea y un enlace para acceder a la información completa (Fig. 4.4). Cada página del listado del portafolio mostrará hasta tres respuestas. El cliente puede determinar qué página desea consultar a través de la petición a la operación 05. Si no se indica ningún valor se tomará por defecto que se está solicitando la primera página del listado del portafolio (respuestas más recientes). En esta primera página también se indicará un resumen de tareas completadas diferenciadas por su tipo esperado de respuesta.

Una página similar a la de la Fig. 4.4 se puede obtener al realizar una consulta a la operación 07 de la Tabla 4.2. En ella se mostrará la información completa de la respuesta, donde se proporcionará la misma información que en la lista de respuestas y además la imagen descriptiva del lugar (si estuviera disponible) y un mapa virtual con un marcador interactivo donde se destacará la ubicación del POI. Al pulsar sobre el marcador se podrá acceder a distintos proveedores de mapas para poder iniciar una exploración de la zona cercana al lugar donde se ha iniciado la tarea educativa.

Los datos que se mostrarán en el portafolio serán una combinación de la información almacenada en el repositorio de triplas junto con las respuestas de los usuarios, almacenadas en la base de datos del servidor de la aplicación (en la arquitectura –Sección 4.2– se mostraba una base de datos privada donde se almacenaban las respuestas de los usuarios). De esta forma, en la base de datos privada se estará almacenando la mínima cantidad de información necesaria, haciendo que la aplicación pueda soportar un mayor número de usuarios y evitando almacenar repetidas

veces la misma información. Además, si se detecta algún error en la información accesible desde el punto SPARQL, se podrá subsanar, garantizando que la información mostrada en el portafolio esté actualizada. Este método tiene como desventaja que el número de peticiones que se podrían realizar a Casual Learn SPARQL fuera elevado en usuarios con un número elevado de tareas completadas públicas. Por esta desventaja se decidió limitar el número de respuestas que devuelve el servidor al realizar la petición 05.

## 4.4. Diseño del cliente

El cliente de Casual Learn será la interfaz con la que los usuarios interactuarán con el sistema. En la Fig. 4.5 se pueden ver las maquetas de algunas de las pantallas de la aplicación. Como se puede observar en estas maquetas, la interfaz de la aplicación será similar a otras aplicaciones existentes en el mercado (por ejemplo, las vistas en la Sección 2.2 o Google Maps<sup>2</sup>). El motivo por el cual se decide diseñar la interfaz gráfica de manera similar a la de las aplicaciones más populares es para que la curva de aprendizaje para poder empezar a utilizar la aplicación sea fácil de superar. El diseño de la interfaz de usuario de la aplicación se realizará siguiendo las recomendaciones de la guía de diseño Material Design<sup>3</sup>, diseño seguido por las aplicaciones desarrolladas por Google.

El usuario podrá decidir si identificarse una vez que inicie la aplicación. Si decide utilizar el cliente sin iniciar sesión podrá obtener información tanto de los POI como de sus tareas de aprendizaje asociadas. De esta forma cualquier usuario podrá familiarizarse con el funcionamiento de la aplicación sin necesidad de proporcionar información personal. Estos usuarios invitados no podrán realizar ninguna tarea educativa hasta que se identifiquen. Otra ventaja que dispondrán los usuarios identificados frente a los invitados es que se les ofrecerá, mediante notificaciones del sistema, lugares cercanos a su posición para que los visiten.

### 4.4.1. Búsqueda activa de POI

Los usuarios de la aplicación podrán obtener la información de puntos de interés y tareas educativas a través de una pantalla similar a la de la maqueta mostrada en la Fig. 4.5a. El proceso por el cual se obtiene esta información será transparente para el usuario y solamente se precisará de una conexión a Internet para poder realizar las consultas y obtener las respuestas del servidor de Casual Learn. La base de la pantalla donde se muestra la información de POI y tareas de aprendizaje será un mapa con el que el usuario podrá interactuar para desplazarse por él o incrementar o disminuir su nivel de zoom. Si en la zona que esté visualizando existen POI se le mostrarán al usuario a través de una serie de marcadores en los que se indicará el número de tareas educativas que el usuario todavía no haya realizado. El cliente habrá tenido que solicitar al servidor la información de la zona mostrada utilizando la operación 00 de la subSección 4.3.1.

Para evitar que el cliente esté continuamente solicitando información de POI al servidor se deberá almacenar en el dispositivo qué zonas se han consultado con anterioridad y qué información se tiene de cada una de estas zonas. Para ello, el cliente de Casual Learn realizará la simplificación de dividir el mapamundi en teselas cuadradas de lado constante. El lado de las teselas se mide en grados sexagesimales y se ha fijado en  $0,0254^\circ$ . Utilizando la simplificación de la fórmula del semiverseno [Rob57] la diagonal de cada tesela sería de, aproximadamente,

---

<sup>2</sup><https://www.google.com/maps/about>

<sup>3</sup><https://material.io/>



Fig. 4.5: Maquetas de las pantallas más relevantes del cliente Android de Casual Learn. Para su diseño se ha decidido utilizar la escala de grises para destacar el formato general de cada una de las pantallas. En el Capítulo 5 se mostrarán los colores que se han elegido para el cliente en producción. (a) Pantalla de mapas donde los usuarios podrán consultar los POI existentes en una zona. Al pulsar sobre un POI aparecerá la lista de tareas educativas asociadas a este. El mapa mostrado ha sido generado mediante *city-roads* (<https://github.com/anvaka/city-roads>) utilizando datos de OpenStreetMap [OSM]. (b) Pantalla donde el usuario realizará la tarea de aprendizaje seleccionada. La interfaz de esta pantalla se adaptará al tipo esperado de respuesta para cada tarea educativa. (c) Pantalla donde el usuario podrá valorar la tarea educativa (si lo desea) y compartir la respuesta proporcionada en las distintas redes sociales (RRSS) y entornos educativos soportados por la aplicación. (d) Pantalla con la lista de tareas completada por el usuario. Se proporciona información suficiente (tipo esperado de respuesta, nombre del POI e instante de realización) para que el usuario pueda diferenciar las distintas respuestas.

4km. Las coordenadas de origen que se tomarán para llevar a cabo la división en teselas serán fijas. Con esta información el cliente será capaz de calcular las teselas que se estén mostrando al usuario en cada momento y conocerá si dispone de la información de los POI en la memoria local del dispositivo o tiene que recuperar la información de las teselas que no tenga disponibles del servidor (que a su vez el servidor obtendrá esta información de Casual Learn SPARQL mediante la consulta del Listado 4.1). El cliente proporcionará una validez de un día a la información de la tesela que se hubiera recuperado del servidor para evitar que no se le muestre al usuario información que se hubiera agregado posteriormente al repositorio de triplas. Con esta acción se consigue llegar a una solución de compromiso en la que se estará evitando solicitar constantemente la misma información al servidor y el usuario dispondrá de información actualizada. Además, como la mayoría de usuarios que utilicen la aplicación Casual Learn utilizarán el cliente desarrollado en este TFM para acceder a la información almacenada en el repositorio de triplas se estará aprovechando la memoria caché que gestiona Virtuoso. Por ejemplo, si dos estudiantes deciden quedar para realizar tareas educativas a la vez, Virtuoso devolverá al servidor con más celeridad las respuestas a las consultas de POI que realice el cliente del estudiante que haya iniciado la búsqueda de contextos en segundo lugar.

Una vez que el cliente disponga de la información de los POI de la zona del mapa que se esté mostrando en ese momento se representará en pantalla. La representación de la información se reducirá a una serie de marcadores. Cuando el usuario pulse sobre uno de estos marcadores se mostrará la información del POI que esté disponible. Además, se consultará al servidor de la

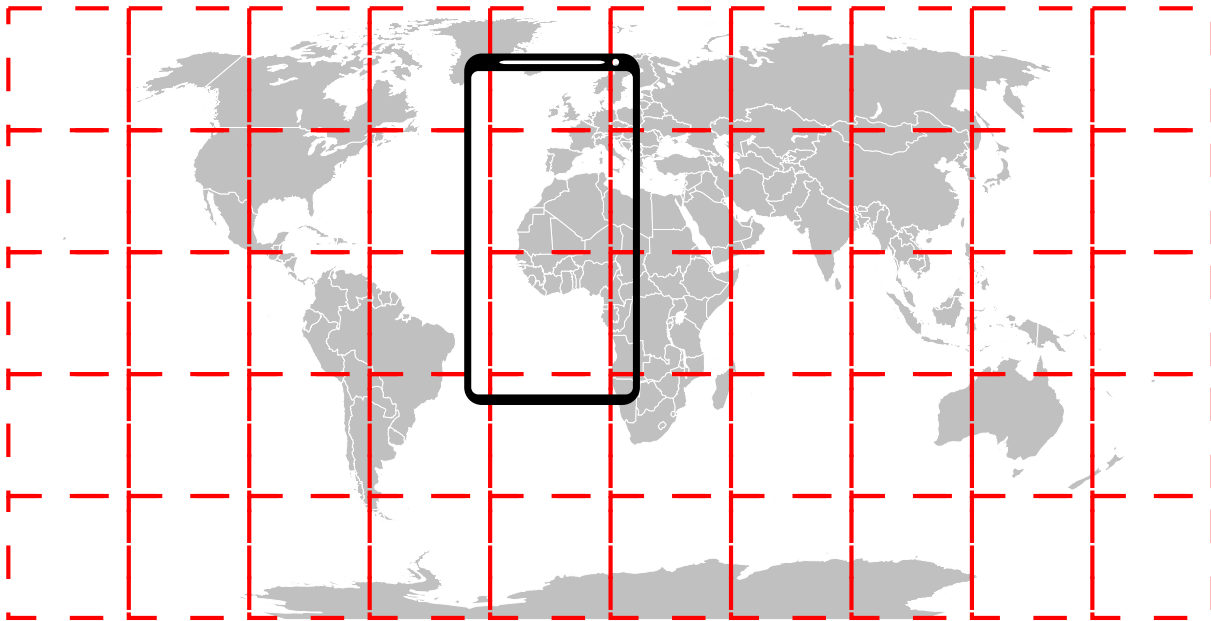


Fig. 4.6: Ejemplificación de la simplificación que realiza el cliente de Casual Learn para dividir el mapa-mundi en teselas cuadradas de tamaño constante. El origen desde donde se inicia la división del mapa en teselas será fijo para el cliente desarrollado en este TFM. En el caso del ejemplo representado, el cliente deberá solicitar al servidor como máximo la información de los POI de 12 teselas. La totalidad o parte de estas teselas se estarían mostrando en la pantalla del dispositivo móvil. Si el cliente dispone de la información de alguna de las teselas en la memoria local del dispositivo del usuario –y esta información continúa siendo válida– no tiene que realizar la solicitud de los POI de esta tesela al servidor.

aplicación por las tareas de aprendizaje que estén asociadas a este contexto –mediante la solicitud del Listado 4.2– si no se dispone de esta información en la memoria local del dispositivo. Al igual que sucedía con la información de los POI, los datos de las tareas educativas que se encuentren en la memoria local del dispositivo tendrán una validez de un día. Pasado ese día, si el usuario vuelve a pulsar sobre ese POI el cliente deberá volver a solicitar al servidor la información de las tareas educativas asociadas al POI. Una vez que se disponga de la información de las tareas en la memoria local del dispositivo se le mostrará al usuario una lista con cada una de ellas dentro de la información del POI. Al pulsar sobre uno de los elementos de la lista, y si el usuario se encuentra cerca del POI, se le permitirá al usuario realizar la tarea.

La pantalla de la maqueta de la Fig. 4.5a muestra otros tres elementos aparte del mapa y los marcadores de POI y usuario: un menú en la barra de título para acceder a la lista de tareas completadas y a los ajustes de la aplicación; un cuadro de texto para realizar búsquedas y un botón para centrar la vista en la posición del usuario. Con el cuadro de texto el usuario podrá centrar el mapa en la ubicación de los municipios de Castilla y León. El registro utilizado para diseñar esta característica está disponible en el Portal de Datos Abiertos de Castilla y León como «Municipios de Castilla y León»<sup>4</sup>. Cuando el usuario comience a escribir el nombre del municipio que desea buscar le aparecerán sugerencias de nombres de municipios. Cuando el usuario escriba una letra se realizará un filtrado utilizando los nombres de los municipios disponibles. El orden de los municipios sugeridos vendrá determinado por su población: en la parte superior de las sugerencias se colocarán los municipios de mayor población ya que son los más conocidos y será más probable que el usuario quiera centrar el mapa en uno de estos municipios (los municipios más populares suelen tener una mayor cantidad de POI debido a la importancia del lugar). Cuando el usuario seleccione uno de los elementos sugeridos, la pantalla de la Fig. 4.5a centrará

<sup>4</sup>[datosabiertos.jcyl.es/web/jcyl/set/es/sector-publico/municipios/1284278782067](https://datosabiertos.jcyl.es/web/jcyl/set/es/sector-publico/municipios/1284278782067)

el mapa en las coordenadas que se tengan del municipio.

#### 4.4.2. Búsqueda pasiva de contextos

Los usuarios que se hayan identificado podrán tener activo un servicio en segundo plano que busque POI cercanos a su posición y les notifique de su existencia. Para ello el cliente dispondrá de un servicio en segundo plano que monitorizará la ubicación del usuario una vez que haya vencido un temporizador, cuyo valor pueda ser personalizado por el usuario, que determina el tiempo mínimo que debe existir entre cada notificación que genere el cliente de Casual Learn.

Una vez que el temporizador haya vencido, el servicio en segundo plano recuperará de forma periódica la ubicación del usuario y comprobará a qué distancia se encuentran los POI. Si alguno de los POI está cerca de la posición del usuario, no se ha notificado de este POI al usuario en más de una semana y el usuario no ha realizado ninguna tarea educativa asociada al POI se genera una notificación para el usuario invitándole a visitarlo. La información de los POI de la zona donde se encuentre el usuario se recupera del servidor de Casual Learn. Por cómo se ha diseñado la operación 00 del servidor, el cliente deberá proporcionar los límites de la zona de la que desea obtener los puntos de interés. Estos límites se obtienen sumando (o restando) un valor fijo a la posición del usuario. Solamente cuando el usuario salga de la zona descargada (o haya pasado más de un día desde la última solicitud de POI) el servicio en segundo plano del cliente de Casual Learn volverá a solicitar al servidor la información de los POI cercanos a la zona donde se encuentre el usuario.

Cuando el usuario pulse en la notificación se le mostrará una pantalla con toda la información disponible del POI que se le ha sugerido visitar. Además, el cliente solicitará al servidor la lista de tareas de aprendizaje asociadas al POI indicando el identificador del usuario. De esta forma, el servidor enviará en la respuesta de esta petición la lista de tareas educativas en el orden recomendado para este usuario. Cuando se disponga de esta información se representará en la pantalla informativa del POI con una lista similar a la que se utilizaba para mostrar las tareas educativas en la búsqueda activa de contextos.

#### 4.4.3. Realización de tareas educativas y edición de respuestas

Cuando un usuario identificado decida realizar una de las tareas de aprendizaje de las mostradas dentro de la información del punto de interés y se encuentre cercano a la ubicación del POI se le mostrará una pantalla similar a la maqueta de la Fig. 4.5b. Esta pantalla se adaptará al tipo de respuesta de la tarea que se vaya a realizar. Por ejemplo, en una tarea en la que se solicite realizar una fotografía la pantalla deberá mostrar un botón para acceder a la cámara del dispositivo mientras que otra en la que se solicite responder a una pregunta no se deberá mostrar dicho botón. Aparte de los elementos que se generen para adaptar la pantalla al tipo de respuesta de la tarea, esta pantalla mostrará el nombre del monumento o lugar donde se vaya a iniciar la tarea, el enunciado donde se le explique al usuario qué tarea tiene que realizar, una caja de texto donde podrá agregar sus notas o contestar a las preguntas que se soliciten (en las tareas en las que el tipo esperado de respuesta sea textual) y un botón para finalizar la realización de la tarea educativa. Además, si en la información de la tarea se encuentra una imagen asociada se mostrará entre la etiqueta del lugar y el enunciado de la tarea.

Cuando el usuario pulse en el botón de terminar de contestar a la tarea se le mostrará una pantalla (maqueta representada en la Fig. 4.5c) donde podrá valorar la tarea educativa (información que se utilizará para personalizar el orden en el que se muestren las tareas relacionadas

con la búsqueda pasiva de contextos) y compartir la respuesta dada en las diferentes RRSS y entornos educativos soportados por la aplicación. El usuario podrá omitir las dos acciones y realizarlas más tarde si lo desea.

En la maqueta representada en la Fig. 4.5d se muestra la pantalla de tareas completadas por el usuario. Al pulsar sobre alguno de los elementos representados en esta lista el usuario podrá visualizar la respuesta dada en una pantalla con un formato similar a la de la Fig. 4.5b. Desde esta pantalla el usuario podrá compartir la respuesta actual con las RRSS y entornos educativos ofrecidos en la pantalla de la Fig. 4.5c y editar la respuesta. A la hora del envío de las respuestas al servidor se deberá tener en cuenta que el usuario puede estar realizando las tareas educativas sin conexión a Internet (por ejemplo, puede existir un usuario que haya consultado los POI en su casa a través de una red Wi-Fi, habiendo guardado en la memoria de su dispositivo tanto la información de los POI de una zona como las tareas de los POI consultados, y no disponer de una tarifa de datos contratada). En estos casos, el cliente de Casual Learn deberá crear un proceso en segundo plano que compruebe cuándo le será posible el envío de los datos de las respuestas completadas por el usuario. Además, se habilitará una preferencia en el menú de ajustes para aquellos usuarios que deseen que la aplicación se conecte las mínimas veces posibles a Internet cuando el dispositivo no se encuentre conectado a una red Wi-Fi. Si el usuario activa esta característica las respuestas que proporcione solo se enviarán al servidor de Casual Learn cuando el dispositivo móvil se conecte a una red Wi-Fi.

El cliente de Casual Learn enviará las respuestas del usuario cuando haya finalizado la valoración de la tarea (02) o cuando la haya editado (03). De esta forma, las respuestas se guardarán tanto en el dispositivo móvil del usuario como en el servidor del sistema (en este segundo caso para utilizarlas en el portafolio). Como la aplicación va a ser utilizada por estudiantes de los últimos cursos de Educación Secundaria Obligatoria y bachillerato (usuarios menores de edad) al servidor no se enviarán los elementos multimedia que se generen desde la aplicación, solamente se enviarán las respuestas textuales. De esta forma, los elementos multimedia solo los podrá visualizar desde su dispositivo. También podrá compartir estos elementos con otras personas a través de las RRSS y entornos educativos soportados por la aplicación, pero en este caso la publicación se realiza con consentimiento explícito del usuario, no como con la carga automática de respuestas en el servidor de Casual Learn.

## 4.5. Conclusiones

La etapa de diseño de la metodología ágil [SS05] que se está utilizando para la creación de Casual Learn ha servido para concretar los aspectos a realizar en la fase de implementación. Con la fase de diseño se ha especificado cuál va a ser la arquitectura lógica de la aplicación y qué funciones tendrá que realizar el cliente y el servidor que se van a desarrollar. Este segundo elemento se ha diseñado con la idea de que pueda ser utilizado en otras aplicaciones. Por ejemplo, podría ser utilizado por un desarrollador web que esté creando una nueva página y necesite la información de los puntos de interés de Castilla y León. El servidor de Casual Learn podría servir a este desarrollador para obtener los datos que necesita sin tener que conocer las tecnologías de la Web Semántica, ya que únicamente deberá realizar peticiones `GET` a una API RESTful y las respuestas estarán en un formato nativo de la Web como es JSON (*JavaScript Object Notation*).

El cliente de Casual Learn se ha diseñado con el objetivo de que se pueda ejecutar en la mayor cantidad posible de dispositivos. Por este motivo, su interfaz tiene la cantidad mínima de elementos para dispositivos de bajas prestaciones puedan mostrar la interfaz de forma correcta. Además, en el diseño de la aplicación también se tiene en cuenta la existencia de usuarios que

no disponen de una tarifa de datos ilimitadas en sus dispositivos móviles, por lo que el cliente de Casual Learn hace un uso responsable de estos (almacenamiento en memoria local de la información de POI y tareas educativas, posibilidad de utilizar únicamente conexiones Wi-Fi para efectuar la carga de las respuestas en el servidor de Casual Learn, etc.).

El cuadro de búsqueda de municipios de la pantalla de la maqueta de la Fig. 4.5 puede servir para ampliar el público objetivo de la aplicación Casual Learn: puede ser utilizado por turistas. Por ejemplo, un turista que tenga pensado realizar una visita a un municipio de Castilla y León y quiera conocer de antemano la ubicación de los POI para planificar una ruta por los lugares que le resulten más interesantes. Este tipo de usuario podrá utilizar la aplicación como invitado para llevar a cabo la planificación del viaje. Además, una vez que esté visitando la ciudad puede que le resulte interesante la característica de la búsqueda de contextos en segundo plano para que la aplicación sea la que le notifique cuando se encuentre próximo a un POI.



# Capítulo 5

## Implementación

En este capítulo se indica cómo se ha desarrollado tanto el cliente como el servidor de Casual Learn. Por un lado, el servidor se desarrolla para ser ejecutado sobre una máquina que disponga de JRE, utilizando MongoDB para almacenar de forma permanente las respuestas de los estudiantes y Recombee para ordenar las tareas de aprendizaje que se le ofrecen de manera personalizada a los usuarios. Por otro, el cliente de Casual Learn se implementa como una aplicación móvil nativa del sistema operativo Android, contará con dos modos de búsqueda de POI y permitirá realizar tareas educativas a los usuarios registrados. Se muestran ejemplos de cómo es la interfaz gráfica de la versión actual del portafolio y del cliente. Para terminar el capítulo se explican las comunicaciones entre los distintos servicios cuando el usuario realiza una acción u otra.

### 5.1. Introducción

Definido el diseño de la aplicación distribuida se pasa ahora a la fase de implementación de las metodologías ágiles [SS05]. En ella se definirán las tecnologías que se van a utilizar para implementar cada una de las partes del sistema que necesitan ser desarrolladas. Por consiguiente también se tendrán que especificar los servicios externos al sistema que van a utilizarse para delegar alguna de las funciones necesarias para el correcto funcionamiento de la aplicación.

En esta sección se explicará cómo se ha realizado la implementación del servidor, cómo se ha estructurado su código y qué servicios se han elegido para identificar a los usuarios, mantener la persistencia de los datos y que a cada usuario se le puedan ordenar las tareas adaptándolas a sus gustos. La forma de explicar cómo se ha implementado el cliente será diferente, ya que se utilizarán las pantallas que componen el mencionado cliente para resaltar sus características. Por ello se hablará de la identificación de los usuarios –y de las ventajas que tendrán los usuarios identificados frente a los no identificados–, la configuración del portafolio, la búsqueda en activa y pasiva de puntos de interés y la realización y edición de tareas educativas. Para finalizar este capítulo se tendrá una sección en la que se resaltarán las interacciones entre las distintas partes de la aplicación para que se satisfagan los requisitos establecidos en la etapa de análisis (Capítulo 3).

## 5.2. Implementación del servidor

Las dos funciones principales del servidor de Casual Learn son: ocultar al cliente las tecnologías relacionadas con la Web Semántica y almacenar de forma persistente las respuestas de los usuarios. El lenguaje escogido para llevar a cabo el desarrollo del servidor ha sido **Java** debido a la experiencia que se tiene con este lenguaje, a su gran popularidad –tercer puesto entre los lenguajes de programación más populares del año 2020 [Git]– lo que facilitará encontrar desarrolladores que puedan seguir implementando características en el servidor y el alto número de proyectos disponibles para facilitar la implementación de nuevas características (más de 14 millones según Maven Repository<sup>1</sup>). El código desarrollado se ejecutará dentro del entorno de ejecución JRE (*Java Runtime Environment*) de la implementación de OpenJDK<sup>2</sup>. Todas las herramientas externas se han integrado en la estructura del proyecto del servidor mediante la utilización de Maven [Fou], una herramienta para gestionar proyectos *software*, facilitando tanto la integración de bibliotecas externas (mediante la configuración de un único fichero `pom.xml`) como la compilación del propio proyecto con sus dependencias.

La implementación de la API RESTful diseñada en el Capítulo 4 se realizará utilizando el *framework* Restlet<sup>3</sup> en su última versión (2.4.3). Restlet proporciona un conjunto de clases y rutinas que se pueden utilizar para crear rápidamente aplicaciones Web ya que ayuda a gestionar características de HTTP como el enrutar las solicitudes hacia el método encargado de procesar, la negociación de contenido de las peticiones, la gestión de errores y comunicación con el cliente, etc. [Lou13, Parte 1]. Dentro del *framework* Restlet se encuentran un conjunto de extensiones que no pertenecen al núcleo de Restlet, compuesto por la API Restlet y el motor que implementa la API [Lou13, Parte 1]. Estas extensiones pueden proporcionar compatibilidad con nuevos tipos de representaciones (por ejemplo, JSON, JAXB, etc.) y conectores para ayudar al motor de Restlet (por ejemplo, Jetty está recomendado para entornos en producción [Lou13, Apéndice A]).

La estructura de código seguida para el servidor Web estará condicionada por los recursos que se van a exponer en la API RESTful siguiendo el diseño de la Fig. 4.2. En el Listado 5.1 puede observarse la organización de los elementos que formarán el código del servidor de Casual Learn, donde puede observarse la existencia del fichero `pom.xml`. En este fichero se incluirán todas las dependencias que utilice el servidor para que, a través de Maven, sean descargadas de repositorios remotos y agregadas al producto final del servidor. En el directorio `srv` se encuentra el código desarrollado. Contará con dos subdirectorios: `casualLearnPasarela`, con todos los métodos necesarios para que el servidor funcione de manera correcta; `main/resources`, con el fichero `log4j.properties` donde se indican las directrices para el registro de eventos en el servidor. A continuación se pasa a explicar la estructura interna del directorio `casualLearnPasarela`:

- **contextos**: este directorio contiene la clase `Contextos`, clase encargada de gestionar las peticiones realizadas por el cliente para la obtención de los POI. Tendrá un método encargado de realizar la petición correctamente formulada a Casual Learn SPARQL. La petición se llevará a cabo utilizando las bibliotecas necesarias del proyecto Apache HttpComponents<sup>4</sup> (versión 4.5.3). La clase `Contextos` heredará de `org.restlet.resource.ServerResource`.
- **creacion**: contiene la clase `CreacionIdentificadores` que tendrá los métodos neces-

---

<sup>1</sup><https://mvnrepository.com/>

<sup>2</sup><https://openjdk.java.net>

<sup>3</sup><https://restlet.talend.com/>

<sup>4</sup><https://hc.apache.org/>

Listado 5.1: Estructura de paquetes y ficheros del código del servidor de Casual Learn.

```

1  /srv/
2  casualLearnPasarela/
3      contextos/Contextos.java
4      creacion/CreacionIdentificadores.java
5  persistencia/Persistencia.java
6  portafolio/
7      helpers/
8          Estadisticas.java
9          RespuestaGeneral.java
10         RespuestaIndividual.java
11         Siguiente.java
12     plantillas/
13         PlantillaGeneral.vtl
14         PlantillaVacía.vtl
15         RespuestaIndividual.vtl
16     Portafolio.java
17     PortafolioTarea.java
18     PortafolioUsuario.java
19     respuestas/
20         TareaCompletada.java
21         TareasCompletadas.java
22     servidor/
23         Aplicacion.java
24         Componente.java
25     tareas/Tareas.java
26     util/
27         Auxiliar.java
28         ConsultasSparql.java
29     main/resources/log4j.properties
30 /pom.xml

```

rios para que en cada inicio del servidor se almacene en el servicio de recomendación la información que necesite para su correcto funcionamiento. Esta información se obtiene de Casual Learn SPARQL realizando la solicitud de la consulta 4.3. Se recuerda que con esta consulta se obtienen todos los identificadores de las tareas educativas, su tipo de respuesta esperada y las coordenadas del lugar donde se puede iniciar la tarea. Como no se espera que la información del punto SPARQL cambie constantemente no es necesario que se esté constantemente monitorizando si se ha agregado nueva información al repositorio de triplas. Desde que Casual Learn está en producción únicamente se han agregado nuevas tareas educativas para las experiencias piloto (Sección 6.3). En estos tres casos se ha tenido que reiniciar el servidor para que recuperara del repositorio de triplas la nueva información y la enviara al servicio de recomendación.

- **persistencia:** se encuentra la clase `Persistencia` utilizada para almacenar en la base de datos privada del servidor las respuestas que proporcionen los usuarios del sistema. Esta clase tendrá un método para comprobar si el usuarios que está intentando registrar una nueva respuestas se ha identificado previamente utilizando el servicio de identificación. Si el usuario no existe en este servicio se descartará la inserción de los datos en la base de datos.
- **portafolio:** en este directorio estarán las clases que contendrán los métodos para la creación del portafolio (`Portafolio`), edición de las preferencias y envío del listado de

respuestas de un usuario (`PortafolioUsuario`) y la respuesta a la solicitud de una respuesta concreta de un usuario (`PortafolioTarea`). Estas clases heredarán de la clase `ServerResource`. Las respuestas a las peticiones del listado de respuestas o de una respuesta específica se generarán utilizando la plantilla que corresponda del directorio `plantillas`. El contenido de estas plantillas se completará utilizando la extensión `Velocity` a través de los objetos del directorio `helpers`. El código de las plantillas, que se puede visualizar en el Anexo B, contendrá palabras reservadas de la sintaxis de `Apache Velocity`<sup>5</sup> para agregar de manera dinámica la información. En el Listado 5.2 se puede ver un extracto de una parte de la plantilla B.3 donde se itera sobre un vector que contiene la información y dependiendo del contenido de sus objetos realizará una u otra acción. En la Fig. 5.1 se encuentra un ejemplo de páginas del portafolio generadas de manera dinámica.

- **respuestas**: este directorio contendrá las clases con los métodos para la gestión de las peticiones de almacenamiento de una respuesta (`TareasCompletadas`) y de su edición (`TareaCompletada`). Estas clases heredarán de `ServerResource`.
- **servidor**: que contendrá dos clases que estarán directamente relacionadas con la configuración y el enrutamiento de las peticiones que le realicen al servidor. `Componente` heredará de `org.restlet.Component`. Será la clase del servidor que se utilizará como punto de entrada colocando el método `main`. Además, en esta clase se llevará a cabo la configuración del servidor (como puerto elegido, conectores que ayudarán al servidor...). Por otro lado se tendrá la clase `Aplicacion` que heredará de `org.restlet.Application`. En esta clase se establecerán las rutas con las colecciones y recursos permitidos por el servidor. Además, en esta clase se crea la instancia del elemento con el que se llevará a cabo el registro de acciones del servidor.
- **tareas**: directorio donde se encuentra la clase `Tareas` encargada de proporcionar al cliente la información de las tareas de un determinado contexto cuando un cliente realice una petición. Al igual que sucedía con `Contextos`, la clase `Tareas` tendrá un método que creará la solicitud para obtener la información necesaria del punto SPARQL.
- **util**: en este directorio se encuentran clases con métodos utilizados por las clases descritas anteriormente. Se crean con el objetivo de repetir la menor cantidad de código posible. Por ejemplo, en la clase `Auxiliar` se tendrán métodos para calcular la distancia entre dos coordenadas según la aproximación de la fórmula del semiverseno [Rob57], las plantillas para el registro de los eventos que se produzcan en el servidor, la construcción del texto de la licencia de la imagen, el método para comprobar que la información que se va a agregar en la petición de solicitud de contexto es válido (que los valores sean válidos, por ejemplo, que el valor asignado al límite norte no sea superior a  $90^\circ$  y que la distancia entre norte y sur y este y oeste sea inferior a  $0,1^\circ$ ), etc. La clase `ConsultasSparql` contendrá los métodos para la creación de las peticiones que se utilizarán para obtener información de Casual Learn SPARQL.

Las líneas de código del fichero `log4j.properties` servirán para configurar el registro de acciones completadas en el directorio `log`. Este directorio se creará de manera automática al lanzar el servidor si no existiera previamente. Para que el fichero `log4j.properties` tenga efecto se tuvo que agregar al fichero `pom.xml` la dependencia del proyecto `log4j`<sup>6</sup>. Los ficheros de registros de acciones rotan diariamente, es decir, se tiene un fichero por cada día en el que se ha producido alguna acción. En cada entrada del registro se almacena la fecha en el que se produce.

---

<sup>5</sup><https://velocity.apache.org>

<sup>6</sup><https://logging.apache.org/log4j>

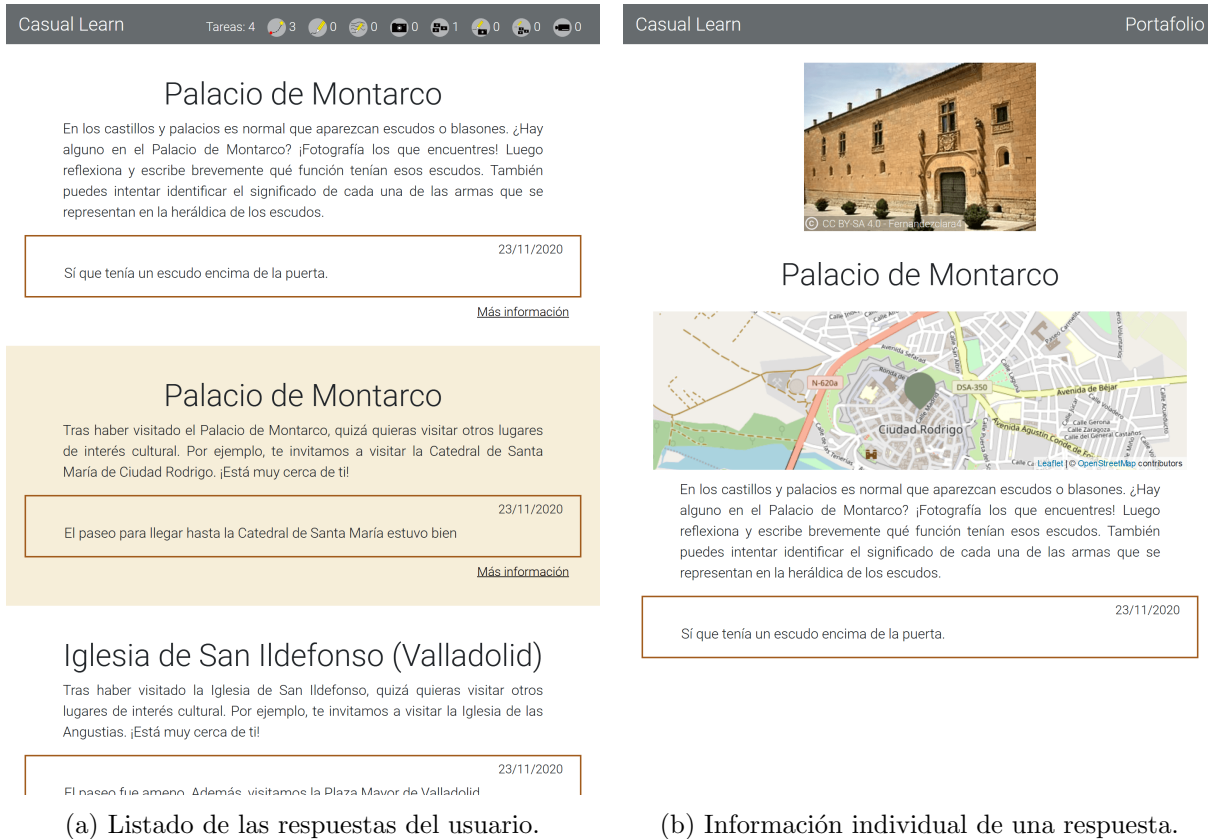


Fig. 5.1: Capturas de pantalla de páginas de un portafolio de ejemplo. Las capturas han sido realizadas con una tableta que dispone de una resolución de pantalla de 768x1024 px.

Actualmente se está registrando la acción de responder a una petición de POI (registrando el tiempo que tardó en procesarse la petición y el número de contextos recuperados), la de respuesta a una petición para obtener la información de las tareas de aprendizaje asociadas a un POI –diferenciando cuando entra en acción el servicio de recomendación (petición de tareas personalizadas para un usuario concreto) y cuando no– registrando el tiempo necesario para procesar la petición y las acciones de creación y edición de una respuesta.

Listado 5.2: Código de ejemplo para mostrar la construcción de bucles y las sentencias if-else en Apache Velocity.

```

1  **
2  Se itera en cada uno de los elementos que compongan
3  al vector "respuestas". En cada "iteración", tareas
4  obtiene el valor de un elemento del vector "respuestas".
5  **
6  #foreach ($tarea in $respuestas)
7  ## Se utiliza una clase u otra para el contenedor de la respuesta.
8  #if ( $foreach.count % 2 != 0 )
9  <div class="sinColor">
10 #else
11 <div class="color">
12 #end
13
14 **
15 Con el nuevo valor de tareas se establece el nombre del lugar
16 donde se realizó la tarea (o monumento) y su enunciado.
17 **

```

```
18     <div class="container contenedorDatos">
19         <h1>${tarea.titulo}</h1>
20         <p>${tarea.textoPregunta}</p>
21     </div>
22
23     ## Resto de acciones para cargar la info. dinámica en la plantilla
24     [...]
25 </div>
26 #end
```

Antes de poder realizar tareas educativas los usuarios de Casual Learn deberán haberse identificado utilizando el servicio de identificación. El servicio externo de identificación de usuarios elegido para esta aplicación es Firebase Authentication [FIRAUT]. Este servicio ofrece distintas alternativas para poder identificar a los usuarios. Se puede hacer que los usuarios se registren en el sistema mediante un nombre de usuario y contraseña específico para la aplicación o utilizando proveedores en los que el usuario pueda tener una cuenta como Google, Facebook, Microsoft, Apple. . . Para la implementación de Casual Learn se ha elegido que los usuarios se identifiquen en el sistema utilizando una cuenta de Google. Esta decisión está motivada por el cliente de Casual Learn, que es una aplicación nativa de Android, que solo se podrá descargar<sup>7</sup> a través de Google Play<sup>8</sup>. Para realizar esta descarga, o cualquier otra desde la tienda Google Play, Google obliga a que el usuario se identifique mediante una cuenta de Google (o cree una nueva). Por este motivo todo usuario que utilice Casual Learn dispondrá de una cuenta de este proveedor. Además, utilizando este proveedor el usuario no tendrá que introducir la contraseña de la cuenta de Google puesto que probablemente mantenga iniciada la sesión para actualizar las aplicaciones que tenga instaladas en su dispositivo. Cuando el usuario inicie sesión en el cliente de Casual Learn se registrará en el proyecto de Firebase que le asignará un identificador único para esta aplicación. Este *token* se utilizará para identificar al usuario en todo el sistema Casual Learn.

### 5.2.1. Persistencia de datos

El servidor tiene que almacenar de forma persistente las respuestas de los usuarios. La forma de guardar esta información variará dependiendo del sistema que se escoja. Existen varias opciones para almacenar este conjunto de datos persistentemente: desde la más simple e insegura, que consistiría en almacenar la información en ficheros de texto plano; hasta la más elaborada, utilizando una base de datos (definida como un conjunto de datos organizados que se guardarán en almacenamiento secundario [Re03] para garantizar su persistencia) junto con un gestor de bases de datos que permita consultar la información almacenada y su gestión.

Para la aplicación Casual Learn se decide utilizar una base de datos para almacenar la información. La primera decisión que se tiene que tomar es qué tipo de base de datos utilizar. Se decidió elegir entre las de tipo SQL (*Structured Query Language*) y las de NoSQL. La principal diferencia entre estos dos modelos, aparte de que en NoSQL no se utiliza el lenguaje SQL para realizar las consultas, es que NoSQL no es tan rígida en lo que se refiere a la estructura de almacenamiento de los datos, por lo que su uso cada vez es más popular en servicios web interactivos donde los datos son cada vez más heterogéneos [MM19]. Además, no se necesitan realizar transacciones (conjunto de operaciones que se tienen que llevar a cabo una después de otra de manera atómica), que sería una de las principales razones para elegir una base de datos relacional [Re03]. Por estos motivos, para el caso de Casual Learn parece más adecuado el uso de

<sup>7</sup><https://play.google.com/store/apps/details?id=es.uva.gsic.adolfinstro>

<sup>8</sup><https://play.google.com/store>

una base de datos NoSQL. Además, durante los estudios realizados en el Grado en Ingeniería de Tecnologías Específicas de Telecomunicación, Mención en Telemática, y en Máster en Ingeniería de Telecomunicación ofertados por la Universidad de Valladolid se trabajó con bases de datos SQL, pero no con las de tipo NoSQL, por lo que se quería conocer cómo se comportaban estas bases de datos. Para el servidor de Casual Learn se optó por utilizar MongoDB<sup>9</sup>, una base de datos basada en documentos. Concretamente se instala la versión 4.4.6 en la máquina virtual donde se ejecutará el servidor de la aplicación.

Dentro de cada base de datos de MongoDB se pueden agregar una o más colecciones. Las colecciones son análogas a las tablas de las bases de datos relacionales [Mon]. Dentro de cada colección los datos se almacenan en documentos. Estos documentos tienen un formato similar a JSON (documentos BSON<sup>10</sup>), que será el tipo de formato en el que el cliente de Casual Learn enviará las respuestas del usuario al servidor. MongoDB se encarga de crear automáticamente tanto la base de datos como la colección cuando se agrega un documento a una base de datos o colección que no exista con anterioridad.

Como cada usuario de Casual Learn tendrá asignado un identificador único (generado por Firebase Authentication) se utilizará este *token* como nombre de la colección propia del usuario. Las respuestas de cada usuario se registrarán dentro de la colección propia del usuario en un documento. Además, se tendrá una colección resumen con documentos que relacionen el *token* de usuario que genera Firebase con el identificador de portafolio que se crea en el propio servidor. En cada documento de la colección de índices también se almacena la visibilidad del portafolio configurada por el usuario. Las comunicaciones con la base de datos se realizarán utilizando el *driver* proporcionado por MongoDB compatible con Java<sup>11</sup>. Esta dependencia se agrega al proyecto utilizando nuevamente el fichero `pom.xml`. En la clase `casualLearnPasarela.persistencia.Persistencia` se encuentran los métodos necesarios para gestionar documentos y conexiones a la base de datos.

Como se indicó en la etapa de diseño, la primera vez que un usuario intente almacenar datos en el servidor de Casual Learn se comprobará si dicho usuario se ha identificado previamente con Firebase Authentication. El servidor, a través del SDK que proporciona Firebase, solicita a Firebase Authentication los datos del usuario. Si el usuario se ha identificado previamente, el servicio de identificación enviará los datos que disponga del usuario, entre los que se encuentra el nombre del usuario y su dirección de correo electrónico. Para intentar preservar la privacidad de los usuarios esta información no se almacena en ningún lugar del servidor de Casual Learn. De esta forma, la única forma de poder identificar al usuario en el sistema será que el propio usuario comparta el identificador de portafolio que puede recuperar desde el cliente de la aplicación.

### 5.2.2. Servicio de recomendación: Recombee

Los sistemas de recomendación generan sugerencias de elementos que pueden ser del interés de un usuario [Fel+14]. Dependiendo en qué se basen los sistemas de recomendación se pueden diferenciar tres tipos [Fel+14]: los basados en **filtrado colaborativo**, que asumen que las personas pueden dejarse influenciar por las opiniones de sus conocidos, por lo que agrupa a los usuarios por intereses similares y las valoraciones que puedan realizar personas de este grupo afectarán a las recomendaciones que reciban el resto de personas del mismo grupo; los que se basan en **filtrado por contenido**, donde se asume que cada usuario tiene unos intereses propios, por lo que sugieren elementos similares a sus intereses; y en último lugar, los basados en

---

<sup>9</sup><https://www.mongodb.com/>

<sup>10</sup><https://docs.mongodb.com/manual/core/document/#std-label-bson-document-format>

<sup>11</sup><https://docs.mongodb.com/drivers/java/sync/current/>

**recomendación por aprendizaje profundo**, que a diferencia de los anteriores se basa en el conocimiento de los productos que se ofrecen en vez de en las valoraciones que realicen los usuarios.

Para el servicio de recomendación de Casual Learn se va a utilizar Recombee<sup>12</sup>. Recombee es capaz de combinar los tres tipos de sistemas de recomendación [Recb] para mejorar las sugerencias ofrecidas a los usuarios. Además, a medida que los usuarios van interactuando con el sistema, Recombee es capaz de adaptarse en tiempo real para proporcionar mejores sugerencias. Es un *software* propietario con varios modelos de suscripción de pago mensual y uno gratuito. Este modelo permite solicitar hasta 100 000 recomendaciones al mes, tener como máximo 20 000 usuarios mensuales, un catálogo de 20 000 elementos y la imposibilidad de tener varias bases de datos. Los planos de pago aumentan las cantidades de recomendaciones, usuarios, elementos en el catálogo y número de bases de datos, además de otras características como permitir las comunicaciones entre el equipo que contrata la suscripción y Recombee. Por el volumen de usuarios que utilizan Casual Learn no se necesita contratar una suscripción superior al plan gratuito.

La configuración inicial de Recombee se realiza desde el servidor de Casual Learn, habiendo realizado previamente la creación de una cuenta en la plataforma Recombee, creado un proyecto y generado un *token* que permita realizar operaciones sobre el proyecto. Esta configuración se reduce a la creación de las **propiedades** de los elementos y al **envío del conjunto de elementos**, que son recuperados de Casual Learn SPARQL utilizando la consulta del Listado 4.3. En el caso de Casual Learn las propiedades que se crean son la del **tipo de respuesta esperada**, **latitud** y **longitud**. Como se ha visto previamente, cada tarea educativa tiene asignado un tipo de respuesta esperada y está relacionada con un contexto que a su vez tendrá asignado una latitud y una longitud. Por este motivo resultaría más sencillo la creación de una propiedad para el identificador del POI donde está agrupada la tarea. El motivo por el cual se hace de esta forma –indicando latitud y longitud– es porque en las primeras versiones de Casual Learn no existía el concepto de POI, por lo que cada tarea educativa tenía asignada unas coordenadas. Para no tener que reiniciar la base de datos, y perder todas las relaciones que Recombee pudiera haber realizado hasta el momento, se mantuvieron las propiedades que existían previamente. Por lo tanto, cada elemento que representa a una tarea de aprendizaje dentro de Recombee estará formado por el identificador de la tarea, la latitud y longitud del POI y el tipo de respuesta esperada a la tarea. Si fuera necesario se realizaría una adaptación para que todos los identificadores de tareas fueran compatibles con los tipos de identificadores considerados como válidos en Recombee: tiene que ser una cadena de texto validada mediante la expresión regular  $\wedge[a-zA-Z0-9_-:\@.]+\$, [Reca]$ . Por este motivo, si el identificador de la tarea contiene algún carácter que no cumpla con esa expresión, se cambia dicho carácter por '@'.

Cuando un usuario valora una tarea desde el cliente de Casual Learn lo hace con una puntuación de entre una y cinco estrellas. Recombee valoraciones con valores comprendidos entre  $-1$  y  $1$ , por lo que se realiza una normalización de valores mediante la expresión 5.1. Una vez realizada esta traducción, a Recombee se le enviará el identificador del usuario que ha valorado la tarea educativa, el identificador de la tarea y la puntuación obtenida. Tanto el identificador del usuario como el de la tarea se adapta a uno permitido por Recombee si fuera necesario.

$$pRecombee = \frac{estrellasUsuario - 3}{2} \quad (5.1)$$

En último lugar, cuando al servidor se le solicita las tareas educativas de un POI recomendadas para un usuario concreto, el servidor se comunica con Recombee solicitando las tareas

---

<sup>12</sup><https://www.recombee.com>



educativas que se encuentren en la posición del POI. Para ello el servidor enviará a Recombee el identificador de usuario (adaptado si fuera necesario), el número máximo de recomendaciones (número de tareas agrupadas en el POI) y la latitud y longitud del POI. Recombee devolverá los identificadores de las tareas en un vector. La posición que tenga cada una de las tareas educativas dentro del vector indicará el grado de idoneidad de la tarea para ese usuario, así que se utilizará ese orden para mostrárselas al usuario. Es por este motivo que Casual Learn está utilizando a Recombee más como un sistema de clasificación de las tareas asociadas a una posición que como un sistema de recomendación. El usuario verá en primer lugar las tareas que le puedan resultar más interesantes. A la hora de comparar el orden sugerido por Recombee con las tareas recuperadas de Casual Learn SPARQL habrá que tener en cuenta que los identificadores pueden ser diferentes debido a la adaptación que se tuvo que realizar para que cumpliera con las reglas del servicio de recomendación utilizado. Por este motivo, a la hora de comparar identificadores para reordenar la información que se le va a enviar al cliente se tiene que adaptar los identificadores de tareas obtenidos del punto SPARQL. Posteriormente, en la información que se envía al cliente, se utiliza el IRI de la tarea educativa indicado en el repositorio de triplas.

### 5.3. Implementación del cliente

El cliente de Casual Learn será la puerta de acceso de los usuarios a la información almacenada en Casual Learn SPARQL. La obtención de esta información será transparente para el usuario. Como se indicó en la fase de diseño, el cliente de Casual Learn dispondrá de dos modos de funcionamiento –búsqueda activa de POI y búsqueda pasiva– en la que el cliente solicitará de manera automática al servidor la información que necesite el usuario en cada momento. La implementación de este cliente se llevará a cabo como una aplicación nativa de Android, compatible con dispositivos que ejecuten desde API 19 (KitKat 4.4<sup>13</sup>) a la 30 (Android 11<sup>14</sup>) de este sistema operativo. Siendo una aplicación compatible con la API 19 se estará dando soporte al 98,1 % de los dispositivos Android según la información que proporciona el entorno de desarrollo Android Studio<sup>15</sup> (18 de septiembre de 2021), siendo del 96,2 % cuando se empezó a estudiar qué alternativas se tenían para implementar el cliente del sistema (18 de marzo de 2020). Como requisitos *hardware*, la aplicación se podrá instalar en dispositivos que puedan obtener su ubicación precisa a través de un sistema de navegación por satélite (configuración indicada en el fichero `AndroidManifest.xml` a través del elemento del Listado 5.3).

Listado 5.3: Se activa la característica por la que solo los dispositivos que puedan utilizar un sistema *hardware* de navegación por satélite sean capaces de ejecutar el cliente de Casual Learn.

```

1 <uses-feature
2   android:name="android.hardware.location.gps"
3   android:required="true"
4 />
```

El cliente se implementará utilizando Java para programar lo parte lógica de la aplicación, y se utilizará anotación XML para la parte visual y describir el resto de recursos y configuraciones. La organización del código de la aplicación se puede ver de manera parcial en el Listado 5.4. A diferencia de cómo se hizo el análisis de la estructura del servidor, en este caso se va a relatar la implementación del cliente a través de las pantallas que se le irían mostrando en cada momento a un usuario que estuviera utilizando la aplicación. Solo se indicará que, a diferencia del servidor,

<sup>13</sup><https://www.android.com/versions/kit-kat-4-4/>

<sup>14</sup><https://www.android.com/android-11/>

<sup>15</sup><https://developer.android.com/studio>

en el caso del cliente se utiliza Gradle<sup>16</sup> en vez de Maven para obtener las dependencias que se vayan a utilizar.

Listado 5.4: Estructura del proyecto del cliente de Casual Learn.

```
1 /app/  
2   src/  
3     androidTest/  
4     main/  
5       java/es/uva/gsic/adolfinstro/  
6         auxiliar/  
7           AdaptadorImagenesCompletadas.java  
8           AdaptadorLista.java  
9           [...]   
10          Auxiliar.java  
11          Bocadoillo.java  
12          ClienteWeb.java  
13          ColaConexiones.java  
14          [...]   
15          persistencia/PersistenciaDatos.java  
16          Acerca.java  
17          Ajustes.java  
18          AjustesFragment.java  
19          AlarmaProceso.java  
20          [...]   
21          Tarea.java  
22     res/  
23       drawable/  
24         [...]   
25       drawable-[...]/  
26       font/  
27         roboto_medium.xml  
28         roboto.xml  
29       layout/  
30         activity_acerca.xml  
31         activity_ajstutes.xml  
32         activity_coompartir_respuesta.xml  
33         [...]   
34       layout-land/  
35         activity_login.xml  
36         activity_maps.xml  
37       menu/  
38         menu_completada.xml  
39         menu_preview.xml  
40         menu.xml  
41       raw/municipios_castilla_y_leon.json  
42       values/  
43         arrays.xml  
44         colors.xml  
45         [...]   
46         strings.xml  
47         styles.xml  
48       values-es/  
49         strings.xml  
50       xml/  
51         file_paths.xml
```

<sup>16</sup><https://gradle.org/>

```

52     network_security_config.xml
53     preferences.xml
54     AndroidManifest.xml
55     ic_launcher-playstore.png
56     test/
57     build.gradle
58     proguard-rules.pro
59 /gradle/
60 /build.gradle
61 [...]

```

### 5.3.1. Identificación del usuario y configuración del portafolio

La primera pantalla que se encontrará un usuario al acceder por primera vez al cliente será la pantalla de identificación (Fig. 5.2a). Esta pantalla adaptará la distribución de sus componentes a si el dispositivo se encuentra en posición vertical o apaisado. Se compondrá por el logotipo de la aplicación Casual Learn, el botón de identificación mediante una cuenta de Google y el botón para acceder a la aplicación en modo invitado. Al pulsar sobre el botón de «Iniciar sesión» de Google se estará haciendo uso del SDK de Firebase para que el usuario pueda identificarse (Fig.5.2b) indicando al usuario qué información estará compartiendo con Casual Learn (aunque luego esta aplicación no haga uso de ella). Si el usuario pulsa sobre el botón de «Acceso sin iniciar sesión» se le mostrará un diálogo indicando que no podrá realizar tareas educativas ni activar el servicio de notificación de puntos de interés. El usuario invitado podrá identificarse con una cuenta de Google en las siguientes pantallas de la aplicación. Si decide no hacerlo, la próxima vez que inicie el cliente de Casual Learn volverá a visualizar la pantalla de identificación.

La siguiente pantalla que verá el usuario será la pantalla de mapas donde podrá visualizar los POI almacenados en Casual Learn SPARQL. Lo primero que se le solicitará al usuario en esta pantalla es conceder los permisos necesarios para obtener su posición. Dependiendo de la versión del sistema operativo se pedirán unos permisos u otros (se obtiene la versión del sistema operativo que el dispositivo está ejecutando y se actúa en consecuencia). En las versiones más modernas de Android se hace una diferenciación entre el permiso para obtener la posición cuando la aplicación está en primer plano y cuando está en segundo plano. Como el servicio de búsqueda de POI en segundo plano es parte del diseño de la aplicación es necesario que el usuario identificado conceda este permiso. En la Fig. 5.2c se muestra el mensaje que se tuvo que añadir para informar al usuario el motivo por el que se le solicitaba el permiso. Indicarle claramente esta información al usuario es necesario para poder publicar la aplicación en Google Play. El último diálogo que se muestra en esta pantalla estará relacionado con la configuración del portafolio (Fig. 5.2d). La visibilidad de las respuestas estarán ocultas por defecto y deberá ser el usuario el que lo active. Además, el diálogo también tiene una opción para que el usuario active el retardo en la publicación de las tareas (opción por defecto) o permita que las respuestas se publiquen en su portafolio inmediatamente después de que el servidor las reciba.

El envío de peticiones al servidor se realiza a través de la biblioteca Volley<sup>17</sup>. Para poder utilizar esta biblioteca lo primero que se debe hacer es agregar su dependencia en el fichero `app/build.gradle`. Además, se deberá crear una clase que gestione la cola de conexiones de la petición. En el caso del cliente de Casual Learn esta clase se puede encontrar en (`es.uva.gsic.adolfinstro.auxiliar.ColaConexiones`) y estará basada en los ejemplos de la documentación [ANDVOL]. Volley permitirá enviar y recibir fácilmente objetos de tipo JSON

<sup>17</sup><https://github.com/google/volley>

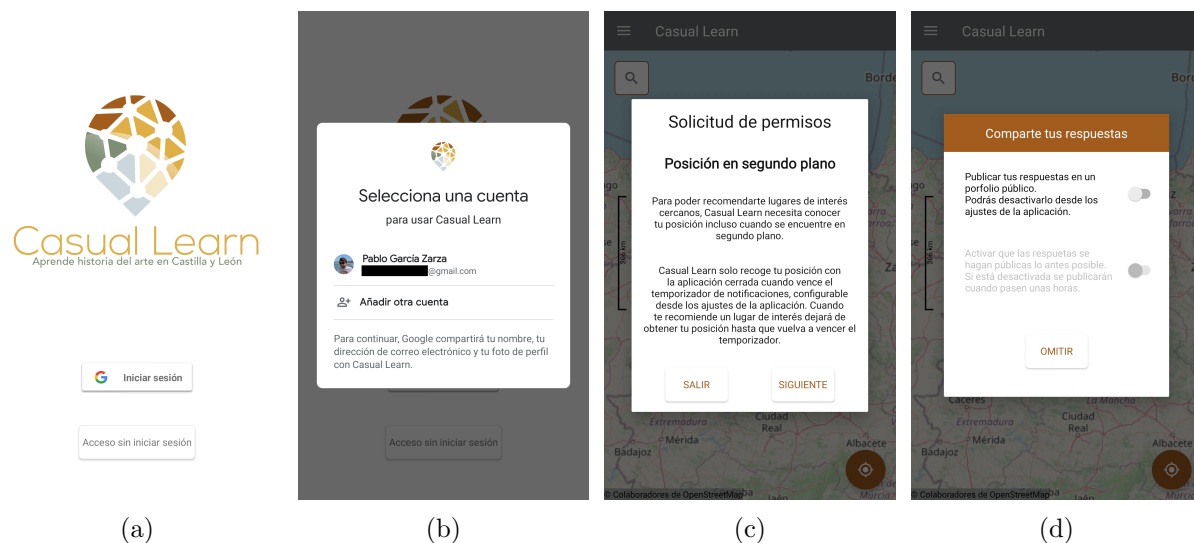


Fig. 5.2: (a) Pantalla de identificación y (b) selección de cuenta. (c) Mensaje explicativo del motivo por el que se solicita el permiso para obtener la posición del usuario en todo momento. (d) Configuración de la visibilidad del portafolio y el retardo en la publicación.

o vectores que contengan objetos de este tipo. Por ejemplo, en el Listado 5.5 se ha indicado un extracto del código utilizado para enviar la configuración del portafolio al servidor.

Listado 5.5: Extracto del código para el envío de la solicitud de creación de un portafolio al servidor.

```

1  [...]
2  JSONObject infoUsuario = new JSONObject();
3  infoUsuario.put("publico", publico);
4  infoUsuario.put("retardado", retardado);
5  infoUsuario.put("idUsuario", idUser);
6  JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(
7      Request.Method.POST,
8      urlServer + "portafolio",
9      infoUsuario,
10     new Response.Listener<JSONObject>() {
11         //El servidor envía el identificador del portafolio del usuario que
12         //se debe almacenar.
13         [...]
14     },
15     new Response.ErrorListener() {
16         @Override
17         public void onErrorResponse(VolleyError error) {
18             //Se gestiona el error indicándoselo al usuario.
19             [...]
20         }
21     });
22 ColaConexiones.getInstance(context).getRequestQueue().add(
23     jsonObjectRequest);
24     [...]

```

### 5.3.2. Búsqueda activa de POI

Con el portafolio configurado se le mostrará al usuario un mapa en el que aparecerá la comunidad autónoma de Castilla y León, España. El mapa estará formado por un conjunto de imágenes proporcionadas por OpenStreetMap [OSM] cuya información se puede utilizar libremente (datos abiertos) siempre y cuando se les proporcione el reconocimiento correspondiente tanto a los colaboradores del proyecto como a OpenStreetMap. Esta información se recupera utilizando la biblioteca `osmdroid` [OSMDR] en su versión 6.1.5. Esta biblioteca, aparte de permitir facilitar la obtención de la información de OpenStreetMap, se utilizará para gestionar las acciones que realice el usuario frente al mapa (aumentar o disminuir el nivel de zoom, desplazamientos por el mapa, centrar la vista en unas coordenadas, etc.). `osmdroid` también está preparada para agregar marcadores y que el usuario pueda interactuar con ellos. Antes de implementar esta opción se realizó una búsqueda para encontrar posibles alternativas que se pudieran utilizar en este proyecto. Ejemplos de las otras bibliotecas encontradas son `Maps`<sup>18</sup> y `mapbox`<sup>19</sup>. Se descartaron estas opciones ya que el mapa de Google es *software* propietario en la que la información del mapa pertenece a Google y `mapbox` porque, aunque se pueda utilizar información proveniente de OpenStreetMap, es un *software* propietario de pago (cobro mensual a partir de las 25 000 usuarios activos).

En la Fig. 5.3a se muestran una serie de marcadores colocados en las posiciones de los POI. En primer lugar, para que el cliente comience a hacer las comprobaciones de si dispone de la información de los POI de la zona del mapa mostrada o tiene que solicitarla al servidor, el usuario deberá haber aumentado el nivel de zoom hasta que la diagonal del mapa mostrado sea inferior a 10km (la distancia de la diagonal del mapa mostrado se obtiene con un método de la biblioteca `osmdroid`). Con ello se evita que se realicen muchas peticiones en muy poco tiempo al servidor, lo que podría provocar problemas si el número de usuarios fuera elevado. Si el nivel de zoom es lo suficientemente alto se calcula el número de teselas verticales y horizontales que se pueden dibujar desde la posición de origen (establecida de manera arbitraria en (41,66247, -4,70605), un punto del edificio de la Escuela Técnica Superior de Ingenieros de Telecomunicación) hasta los límites de la zona mostrada. Este número se calcula utilizando el código del método de la expresión 5.2, donde `latNoLonE` es la latitud más al norte de los dos puntos o la longitud más al este, `latSoLonO` es la latitud al sur o la longitud al oeste e `incremento` será el lado constante de la tesela. Se ha fijado un lado de  $0,0254^\circ$  lo que, aproximadamente, equivale a una tesela de 4km de diagonal utilizando la fórmula del semiverseno [Rob57]. Con estos valores calculados se implementa un bucle que comprueba la disponibilidad de la información de los POI de la tesela. Los datos de los POI de las distintas teselas descargadas se almacenan en ficheros de texto. La información de qué datos contiene cada fichero de texto se tendrá en un fichero de índices (en formato JSONArray). Cada objeto JSON de este fichero de índices especifica los límites de la tesela, el nombre del fichero donde está contenida la información y hasta qué instante la información es válida (información válida por un día desde el momento de su descarga).

$$nCuadrículas = \left\lceil \frac{latNoLonE - latSoLonO}{incremento} \right\rceil \quad (5.2)$$

Con toda la información de las teselas mostradas por pantalla descargada en el dispositivo móvil se pasa a representarla a través de los marcadores. Esta representación se realiza utilizando la biblioteca `osmdroid` y una serie de recursos con la forma del marcador representado en la Fig. 5.3a. Sobre la representación de los marcadores, en su parte central se utiliza un color de

<sup>18</sup><https://developers.google.com/maps/documentation/android-sdk/overview>

<sup>19</sup><https://docs.mapbox.com/android/maps/guides/>

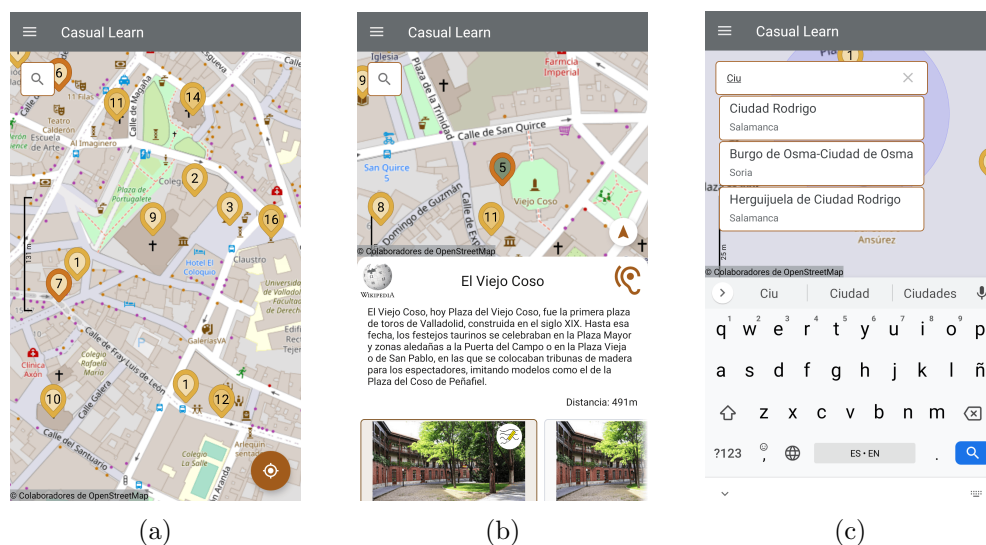


Fig. 5.3: (a) Pantalla que muestra el mapa de la zona consultada por el usuario. Se están mostrando marcadores porque existen POI en la zona que se está consultando y porque el nivel de zum es lo suficientemente alto (se muestra menos región del mapa). (b) Información de un POI con la lista de tareas educativas asociadas a este. (c) Búsqueda de un municipio de Castilla y León para centrar el mapa en su posición.

fondo y un número para indicar el número de tareas educativas que el usuario aún no ha realizado en ese marcador. Si dos marcadores están muy próximos se agruparán en uno para mejorar la usabilidad de la aplicación. Al pulsar un marcador que tenga varios POI agrupados se mostrará un diálogo en el que se solicitará que elija uno de los POI para mostrarle su información.

La información del POI se muestra en la parte inferior de la pantalla (como en la Fig. 5.3b) en la parte final si el dispositivo se encuentra en formato apaisado. En la parte inferior de la tarjeta emergente aparece la lista de tareas de aprendizaje asociadas al contexto. Esta información se solicita al servidor cuando se pulsa sobre el marcador del POI (si no se dispone de la información de estas tareas en la memoria del dispositivo). La forma de almacenar esta información es similar a la utilizada para los POI: en el fichero del POI se almacena el nombre del fichero donde se guarda la información de las tareas educativas y la fecha hasta cuando esta información es válida; en el fichero de las tareas educativas se almacena la información que haya enviado el servidor. Como se puede apreciar la lista de tareas educativas aparece como un conjunto de tarjetas (siguiendo el estilo de la recomendación de Material Design<sup>20</sup>) donde el usuario podrá realizar movimientos laterales para poder visualizar todas ellas. Para ello se ha utilizado `RecyclerView`<sup>21</sup> junto con la creación de un adaptador en el que se establece cómo se debe cargar de manera dinámica el contenido en las tarjetas y qué sucede cuando se pulsa sobre una de ellas. Las tarjetas estarán compuestas por el icono del tipo de respuesta esperado de la tarea educativa, su imagen descriptiva (si la tuviera) y las primeras líneas del enunciado de la tarea. Las imágenes se establecen haciendo uso de la biblioteca `Picasso`<sup>22</sup> que facilita este tipo de operaciones puesto que de las imágenes solamente se dispone de la URL donde se encuentra.

En la parte superior inicial de la pantalla se puede ver un cuadro de texto con una lupa en su interior (Fig. 5.3a). Al pulsar sobre este elemento aparecerá un cuadro de texto donde el usuario pueda introducir el municipio de Castilla y León al que quiera desplazar la vista del mapa. Si el usuario comienza a escribir, el cliente le mostrará posibles sugerencias

<sup>20</sup><https://material.io/components/lists>

<sup>21</sup><https://developer.android.com/guide/topics/ui/layout/recyclerview>

<sup>22</sup><https://square.github.io/picasso/>

relacionadas con el texto introducido (Fig. 5.3c). El orden en el que se muestran estas sugerencias vendrá determinado por la población de los municipios (a más población se le coloca en una posición superior). La información de los municipios se obtiene del fichero de texto `municipios_castilla_y_leon.json` almacenado en el directorio `main/res/raw/`. Esta información se obtuvo del registro de «Municipios de Castilla y León» [DACYL] a través del Portal de Datos Abiertos de la Junta de Castilla y León. La información proporcionada en ese repositorio se redujo para dejar solamente la que fuera necesaria (de cada municipio: nombre, provincia, posición y población) para que ocupara la menor cantidad de espacio posible. Al pulsar sobre una de las sugerencias la pantalla se centra en la posición que se tenga del municipio.

### 5.3.3. Búsqueda pasiva de POI

El cliente de Casual Learn, una vez que el usuario se ha identificado y ha proporcionado los permisos necesarios, buscará POI cercanos a la posición del usuario para sugerirle que realice una nueva visita. El usuario podrá configurar –desde la pantalla de preferencias de la aplicación– el periodo temporal mínimo entre cada notificación e incluso desactivar completamente este servicio si no desea recibir notificaciones de Casual Learn.

Este servicio en segundo plano se ha implementado como si fuera una alarma. La alarma se activa de manera periódica para comprobar si ha vencido el temporizador configurado por el usuario. Los métodos configurados para este servicio se pueden encontrar en la clase `AlarmaProceso` que basa su funcionamiento en la clase `android.app.AlarmManager` y heredará de `BroadcastReceiver` para que se active cada vez que el usuario encienda su dispositivo móvil. Cuando el temporizador de periodo sin notificaciones configurado por el usuario ha vencido el servicio en segundo plano realiza las siguientes comprobaciones:

1. Recupera la posición del usuario utilizando la clase `android.location.LocationManager` y la interfaz `android.location.LocationListener`. A través de ellas se consigue recuperar un objeto de tipo `Location` del que, entre otros datos, se puede obtener una latitud y una longitud. Antes de continuar se comprueba la precisión de la medición puesto que si esta es baja no se podrá asegurar que el usuario se encuentre en la posición obtenida. Se descarta todas las mediciones cuyo error de precisión sea superior a 50m.
2. Se comprueba si se dispone de los POI de la zona donde se encuentra el usuario. Si no se dispone de esta información se crea una tesela en la que el centro es el punto que se acaba de recuperar en el paso anterior y el lado es de  $0,0065^\circ$  (aproximadamente una tesela con una diagonal de  $1km$  mediante la aproximación del semiverseno).
3. Se comprueba la distancia que el usuario se ha desplazado entre las dos últimas comprobaciones. Con los valores fijados actualmente, una vez vencido el temporizador configurable por el usuario, la posición del usuario se recoge cada dos minutos. Si en ese periodo temporal la distancia entre los puntos de la posición actual y la anterior se encuentra a más de  $160m$  (aproximación del semiverseno) se inferirá que no está caminando. Si no está caminando se considera que el usuario no va a ver la notificación (o puede ser peligroso que la vea si, por ejemplo, el usuario está conduciendo).
4. Si el usuario se encuentra caminando, o no se ha desplazado, se recupera el POI más cercano a su posición. En esta comprobación se descartan los lugares que hayan sido notificados en menos de una semana o en los que el usuario ya haya realizado una tarea educativa.
5. Se comprueba que la distancia a la ubicación del POI más cercano es inferior a  $150m$

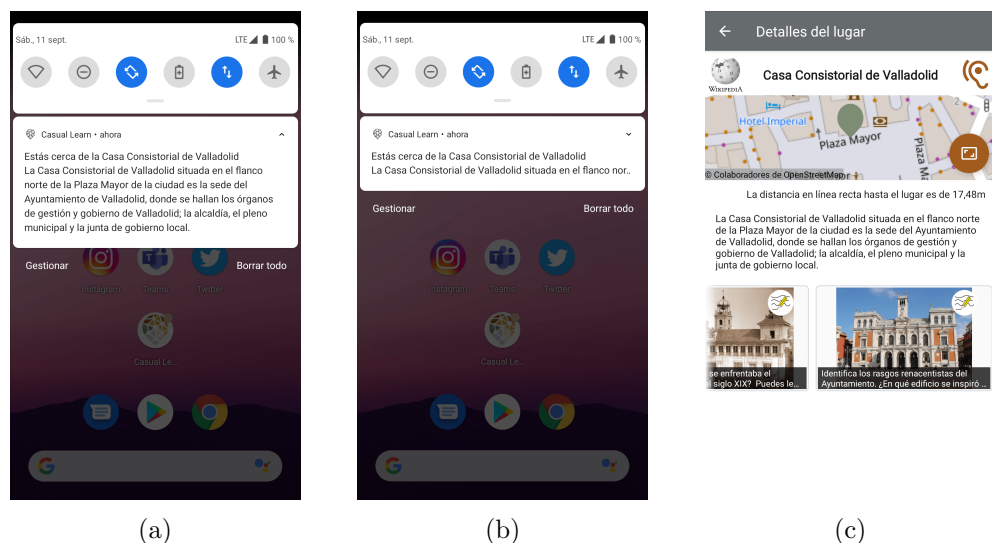


Fig. 5.4: Notificación expandida (a) y compacta (b) de un POI cercano. (c) Información del POI cercano junto con la lista de tareas en el orden personalizado para el usuario.

(aproximación del semiverseno). Si es así se genera la notificación y se le muestra al usuario a través del centro de notificaciones del sistema operativo (Fig. 5.4a y Fig. 5.4b).

Cuando el usuario reciba una notificación de POI cercano y pulse sobre ella accederá a la pantalla mostrada en la Fig. 5.4c. En esta pantalla se muestra la información del POI con un formato similar al de la tarjeta de la búsqueda activa de contextos. Al entrar a esta pantalla se solicitará al servidor las tareas personalizadas para ese usuario y ese momento. La información de estas tareas no se almacenan en la memoria secundaria del dispositivo ya que, por lo que se indicó en la Subsección 5.2.2, Recombee es capaz de adaptar sus sugerencias en tiempo real. Por ello, si el usuario realiza varias tareas se podría dar el caso de que el orden con el que manda Recombee las tareas fuera diferente en cada instante.

### 5.3.4. Realización y edición de una tarea

El usuario puede llegar a la pantalla de información de la tarea (Fig. 5.5a) desde la pantalla donde se muestran los POI en marcadores (subsección 5.3.2) o desde la pantalla con la información del POI cercano notificado (subsección 5.3.3). En cualquiera de los dos casos, en esta pantalla leerá el enunciado de la tarea y decidirá si desea realizarla o no. En el enunciado de la tarea educativa se pueden encontrar términos que pueden ser desconocidos para el usuario. Es por ello que cuando se generaron los enunciados de estas tareas de aprendizaje se embebió en ellos código HTML con un enlace a la descripción de algunos términos. El cliente de Casual Learn es capaz de ocultar al usuario el código HTML embebido y utilizarlo para que, si el usuario pulsa sobre la palabra destacada, abrir un navegador interno a la aplicación donde mostrar su contenido (Fig. 5.3.4). El navegador se genera sobre un diálogo utilizando la clase `ClienteWeb` cuyo código está en el directorio `auxiliar`. Esta clase hereda de `android.webkit.WebViewClient` cargando en el diálogo previamente indicado todas los enlaces que utilicen el protocolo HTTPS enviando al navegador por defecto del usuario (programa externo) el resto de enlaces. En la barra de navegación de la pantalla de información de la tarea se muestra de dos formas el tipo de respuesta esperada para esta tarea: tanto textualmente como con su icono. De esta forma los usuarios aprenderán fácilmente lo que significa cada uno de los iconos de respuesta esperada.



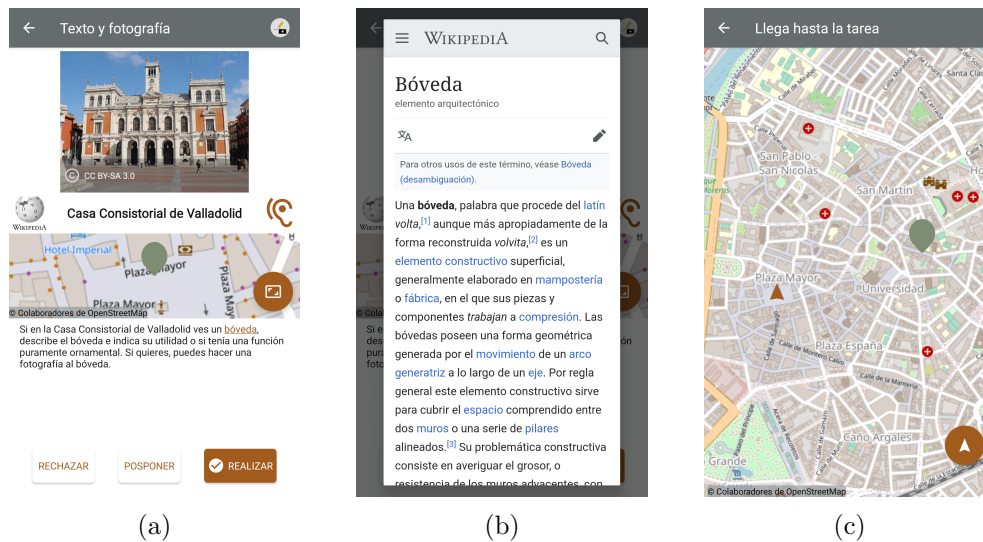


Fig. 5.5: (a) Pantalla con toda la información disponible de la tarea educativa. (b) Navegador interno mostrando la descripción de uno de los términos que aparecen en el enunciado de la tarea. (c) Posición del usuario (representada por un triángulo marrón) frente a la ubicación del POI.

En la descripción de la tarea vuelve a aparecer un mapa donde se muestra la posición del POI. El usuario puede ampliar el mapa pulsando en el botón de la parte inferior final de este. Al realizar esta acción se le muestra, en una nueva pantalla (Fig. 5.5c) su posición frente a la ubicación del mapa. Si pulsa sobre el botón de navegación (parte inferior final) se abrirá, o la aplicación Google Maps si el usuario la tiene instalada o la versión web si no dispone de la aplicación. A esta aplicación (web o nativa) se le enviará mediante un **Intent** la ubicación del POI donde el usuario debe realizar la tarea educativa, además de indicar que quiere que se le guíe hasta ese lugar utilizando de forma predeterminada el transporte público. La posibilidad de elegir el transporte público ha sido el motivo por el cual se ha escogido a Google Maps como servicio de guiado. El cliente de Casual Learn hará la misma operación cuando el usuario pulse en el botón de la parte superior final de la tarjeta con información del POI (Fig. 5.3b).

Si el usuario se encuentra a más de 150m de la ubicación del POI en la parte inferior de la descripción de la tarea educativa se mostrará la distancia en tiempo real a la que se encuentra. Como se puede ver en la Fig. 5.5a, cuando se encuentre a menos de esa distancia se muestran tres botones. Con los botones posponer y rechazar el usuario estará marcando la tarea para que sea mostrada en la lista de tareas pospuestas o rechazadas respectivamente. Estas listas son accesibles desde el menú de la pantalla de mapas. Si el usuario pulsa en realizar se pasará a mostrar una pantalla similar a la de la Fig. 5.6a. Esta pantalla se adaptará al tipo de respuesta esperada de la tarea, tal y como se diseñó en la Subsección 4.4.3. En este caso de ejemplo, el usuario ha escrito un breve texto y ha realizado una fotografía. Antes de realizar esta fotografía el cliente de Casual Learn comprueba que tenga el permiso necesario para utilizar la cámara (solicitándolo si no dispone de él). El usuario podrá eliminar la fotografía o editar el texto antes de finalizar la realización de la tarea.

Si el usuario detecta un problema en el algún dato de la tarea educativa puede reportar la tarea (Fig. 5.6b). La información de este reporte se enviará tanto al sistema de análisis de Firebase (Google Analytics [FIRANA]), como a la dirección de correo de ayuda de la aplicación Casual Learn (si el usuario envía el correo). El envío de información a Google Analytics se realizará a través del SDK proporcionado por Firebase cada vez que se produzca un evento relativo a una tarea: reporte, finalización, edición (de una respuesta) y ponerla en la lista de

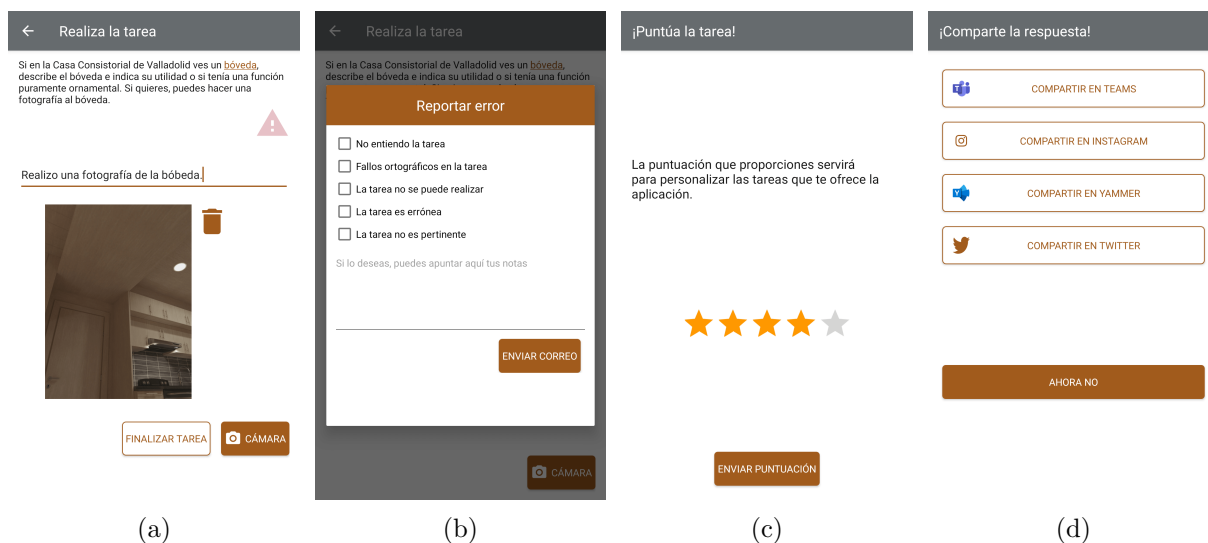


Fig. 5.6: (a) Pantalla de realización de una tarea en la que el estudiante ha contestado a una pregunta y ha realizado una fotografía. (b) Diálogo para reportar algún error en la tarea que esté realizando el estudiante. (c) Pantalla que se le muestra a un estudiante para que valore la tarea educativa que acaba de realizar. (d) Pantalla para compartir la respuesta en diferentes redes sociales y entornos educativos externos a la aplicación.

tareas pospuestas o rechazadas.

Cuando el usuario termine la tarea y pulse sobre el botón **FINALIZAR TAREA** se le dará la opción de valorar la tarea educativa (Fig. 5.6c). Como se ha indicado previamente estas valoraciones se enviarán a Recombee para mejorar el orden de las tareas personalizadas del usuario. En el momento en el que el usuario pulse en el botón de la parte inferior de la pantalla de valoración se intentará enviar la respuesta del usuario junto con la valoración (si hubiera proporcionado una). Antes de realizar este envío se comprueba que se disponga de conexión a Internet. Si no es así, o el usuario ha configurado la aplicación para que solamente envíe las respuestas cuando esté conectado a una red Wi-Fi (como se diseñó en la Subsección 4.4.3), se activará un servicio en segundo plano que comprobará de forma periódica la conexión a Internet y el tipo de red utilizada. Cuando las condiciones sean apropiadas este segundo servicio en segundo plano –que tiene un funcionamiento basado en alarmas como el de búsqueda de POI cercanos– enviará la respuesta del usuario al servidor de Casual Learn.

La siguiente pantalla que se le muestra al usuario es la de compartir la respuesta dada en distintas redes sociales (Twitter, Instagram y Yammer) y entornos educativos (Microsoft Teams) que será similar a la de la Fig. 5.6d. Se decidió dividir la pantalla diseñada con la maqueta de la Fig. 4.5c para dar más peso a la valoración de la respuesta ya que va a utilizarse para mejorar el orden de las tareas del POI notificado. En la pantalla de compartir la respuesta se muestra un botón por cada servicio de terceros. Cada red social o entorno educativo tiene sus peculiaridades (inclusión o no de texto, número máximo de caracteres, número máximo de elementos multimedia. . .) pero el método de comunicación es igual para todas: la información se transfiere a través de un **Intent**. Para compartir la respuesta el usuario necesitará tener instalado el cliente de la red social o entorno educativo en el dispositivo móvil. Si no dispone de él se le muestra un mensaje al usuario cuando pulse sobre el botón de la red social o entorno educativo.

Una vez que el usuario pulse sobre el botón inferior de la pantalla de compartir la respuesta se mostrará la pantalla de mapas con los POI. En el caso en el que todas las tareas de un POI

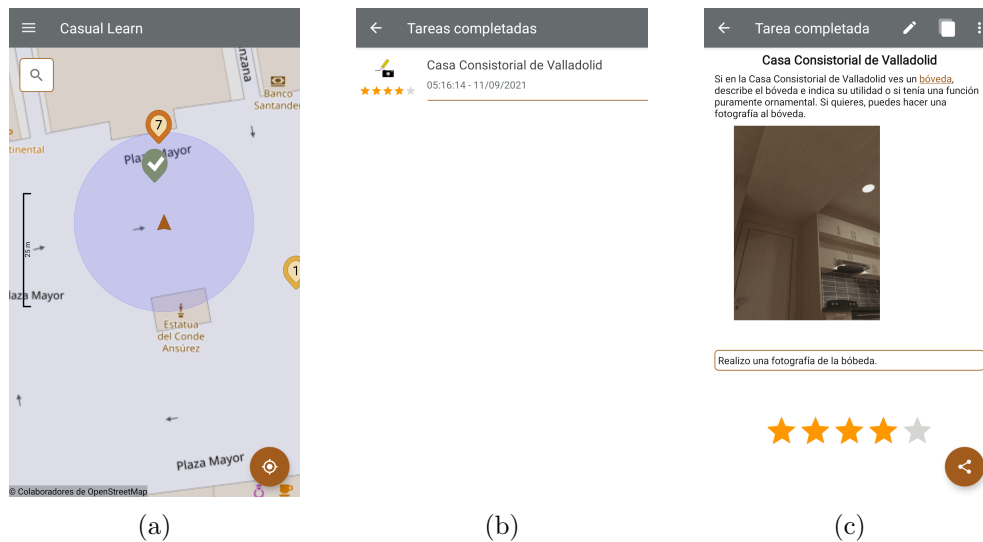


Fig. 5.7: (a) El usuario ha completado todas las tareas educativas de un POI por lo que se modifica su icono. (b) Lista de respuestas que ha generado el usuario. (c) Pantalla para la visualización y edición de la respuesta.

hayan sido realizadas por el usuario, el cliente de Casual Learn cambiará el marcador que se mostraba antes por uno verde con un símbolo de *check* en blanco, como se puede ver en la Fig. 5.7a.

El usuario podrá ver la lista de respuestas que haya proporcionado desde la pantalla de la Fig. 5.7b. En cada elemento de esta lista se indicará el tipo de respuesta esperada de la tarea, la valoración dada por el usuario (si hubiera dado alguna), el nombre del lugar donde se inició la tarea educativa y la fecha en la que se finalizó. Al pulsar sobre una de las respuestas se le abrirá la pantalla de visualización y edición de la respuesta (Fig. 5.7c). En esta pantalla el usuario podrá ver tanto el enunciado de la tarea educativa como la respuesta que haya dado (incluyendo imágenes). Podrá compartir la respuesta dada en las redes sociales y entornos educativos indicados previamente y también obtener la URL con la dirección de la respuesta en el portafolio (si tiene esta característica activa). En referencia al portafolio, desde esta pantalla el usuario podrá cambiar la visibilidad de una respuesta. Por ejemplo, podrá ocultar esta respuesta sin tener que desactivar el resto del portafolio. Por último, desde esta pantalla se permitirá la edición de la respuesta sin que fuera necesario que el usuario se encontrara cerca del POI donde realizó la tarea. El motivo de realizarlo de esta forma es por si el usuario quiere ampliar la respuesta debido a información que ha podido leer después de completarla. Al guardar la respuesta editada se tratará de enviar la respuesta editada al servidor de forma similar a una tarea que se acaba de terminar.

### 5.3.5. Otras pantallas del cliente

En la barra de navegación pantalla donde se muestran los POI sobre un mapa se ha implementado un menú lateral (Fig.5.8a) para poder navegar por el resto de la aplicación. En este menú se muestra el icono de la aplicación, junto con su nombre, seguido de una lista de elementos en los que se especifica un nombre y un icono descriptivo. Pulsando sobre uno de los primeros 4 elementos se accederá a distintas listas (respuestas, tareas completadas, tareas rechazadas y POI notificados (Fig.5.8b)). Las listas relacionadas con las tareas son gestionadas desde la misma clase (`ListaTareas`) con el objetivo de repetir la menor cantidad de código posible.

El *layout* también es el mismo para todas, solamente se cambia el tipo de objeto a cargar y el adaptador utilizado. Al igual que sucede en el resto de listas, se utiliza la clase `RecyclerView` para su implementación.

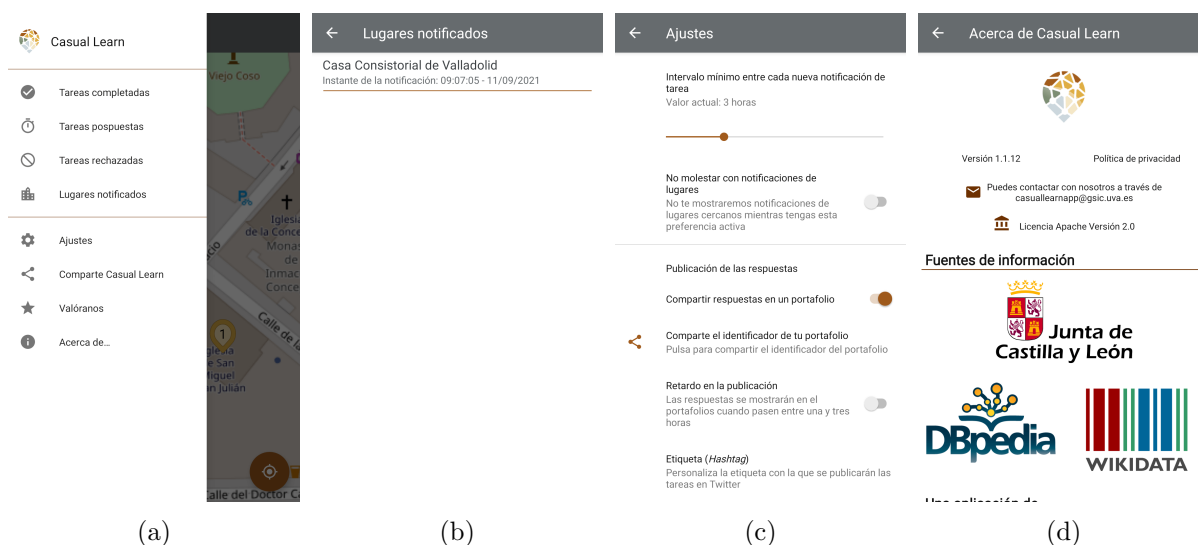


Fig. 5.8: (a) Menú de la aplicación accesible desde la pantalla de búsqueda activa de POI. (b) Lista de ejemplo con los POI que se le han notificado al estudiante. (c) Pantalla para cambiar las preferencias de la aplicación. (d) Pantalla de más información de Casual Learn.

El quinto elemento del menú servirá para acceder a la pantalla de ajustes de la aplicación (Fig. 5.8c). En esta pantalla se podrá configurar el tiempo mínimo entre notificaciones (valor por defecto 3 horas), desactivar completamente el servicio de búsqueda de contextos en segundo plano, configuración la visibilidad del portafolio y compartir (o copiar) la URL del portafolio asignado al usuario. . . Desde esta pantalla el usuario también podrá cerrar su sesión si lo desea. El cierre de sesión se ha pensado por si varios usuarios utilizan el mismo dispositivo móvil (por ejemplo, si un par de estudiantes comparten el móvil porque a uno de ellos se le ha terminado la batería).

El sexto y el séptimo elemento del menú sirven para promocionar Casual Learn compartiendo el enlace de la página de aterrizaje de la aplicación<sup>23</sup> o proporcionando una valoración en la ficha de Casual Learn en Google Play. Pulsando sobre el octavo elemento se accederá a la pantalla de información de la aplicación donde se podrá consultar información como la página con la política de privacidad<sup>24</sup> de Casual Learn, información sobre las fuentes de información de las que se partió para generar los datos contenidos en Casual Learn SPARQL, las bibliotecas utilizadas, etc.

### 5.3.6. Evolución de la interfaz gráfica del cliente

La metodología ágil empleada en el desarrollo de este TFM conlleva asociada una realimentación constante por parte de todas las partes implicadas. La parte del proyecto donde se puede ver con más facilidad la evolución entre iteraciones es la interfaz gráfica. En esta subsección se ha decidido mostrar a través de la Fig. 5.9 algunas de las interfaces diseñadas para la pantalla de búsqueda activa de puntos de interés.

La Fig. 5.9a muestra la primera pantalla de la aplicación: un mapa navegable con un par

<sup>23</sup><https://casuallearnapp.gsic.uva.es>

<sup>24</sup><https://casuallearnapp.gsic.uva.es/privacidad>

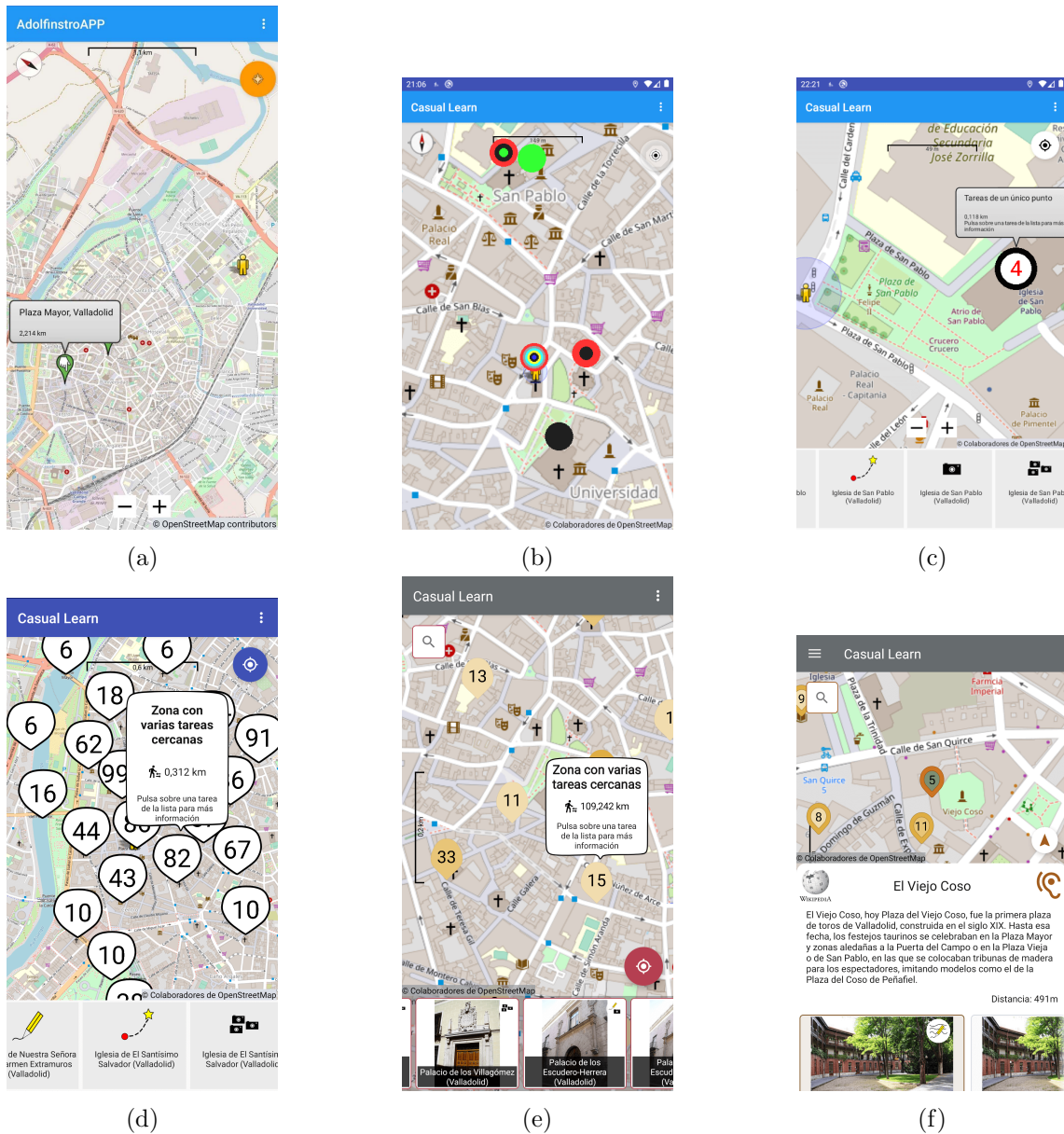


Fig. 5.9: Evolución de la interfaz gráfica de la aplicación.

de marcadores que al pulsarlos mostraban la distancia entre este punto y el usuario. En este momento el servidor todavía no era funcional ni se habían terminado de diseñar las consultas SPARQL. En la Fig. 5.9b y en la Fig. 5.9c se trabajaba con un pequeño conjunto de tareas educativas de forma local. En la Fig. 5.9b los marcadores, en vez de mostrar el número de tareas, indicaban el qué tipo de tareas de determinado tipo de respuesta esperada existían en cada punto. Este sistema fue descartado ya que requería que el usuario tuviera que aprender qué color significaba cada tipo de tarea. A partir de ese momento dentro de los marcadores siempre se indicó el número. En la Fig. 5.9d y en la Fig. 5.9e las tareas educativas ya se obtenían del punto SPARQL a través del servidor. Al pulsar sobre uno de los marcadores se mostraba en la parte inferior la lista de tareas que estaban agrupadas en él.

Hasta este momento no se ha hablado de POI puesto que fue un concepto agregado más tarde: en estas versiones se recuperaban solamente tareas educativas del servidor. Las tareas de aprendizaje tenían asociadas unas coordenadas que utilizaban en el cliente para organizarlas

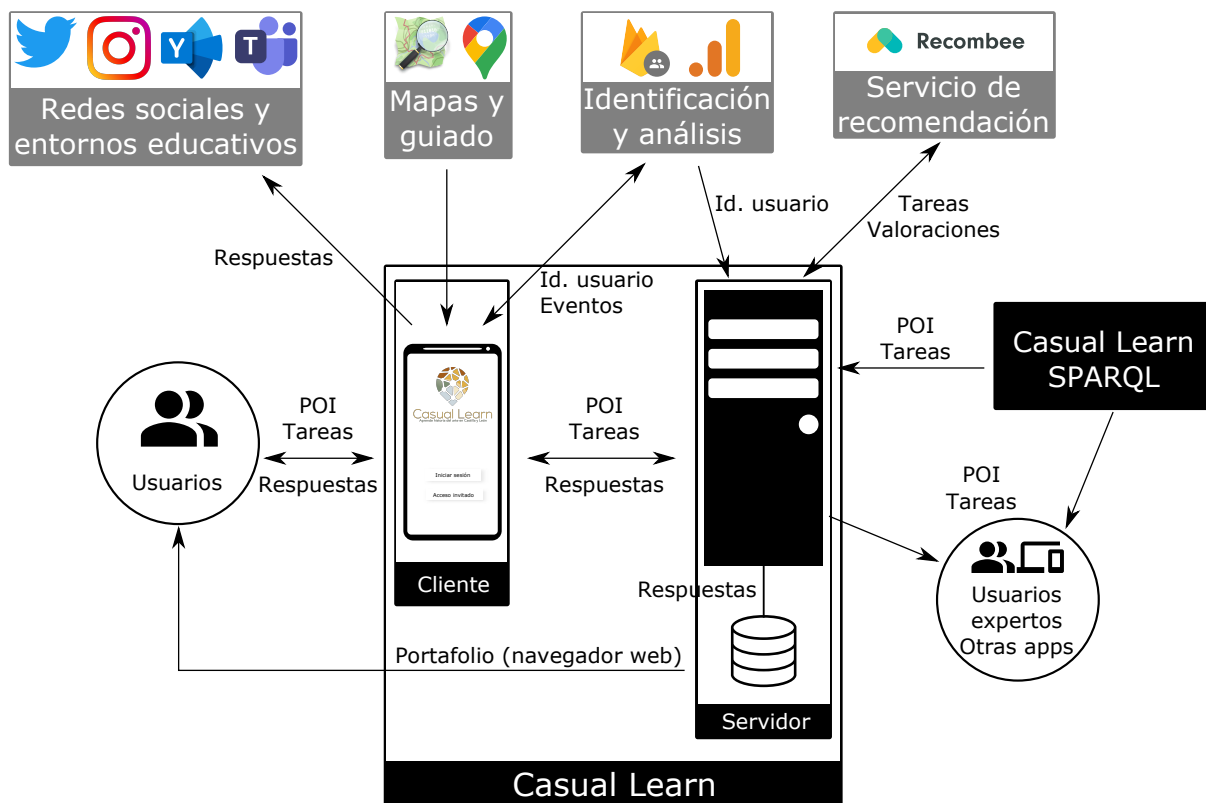


Fig. 5.10: Arquitectura física de la aplicación Casual Learn. Interacción entre las partes desarrolladas y con los servicios de terceros

y agruparlas. Se comprobó que este modo de funcionamiento podía dar problemas en núcleos urbanos donde se concentraran muchas tareas (el móvil tenía que hacer muchas operaciones y el usuario lo percibía ya que ralentizaba el funcionamiento de la aplicación). Este fue uno de los motivos por los que se introdujo el concepto de puntos de interés y se sigue utilizando en la versión actual (Fig. 5.8d).

#### 5.4. Comunicaciones entre los servicios que forman la aplicación

Con la explicación realizada de cómo se ha llevado a cabo tanto la aplicación del cliente y del servidor y de qué bibliotecas externas se ha utilizado en cada caso, se pasa ahora a describir las comunicaciones que existen entre cada uno de los elementos de la arquitectura de Casual Learn. Para comenzar este análisis, en la Fig. 5.10 se indica la arquitectura física de la aplicación, indicando en la parte superior los servicios externos utilizados. Todos los elementos que forman parte de esta arquitectura pasan a explicarse a continuación.

- **Casual Learn:** aplicación implementada en este TFM. Su principal función será la de proporcionar a usuarios no expertos en tecnologías semánticas la información contenida en un repositorio de triplas de forma transparente para ellos.
  - **Cliente:** interfaz gráfica del sistema donde los usuarios podrán visualizar la información que está contenida en Casual Learn SPARQL. Es una aplicación nativa de Android compatible desde la API 19 a la 30. Necesita que el dispositivo tenga el *hardware* necesario para poder posicionarse sin Internet. El cliente permitirá a los usuarios

realizar tareas educativas agrupadas en cada uno de los POI. El cliente almacenará de manera temporal la información obtenida del punto SPARQL, a través del servidor, en el dispositivo para reducir la carga de trabajo al repositorio de triplas del sistema. Su código se encuentra en un repositorio público de GitHub<sup>25</sup> y se puede descargar desde Google Play<sup>26</sup>.

- **Servidor:** expone una serie de recursos para permitir que desde el cliente se pueda solicitar la información de POI y tareas educativas. Tiene otros recursos y colecciones para que el cliente pueda enviar las respuestas dadas por los usuarios a las distintas tareas de aprendizaje. Estas respuestas serán almacenadas en una base de datos local de la máquina donde se ejecute este elemento de la aplicación. Al almacenar las respuestas los usuarios que lo deseen podrán hacerlas públicas para que puedan ser consultadas a través de un portafolio público. El acceso a este portafolio se realizará a través de cualquier navegador web moderno. Se instalará en una máquina virtual dentro de la nube de cómputo privada del grupo de investigación GSIC/EMIC. Necesitará tener instalado el cliente de MongoDB para almacenar los datos de forma permanente y JRE para ejecutar el propio servicio.
- **Casual Learn SPARQL:** repositorio donde está almacenada toda la información de POI y tareas educativas. Se ejecuta en la misma máquina virtual que el servidor por lo que para las comunicaciones se utilizará una de las direcciones de la interfaz de *localhost*.
- **Redes sociales y entornos educativos:** elementos externos donde el usuario podrá publicar las respuestas que proporcione. La aplicación permitirá compartir las respuestas a través de Twitter<sup>27</sup> e Instagram<sup>28</sup> debido a su gran popularidad y Yammer<sup>29</sup> por su utilización dentro de los centros educativos. En cuanto a entornos educativos, los usuarios podrán compartir las respuestas dadas a través de Microsoft Teams<sup>30</sup>. La elección de este entorno viene determinado por su implantación en los centros educativos de educación secundaria de Castilla y León. Se necesita que el usuario tenga instalado los clientes de la red donde quiera compartir.
- **Mapas y guiado:** el cliente del sistema proporcionará a los usuarios una visión de dónde se encuentran los POI con respecto a su posición a través de una interfaz compuesta por un mapa. Se utilizarán datos abiertos de OpenStreetMap para la representación de estos mapas. También se recurrirá a Google Maps para proporcionar al usuarios distintas alternativas para llegar desde su posición a la del POI.
- **Identificación y análisis:** desde el cliente del sistema los usuarios podrán visualizar la información almacenada en Casual Learn SPARQL sin necesidad de identificarse. Sí que será necesaria esta identificación cuando el usuario desee realizar una de las tareas educativas. Para realizar esta identificación se hará uso de Firebase Authentication. También se recurrirá a Google Analytics –también de la plataforma Firebase– para registrar la realización de distintas acciones dentro del cliente de Casual Learn.
- **Servicio de recomendación:** cuando los usuarios completen tareas de aprendizaje podrán valorarlas si lo desean. Con estas valoraciones, y utilizando Recombee, se le podrá ofrecer al usuario las tareas educativas en el orden en el que sean de su agrado. Recombee

<sup>25</sup><https://github.com/gsic-emic/casual-learn>

<sup>26</sup><https://play.google.com/store/apps/details?id=es.uva.gsic.adolfinstro>

<sup>27</sup><https://twitter.com/>

<sup>28</sup><https://www.instagram.com>

<sup>29</sup><https://www.microsoft.com/es-es/microsoft-365/yammer/yammer-overview>

<sup>30</sup><https://www.microsoft.com/es-es/microsoft-teams/group-chat-software>

necesitará conocer previamente qué tareas educativas están almacenadas en Casual Learn SPARQL.

- **Usuarios, usuarios expertos y otras aplicaciones:** la mayoría de usuarios utilizarán Casual Learn instalando en su dispositivo móvil el cliente de la aplicación. Estos usuarios obtendrán la información del repositorio de triplas de forma transparente. En otro lugar se pueden encontrar a usuarios expertos que obtengan esta información realizando peticiones a los recursos expuestos por el servidor de Casual Learn o directamente sobre el punto SPARQL. De la misma forma, se podrán diseñar otros clientes que obtengan la información de Casual Learn SPARQL a través del servidor de la aplicación en vez de directamente. Existe esta posibilidad, ya que mientras que para utilizar Casual Learn SPARQL se necesita tener conocimientos de tecnologías relacionadas con la Web Semántica, para obtener la misma información del servidor solo se necesita realizar operaciones GET estándar.

En la Fig. 5.11a se indican los pasos que se realizan para que el usuario termine por visualizar la información de una tarea educativa o un POI. En primer lugar, el usuario navega por el mapa navegable o pulsa sobre uno de los marcadores del mapa (1). Esta acción conlleva una reacción en forma de solicitud (2), del cliente al servidor, para obtener los POI de una tesela (indicando sus límites) o las tareas asociadas a un POI (indicando su identificador). El servidor se comunica con Casual Learn SPARQL (3) solicitando la información que demande el cliente y procesándola para adaptarla a un formato que el cliente soporte. El servidor responde al cliente (4) con la información obtenida del punto SPARQL y registra en su *log* la acción que acaba de realizar. El cliente muestra al usuario (5) la información obtenida.

La Fig. 5.11b representa a un usuario que pulsa (1) en una notificación de POI de interés cercano. El cliente solicita al servidor (2) las tareas de aprendizaje del POI cercano en el orden adecuado para el usuario (envía el identificador de POI y de usuario). El servidor realiza una consulta al punto SPARQL (3), procesa la información y solicita a Recombee (4) las tareas educativas cuyas coordenadas coincidan con las del POI para el usuario que ha iniciado la acción. El servidor ordena y responde al cliente (5) las tareas según le ha indicado Recombee y el cliente se las muestra al usuario (6) a través de una lista.

En la Fig. 5.11c un usuario activo ha completado una tarea y la ha valorado (1). Al pasar de pantalla se envía a Google Analytics (2a) el evento de tarea finalizada para que se registre en las estadísticas. El cliente también envía al servidor (2b) la respuesta junto con la tarea. El usuario todavía no había registrado ninguna respuesta, por lo que el servidor comprueba mediante Firebase Authentication (3) si el usuario es un usuario identificado del sistema. Si es un usuario legítimo se almacena en la base de datos la respuesta (4a) y se envía la valoración a Recombee (4b) indicando el identificador de usuario, de tarea y la valoración dada por el usuario. El servidor responde al cliente con el identificador de la respuesta en el portafolio (5). Al usuario se le muestra la pantalla de compartir una respuesta (6) y decide enviarla a Twitter mediante su aplicación por lo que el cliente de Casual Learn adapta la respuesta al formato demandado por la plataforma y le envía la información (7).

En la Fig. 5.11d se representan dos acciones. En la azul el usuario accede por primera vez al cliente de Casual Learn (1) y pulsa en el botón de iniciar sesión con la cuenta de Google. El cliente, a través del SDK de Firebase, envía a Firebase Authentication (2) la solicitud del usuario y este le contesta con el identificador de usuario para esta aplicación y otra información como su nombre de usuario. El cliente de Casual Learn le muestra al usuario (3) la siguiente pantalla. En cuanto a la acción negra de la Fig. 5.11d, un usuario quiere acceder a uno de los portafolios existentes en el sistema. Por este motivo envía al servidor (1) una petición para obtener la página individual de una respuesta. El servidor intenta recuperar de la base de datos (2) la respuesta



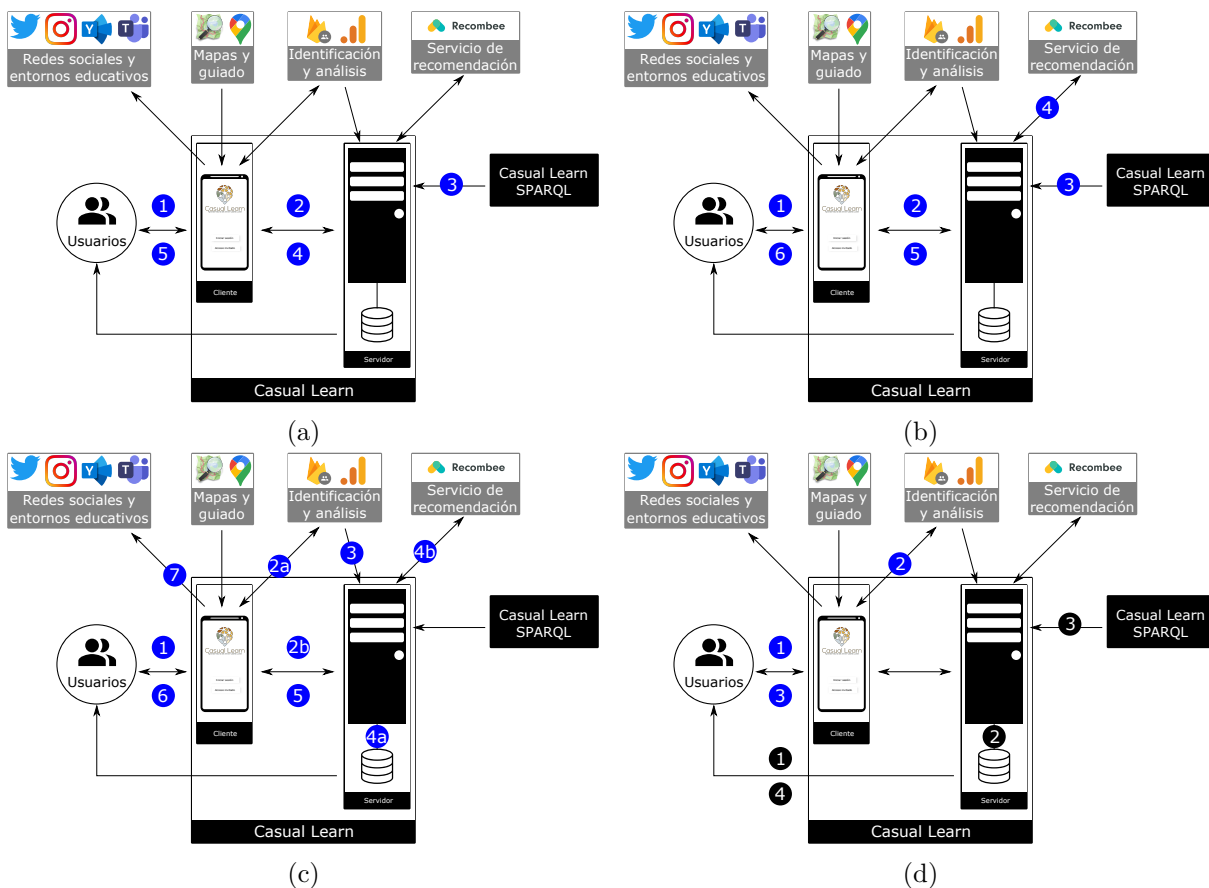


Fig. 5.11: Interacciones entre las distintas partes que forman Casual Learn. (a) Obtención de POI o tareas educativas. (b) Obtención de tareas personalizadas asociadas a un POI. (c) Un usuario completa por primera vez una tarea. (d) Identificación de un usuario (azul) y obtención del portafolio de un usuario (negro).

solicitada. Si existe recoge de Casual Learn SPARQL (3) los datos que no se almacenan en la respuesta. El servidor combina la información del punto SPARQL con la respuesta del usuario obtenida de la base de datos en una plantilla que es la respuesta (4) que se envía al usuario.

## 5.5. Conclusiones

La utilización de servicios externos a Casual Learn para ofrecer algunas características como el que ofrece la identificación de usuarios de Firebase Authentication o el de recomendación de Recombee ha permitido que la implementación se pueda realizar de manera más rápida que si se hubieran tenido que desarrollar soluciones propias para cumplir con las tareas que realizan estos servicios. Además, específicamente hablando del servicio de identificación de usuarios y el de registro de eventos proporcionados por Firebase, utilizar este tipo de servicios permite aumentar la seguridad global de la aplicación: si en el tiempo que se destinó a la implementación de la identificación de usuarios a través del SDK de Firebase (unas pocas horas) se hubiera tenido que implementar una solución propia con este mismo propósito es probable que se hubieran cometido errores que podrían haber supuesto fallos de seguridad que implicarían datos sensibles de usuarios. Y aunque no se hubieran cometido errores de seguridad sería complicado conseguir una solución propia tan robusta como los servicios que ofrece Firebase en tan poco tiempo. Como clara desventaja de utilizar *software* propietario es que en cualquier momento un servicio

que antes era gratuito puede dejar de serlo, dejarlo sin soporte o directamente eliminarlo. Por este motivo parece que aunque a corto plazo, con el fin de tener una versión del producto finalizada lo antes posible, sea interesante utilizar servicios de terceras partes –incluso aunque se utilice *software* propietario– en el largo plazo se debería desarrollar soluciones propias o utilizar soluciones de terceros que garanticen un soporte durante largos periodos de tiempo.

Por otro lado, la implementación del servidor web que utiliza la aplicación Casual Learn puede que hubiera sido más interesante desarrollarla utilizando tecnologías nativas de la Web. Esto es debido a que se está continuamente trabajando con objetos JSON (*JavaScript Object Notation*). Si se hubiera desarrollado el código en JavaScript (lenguaje más popular en el año 2020 según [Git]) para utilizarse sobre el entorno de ejecución de Node.js<sup>31</sup> se podrían haber utilizado las mismas bibliotecas sustituyendo Maven por npm<sup>32</sup> para obtener las dependencias. A cambio de tener estas facilidades se tendría que haber trabajado con lenguajes diferentes en la parte servidor y en el cliente, ya que este se ha desarrollado utilizando Java.

---

<sup>31</sup><https://nodejs.org>

<sup>32</sup><https://www.npmjs.com>

## Capítulo 6

# Evaluación del sistema

En este capítulo se presentan los resultados obtenidos durante los meses en los que el sistema ha estado en producción. Por lo tanto, son datos generados por usuarios reales. De los ficheros de registro del servidor de Casual Learn se han obtenido los tiempos promedio que el servidor necesita para resolver las solicitudes de POI de una zona, tareas agrupadas en un POI y tareas agrupadas en un POI ordenadas para un usuario. Destaca negativamente el tiempo promedio del último tipo de petición ya que se necesita más de 0,5s para procesarlas. Los tiempos promedios de las otras dos peticiones son lo suficientemente bajos para ser utilizados por una aplicación interactiva. Gracias a los eventos que se registraban en Google Analytics también se reportan datos como el número de tareas que han realizado los usuarios, el número de usuarios, número de adquisiciones, etc.

### 6.1. Introducción

La última etapa de la metodología ágil utilizada en el desarrollo de este TFM es la de evaluación de la versión del producto que se desea presentar al cliente [SS05]. Una vez finalizada esta fase se debería decidir si el producto es lo suficientemente maduro como para presentarlo al cliente o se debe volver a fases anteriores de la metodología para solventar los problemas encontrados.

En este capítulo se presenta la evaluación de la aplicación de Casual Learn. La evaluación se lleva a cabo con datos generados en producción, con usuarios reales, ya que la aplicación pasó a poder ser usada por cualquier persona que lo deseara desde el 4 de septiembre de 2020. Por este motivo se utilizan las métricas recogidas en el registro del servidor y en los servicios externos para comprobar el estado del sistema en vez de generar pruebas sintéticas. En primer lugar, se lleva a cabo un análisis del rendimiento del servidor a través de los tiempos de respuesta de las peticiones que realizaron los usuarios. En segundo lugar, se indican los datos obtenidos en el cliente a través de los eventos recogidos mediante el servicio de análisis de Firebase. También se explica cómo se desarrollaron las pruebas piloto que han utilizado Casual Learn para llevar a cabo distintas actividades educativas.

### 6.2. Rendimiento del servidor

El servidor de Casual Learn se ejecuta en una máquina virtual dentro de la infraestructura de nube privada del GSIC/EMIC. En esta máquina virtual se están ejecutando otros proyectos

como el gestor del punto SPARQL (Virtuoso Open Source Edition<sup>1</sup> en su versión 7.2.5). Que el repositorio de triplas se encuentre en la misma máquina donde se ejecuta el servidor de Casual Learn es relevante porque se utiliza la interfaz de *loopback* para las comunicaciones entre ellos. Por este motivo, los tiempos que se expongan en esta sección –cuando solo intervengan el servidor y Casual Learn SPARQL– serán íntegramente de procesamiento de información, ya que los tiempos de comunicación entre los dos servicios se han reducido al mínimo. Los datos reflejados en esta sección no tienen en cuenta los tiempos de latencia que puedan sufrir las comunicaciones entre el servidor y el cliente, este segundo se estará ejecutando en el dispositivo móvil del usuario. El tiempo desde que el servidor recibe cada petición hasta que proporciona una respuesta se comenzaron a almacenar en su registro el 23 de noviembre de 2020, así que los datos que se muestran en este documento parten de esa fecha hasta el 11 de septiembre de 2021.

Tabla 6.1: Resumen de las métricas analizadas en el servidor de Casual Learn, siendo: *POI* la solicitud de los POI dentro de una tesela; *Tareas* la solicitud de las tareas asociadas a un POI; *TareasP* la solicitud de las tareas personalizadas de un POI para un usuario (interviene el servicio de recomendación).

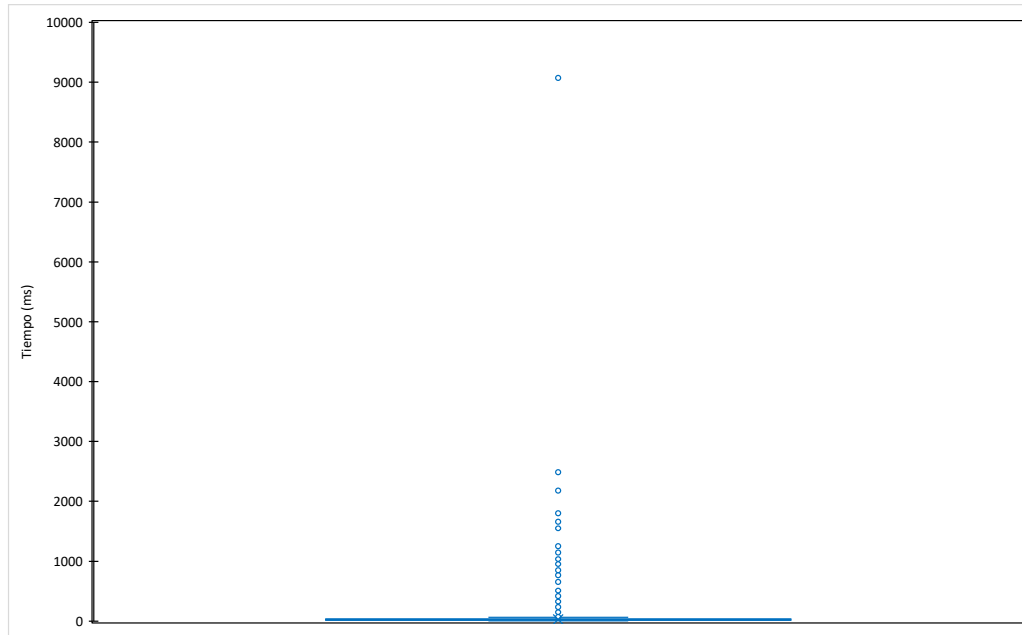
Solicitud realizada	Número peticiones	Mínimo (ms)	Máximo (ms)	Promedio (ms)	Desviac. típ.(ms)
POI	55 947	6	9074	28	50
Tareas	16 100	7	16 396	78	322
TareasP	131	344	979	613	125

En la Tabla 6.1 se pueden ver los resultados que se han obtenido al analizar la información almacenada en los ficheros de *log* del servidor. Como recordatorio, el servidor de Casual Learn genera un fichero de manera diaria para almacenar algunas de las acciones que se producen en él. De estos ficheros de registro se puede extraer los tiempos que ha necesitado el servidor para atender a peticiones de obtención de los POI de una zona y de las tareas que pertenecen a un POI (directamente o solicitando al servicio de recomendación que personalice el orden de las tareas del POI para un usuario específico de la aplicación). En el servidor también se registra el instante en el que se está solicitando una de las páginas del portafolio, pero no se está almacenando el tiempo que necesita el servidor para responder al cliente. Sucede lo mismo con las respuestas. Por este motivo no se reportarán estos dos valores en esta sección. Además, el número de respuestas que han realizado los clientes se presentará en la sección 6.3 a través de los eventos registrados en Firebase.

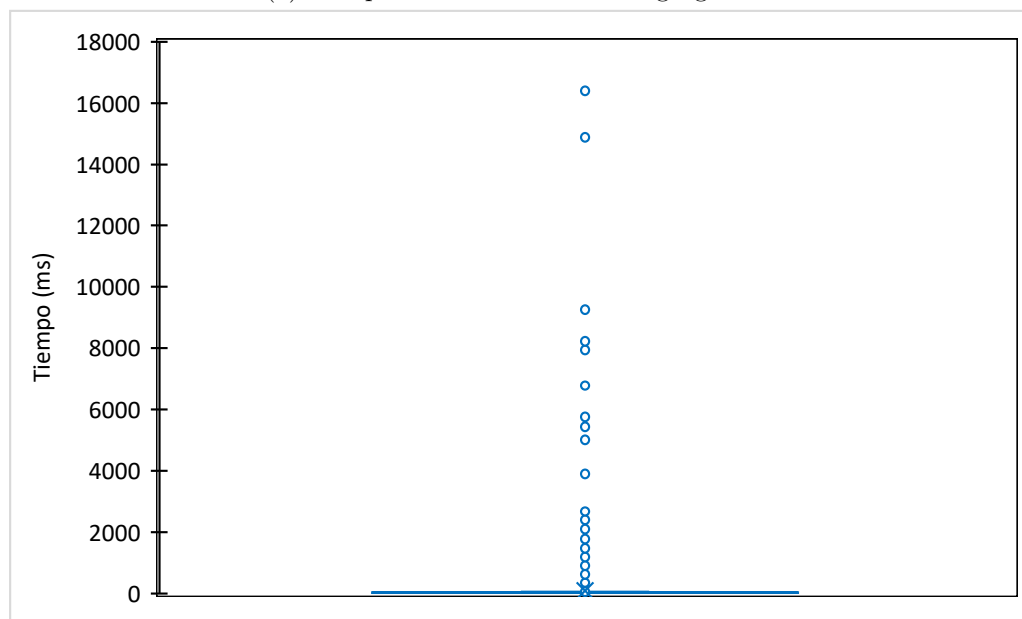
Comparando los resultados de la Tabla 6.1 se observa cómo el tiempo promedio de recuperación de los POI de una zona es más bajo que el que se necesita para obtener las tareas asociadas a un POI (el promedio de recuperación de tareas es 2,8 veces más lento que el de obtención de los POI de una zona). Esta situación es esperable, y está justificada por el comentario realizado en la subsección 4.3.2: en la consulta de las tareas asociadas a un POI se solicitaba más información para enriquecerlas que las que se realizan para la obtención de los POI de una zona. Además, en las consultas SPARQL de las tareas educativas se utiliza más veces la palabra reservada *OPTIONAL*, que añade más complejidad al trabajo que tiene que realizar el motor SPARQL [DuC13]. Con los tiempos mínimos y máximos indicados en la Tabla 6.1 se expresa el valor máximo y mínimo de cada uno de los conjuntos de datos. Los tiempos mínimos tan bajos de estas dos solicitudes se deberán a que dos clientes han realizado la misma petición en un corto periodo de tiempo por lo que punto SPARQL tenía almacenada la respuesta en memoria caché. Los máximos se produjeron el día 15 de julio del año 2021 y posiblemente estuvieran provocados por algún pico de carga en la nube de cómputo. Los tiempos promedio para obtener los POI y las tareas educativas son lo suficientemente bajos como para poder trabajar con ellos

<sup>1</sup><https://github.com/openlink/virtuoso-opensource>

en aplicaciones interactivas.



(a) POI pertenecientes a un área geográfica.



(b) Tareas asociadas a un POI.

Fig. 6.1: Representación de la consistencia del servidor a distintas peticiones.

En la Fig. 6.1a y en la Fig. 6.1b se han representado los diagramas de caja (*boxplots*) de los conjuntos de datos de los tiempos de procesamiento de las solicitudes para obtener los POI de una tesela y las tareas de un POI respectivamente. Con un diagrama de caja se puede observar a simple vista cómo es la consistencia de un conjunto de datos. En estas dos figuras se observa que la mayoría de las solicitudes se agrupan en tiempos próximos al promedio. Aún así ambos gráficos presentan *outliers* (o valores atípicos) que en algunos pueden llegar a ser elevados (el usuario apreciaría que los marcadores han tardado más de lo que habitualmente tardan en mostrarse o las tarjetas de tareas del POI han necesitado más tiempo del normal para aparecer).

En las solicitudes que realiza el cliente para obtener la información de las tareas de un determinado POI para un determinado usuario (solicitud de tareas personalizadas) además de intervenir Casual Learn SPARQL se establece una comunicación entre el servidor de Casual Learn y Recombee. Por este motivo, se puede ver en la Tabla 6.1 que el tiempo promedio en este caso es más elevado que en los anteriores (7,9 veces más lento en promedio que la solicitud de tareas de un POI sin personalizar). Este es uno de los motivos por el que se decidió restringir la utilización de la personalización de las tareas a los POI cercanos a la posición del usuario (al que se le avisa a través de una notificación del sistema).

El diagrama de caja de este caso (Fig. 6.2) permite apreciar mejor la forma de un *boxplot*. En este tipo de gráficos se representan mediante un rectángulo el primer percentil (parte inferior del rectángulo), el segundo (línea horizontal dentro del rectángulo, también denominada mediana) y el tercero (parte superior del rectángulo). Además, el aspa representa el valor promedio. Volviendo a la Fig. 6.2 se aprecia que en el caso de esta solicitud los *outliers* se reducen a un único valor (que se produjo el 4 de abril de 2021), aunque también es cierto que el número de muestras de este caso es más bajo que en los casos anteriores.

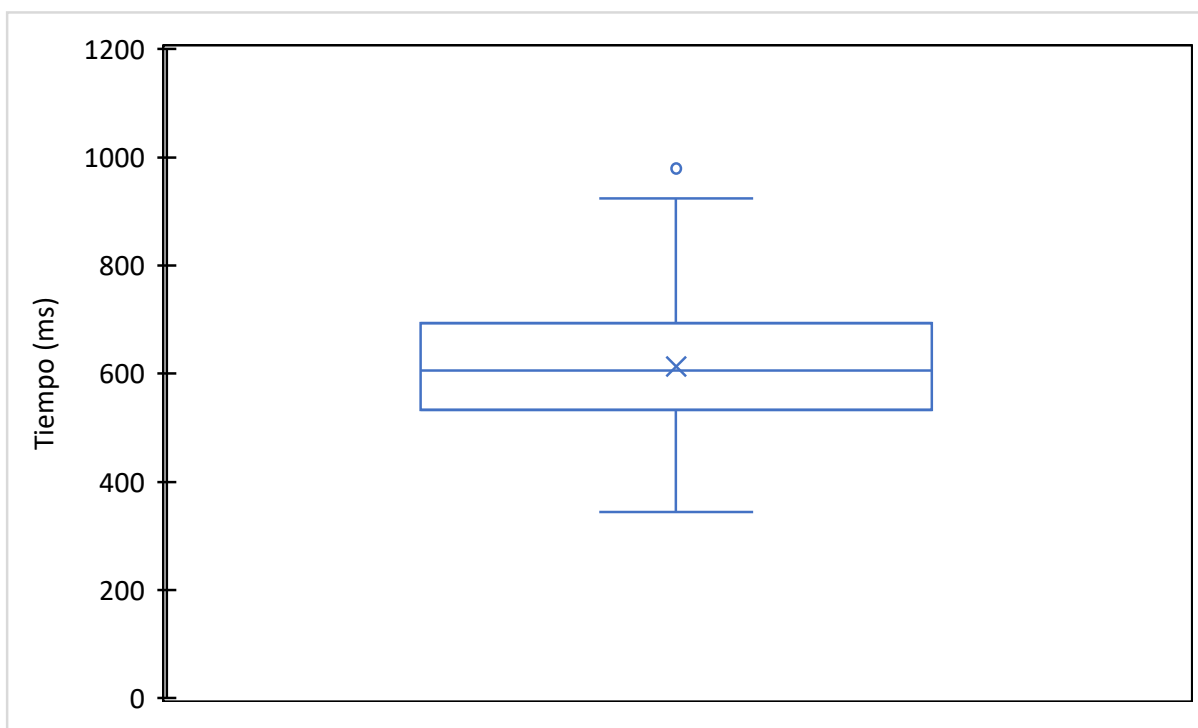


Fig. 6.2: Representación de la consistencia del servidor a las peticiones de solicitud de las tareas asociadas a un POI y personalizadas para un usuario.

### 6.3. Evaluación de la aplicación

Como recordatorio, el cliente de Casual Learn es una aplicación que debe permitir a los usuarios utilizar la información almacenada en Casual Learn SPARQL de manera transparente proporcionando una interfaz de usuario. Además, debe de notificar a los usuarios cuando se encuentren próximos a POI cercanos y adaptar su interfaz para que los usuarios que lo deseen puedan resolver las tareas de aprendizaje que están agrupadas en POI. Todos estos requisitos se han cumplido mediante la solución implementada y se ha podido comprobar con usuarios reales. Los usuarios que lo deseen pueden descargar el cliente de Casual Learn en sus teléfonos Android

desde la tienda Google Play<sup>2</sup>. Como se indicó en el capítulo 5, los requisitos para poder instalar este cliente en un teléfono móvil son: disponer de una versión del sistema operativo Android que ejecute desde la API 19 a la 30 (ambas incluidas) y que el dispositivo cuente con el *hardware* necesario para poder posicionar al usuario a través de un sistema de navegación por satélite, comprobaciones que llevará a cabo Google Play de manera transparente. Por este motivo el cliente solo aparecerá disponible para instalar en los dispositivos que cumplan los requisitos. El cliente de Casual Learn está disponible en Google Play desde el 4 de septiembre del 2020, por lo que se dispone de datos de usuarios reales desde esa fecha.

Tabla 6.2: Resumen de usuarios que han utilizado el cliente.

Número de adquisiciones en Google Play	555
Número máx. de usuarios con la aplicación instalada	267
Usuarios identificados	473
Tiempo medio de interacción por usuario	15min 37seg

La publicación de la aplicación se llevó a cabo desde Google Play Console<sup>3</sup>. El número total de adquisiciones desde Google Play (del 4/9/20 al 12/9/21) es de 555. La mayor parte de los usuarios que han adquirido la aplicación (336) acceden a la ficha de Google Play realizando una búsqueda en Google Play por nombre. La segunda fuente de usuarios son las referencias externas (216), que podrían ser accesos desde artículos de prensa hasta accesos desde la página de aterrizaje<sup>4</sup>. La última fuente de usuarios por la que algunos usuarios han instalado la aplicación es por encontrarse con ella al navegar por los distintos menús de Google Play (3 usuarios).

En Google Play Console, aparte de poder realizar tareas relacionadas con la publicación y gestión de la aplicación (como pudiera ser la configuración de la ficha que se le mostrará al usuario en Google Play o proporcionar respuesta a las reseñas que escriban los usuarios en la mencionada ficha), se pueden consultar una serie de estadísticas (en la Tabla 6.2 se encuentra un resumen de estas). Con la información proporcionada en esta página se ha generado el gráfico de la Fig. 6.3. Este gráfico muestra el número de adquisiciones desde la ficha de Google Play (un usuario que instale la aplicación en varios dispositivos contará como una única adquisición), desinstalaciones y el número diario de usuarios que mantienen la aplicación en su dispositivo móvil. Analizando este gráfico, se puede ver como el número máximo de usuarios con la aplicación instalada se produjo el 4 de abril –siete meses después de su publicación– consiguiendo estar instalada en 267 dispositivos. Este valor máximo coincide con el desarrollo del piloto formado por un número mayor de estudiantes. Se observa cómo después de la finalización de cada piloto el número de dispositivos con la aplicación instalada disminuye, teniendo el número más alto de desinstalaciones (17) el 6 de abril. En la actualidad el número de dispositivos con la aplicación instalada está próximo a los 150 dispositivos. El número más alto de adquisiciones (37) se produjo el 10 de octubre del 2020, que coincide con en el periodo en el que Casual Learn apareció en prensa y en los canales oficiales de la Universidad de Valladolid.

En la fase de implementación relacionada con el cliente se indicó que se trabajaba con Google Analytics, utilizando el SDK de Firebase, para el registro de los eventos relacionados con las tareas. Aparte del número de eventos (cuyo resumen se puede ver en la Tabla 6.3) este servicio ha generado otras métricas de manera automática. Por ejemplo, se ha podido recuperar que el tiempo medio que cada usuario utiliza la aplicación asciende a 15 minutos y 38 segundos. Además, la pantalla con la que más tiempo interactúan es la de realización de la tarea educativa, seguido de la que permite una exploración de los POI a través del mapa y en la que se expone

<sup>2</sup><https://play.google.com/store/apps/details?id=es.uva.gsic.adolfinstro>

<sup>3</sup><https://play.google.com/console/about>

<sup>4</sup><https://casuallearnapp.gsic.uva.es>

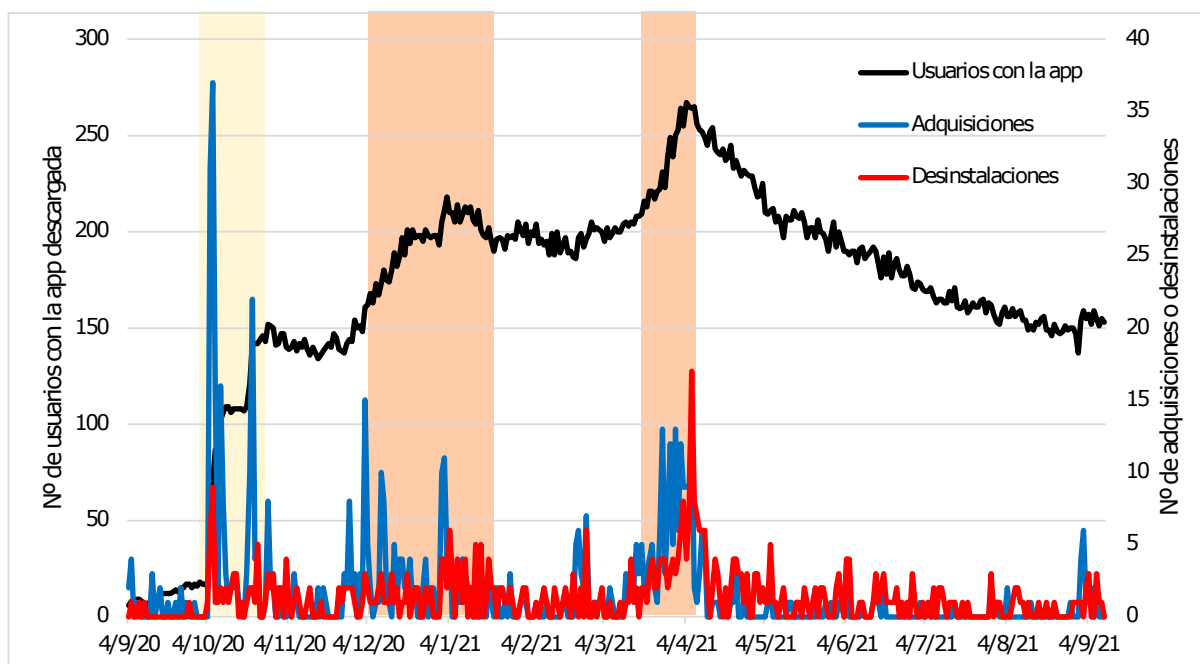


Fig. 6.3: Gráfico en la que se indica el número de usuarios con la aplicación instalada, el número de adquisiciones de usuario y el número de desinstalaciones. Las series de adquisiciones y desinstalaciones muestran el número de eventos diarios. La serie de usuarios con la aplicación instalada muestra en cada día el número total de usuarios que tienen la aplicación instalada. El periodo temporal con el fondo amarillo corresponde con las fechas de la aparición de Casual Learn en prensa y en los canales oficiales de la Universidad de Valladolid. El periodo con el fondo naranja corresponde con el periodo de desarrollo de los distintos pilotos donde destacan las instalaciones y desinstalaciones que se producen al inicio y al final de estos.

lo que se le va a solicitar realizar en la tarea de aprendizaje (con esta información el usuario decidirá si realizarla, posponerla o rechazarla).

Tabla 6.3: Eventos que se han registrado con Casual Learn.

Evento	Nº de eventos	Nº de usuarios	Eventos por usuario
Tarea completada	936	134	7
Contextos notificado	489	128	3,8
Tarea pospuesta	378	86	4,4
Respuesta modificada	157	26	6
Tarea rechazada	56	33	1,7
Tarea notificada (desuso)	22	7	7,1
Tarea reportada	11	6	1,8

Como puede observarse en la Tabla 6.3, el evento que más veces se ha registrado ha sido el de «Tarea completada», que se registra en Google Analytics a la vez que la respuesta a una tarea educativa –y la posible valoración del usuario a dicha tarea– se envía al servidor de la aplicación. En la Fig. 6.4 se muestra el número de eventos de cada tipo de acción que se registró diariamente en Google Analytics. En el caso del evento de «Tarea completada» (Fig. 6.4a) se puede observar que se producen en mayor volumen en los periodos en que se desarrollaron las experiencias educativas que utilizaron Casual Learn.

El segundo evento que más veces se ha registrado ha sido el de «Contexto notificado». Este evento se registra cuando la aplicación genera una notificación de un POI cercano al usuario.



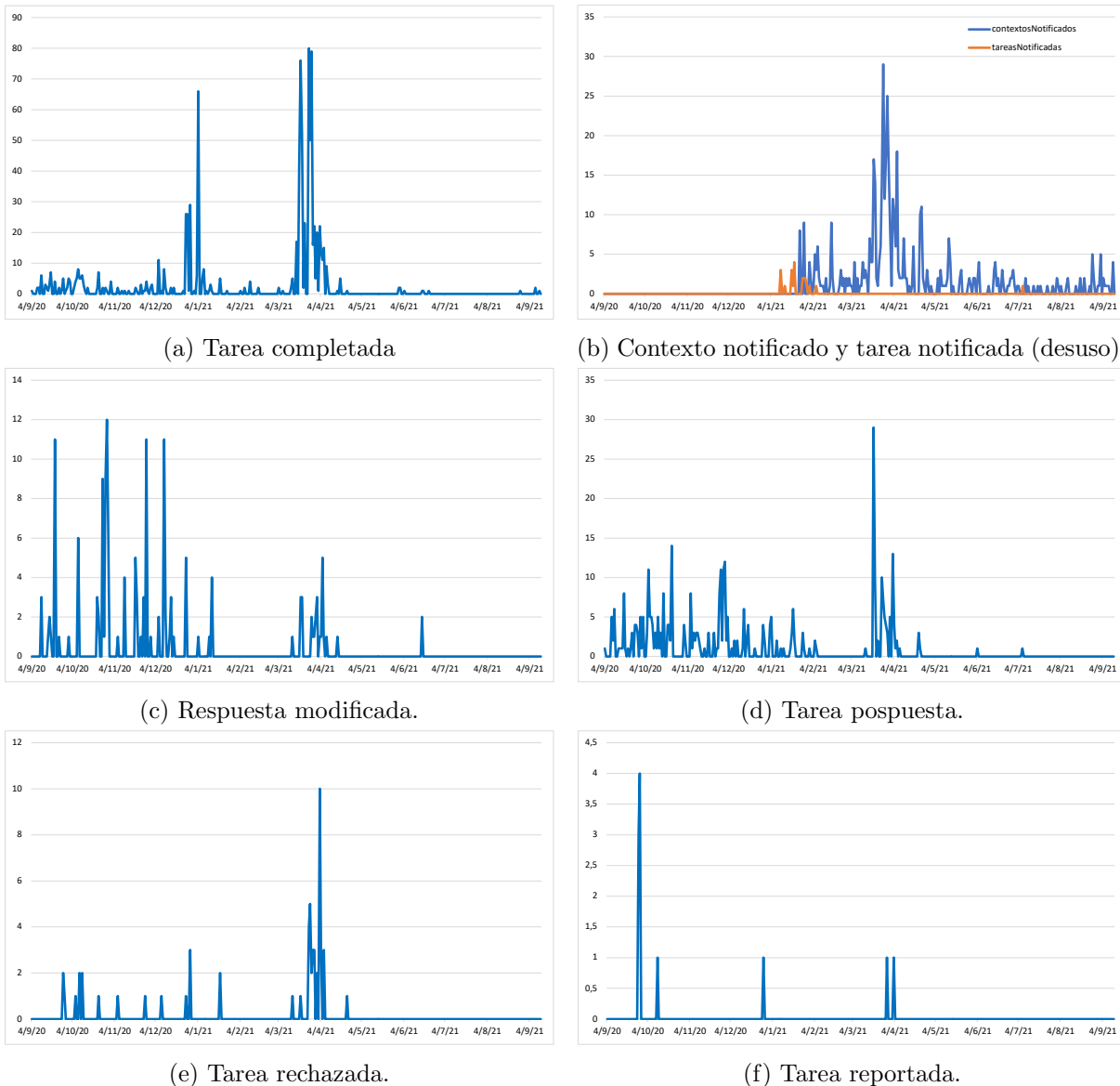


Fig. 6.4: Gráficos de eventos producidos en Casual Learn recogidos con Google Analytics. En cada uno de los gráficos se muestra el número de veces por día que se produjo el evento.

Está relacionado con el evento «Tarea notificada» puesto que en las versiones más antiguas del cliente se recomendaban en las notificaciones tareas que el usuario podría realizar (en la versión presentada solo se notifican contextos). Como se puede ver en la Fig. 6.4b los eventos de «Contexto notificado» no se han dejado de registrar desde su implementación aunque el volumen más alto de este tipo de eventos se concentró durante la última experiencia educativa.

«Tarea pospuesta», «Respuesta modificada» y «Tarea rechazada» son los siguientes eventos que aparecen en la Tabla 6.3. Mientras que «Tarea pospuesta» y «Tarea rechazada» se registran cuando el usuario realiza una acción desde la pantalla de información de la tarea (pulsación en uno de los botones para registrar la tarea en la lista correspondiente), el evento «Respuesta modificada» se produce cuando un usuario modifica una respuesta desde el cliente (editando la propia respuesta y/o la valoración de la tarea). En la Fig. 6.4c se puede ver que la modificación de la respuesta se utilizó más en las etapas iniciales del cliente, aunque esta pantalla apenas ha sufrido modificaciones desde que la aplicación se encuentra en producción. En la Fig. 6.4d y en la

Fig. 6.4e se observa que cuando más se utilizaron las opciones de rechazar y posponer las tareas educativas fue en la última experiencia educativa. Este hecho se produjo por una confusión: los estudiantes pensaban que al pasar las tareas educativas a las listas de tareas pospuestas o rechazadas evitaban la restricción de tener que realizar la tarea de aprendizaje cerca del POI.

En último lugar de la Tabla 6.3 se encuentra el evento «Tarea reportada», que se registra cuando un usuario que esté realizando una tarea educativa reporta un problema. Con este evento se espera que la comunidad de usuarios ayude a mejorar la información almacenada en Casual Learn SPARQL. Aunque el usuario no envíe un correo electrónico para dar más detalles del error encontrado, el evento estudiando se registra en Google Analytics junto con el IRI de la tarea reportada. De esta forma se puede comprobar si existe algún problema en de manera manual. Los instantes en los que un usuario ha reportado una tarea se pueden observar en la Fig. 6.4f.

En la Fig. 6.5 se muestran los días en los que se ha registrado un fallo (o más) en el cliente de Casual Learn. El total de errores detectados asciende a 45. Firebase registra, aparte del instante en el que se produjo el problema y la información del dispositivo en el que se produjo (modelo, versión de Android utilizada, etc.), la pila de métodos llamados hasta que se produjo la extensión (tal y como lo haría el método `printStackTrace`<sup>5</sup> de la clase `Throwable`). Esta información es muy útil para resolver problemas que no se habían encontrado previamente. Por ejemplo, el periodo en el que se concentraron más fallos fue durante las primeras pruebas piloto (entre diciembre de 2020 y enero de 2021). Cuando estos pilotos finalizaron se implementaron una serie de modificaciones en el cliente donde se tuvo en cuenta la información de fallos. Por ello, a pesar de que el número de estudiantes del segundo piloto era mayor que en los primeros se produjeron menos fallos.

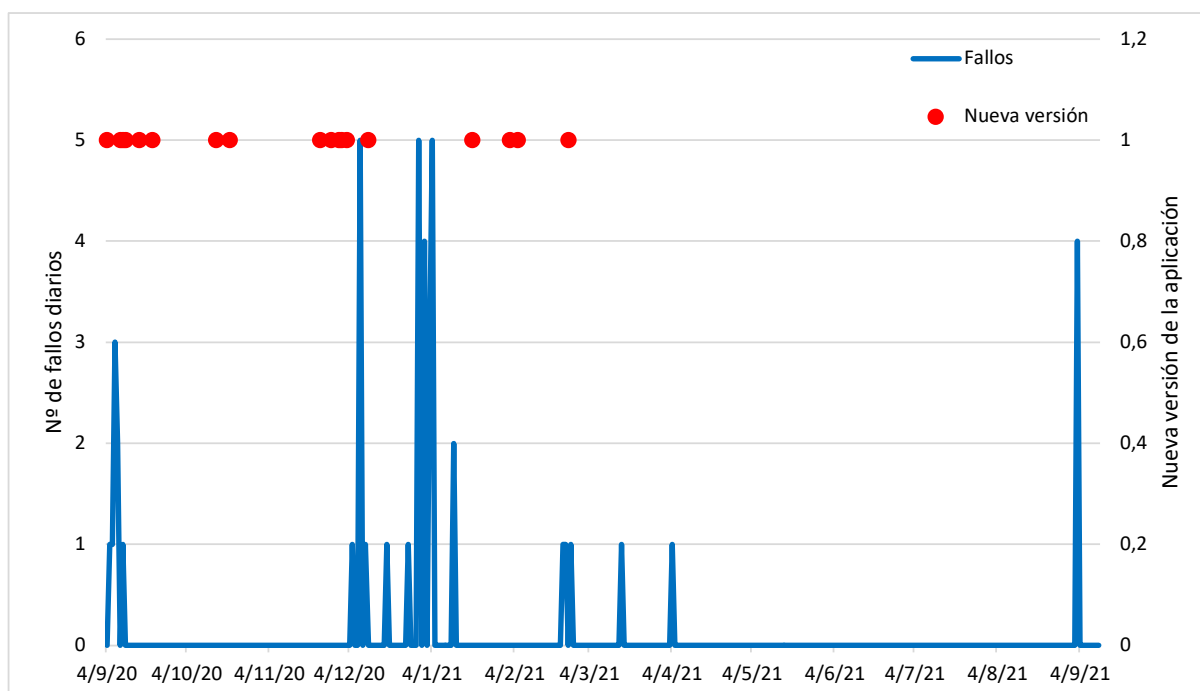


Fig. 6.5: Gráfico con los fallos diarios que se han producido con Casual Learn desde que se encuentra en producción junto con el día en el que se lanzó una nueva versión de la aplicación a través de Google Play.

<sup>5</sup><https://docs.oracle.com/javase/8/docs/api/java/lang/Throwable.html#printStackTrace-->

### 6.3.1. Pilotos realizados con la aplicación

Casual Learn ha servido de soporte para la realización de cuatro experiencias educativas. Dos de estos pilotos se realizaron en Palencia y los otros dos en Valladolid (España). En el piloto «El gótico en Palencia» e «Industrialización de Valladolid» la actividad contaba como parte de la asignatura donde se realizó. Por otro lado, en los pilotos «El arte medieval en Palencia» y «Burguesía en Valladolid» se les recompensaba a los estudiantes subiéndoles la nota si lo completaban. Los estudiantes de los pilotos desarrollados se encontraban en el rango de edad para los que se planteó la aplicación Casual Learn. Los estudiantes de la ciudad de Palencia y los del primer piloto de Valladolid cursaban el primer año de bachiller (16 – 17 años), mientras que los del segundo piloto desarrollado en Valladolid se encontraban en el último curso de Educación Secundaria Obligatoria (15 – 16 años). En la Tabla 6.4 se puede obtener un resumen del resto de la información de cada piloto.

Tabla 6.4: Resumen de las experiencias educativas que han utilizado Casual Learn en la que indica el tema del que trabaja el piloto, el número de estudiantes que lo componían, las fechas de inicio y fin del piloto y el número de tareas que completaron. Estos datos se han obtenido de [Rui+21a; Rui+21d].

Piloto	Nº. estud.	Inicio – Fin	Nº. respuestas
El gótico en Palencia	11	4/12/20 – 20/12/20	26
El arte medieval en Palencia	5	18/12/20 – 10/1/21	60
Industrialización de Valladolid	43	18/12/20 – 15/1/21	103
Burguesía en Valladolid	80	19/3/21 – 7/4/21	451

En todos los pilotos los profesores, y en el último piloto también los estudiantes, propusieron nuevas tareas educativas y POI para agregarlas a Casual Learn SPARQL y realizarlas posteriormente. Esto implicó que se tuviera que realizar un cambio en la forma de presentar los POI para que los estudiantes pudieran diferenciar fácilmente cuáles pertenecían a su piloto y cuáles no. Se solucionó otorgando un color a cada piloto y agregando dicho color al borde del marcador que se coloca sobre el mapa. También se utilizaría el mismo distintivo para el borde de las tarjetas de las tareas que se hayan agregado para el piloto.

Además, y dependiendo también del piloto, los estudiantes debían demostrar que habían realizado las tareas a su profesor. En unos casos los estudiantes utilizaron los métodos del sistema para compartir las tareas: o bien compartieron con el profesor el identificador de cada uno de los portafolios privados; o los estudiantes compartieron las respuestas que realizaban con su docente a través del cliente de Microsoft Teams. En otros casos los estudiantes tenían que realizar una presentación o entregar una redacción por lo que no utilizaban la aplicación Casual Learn para demostrar la realización de la actividad.

Los pilotos se desarrollaron sin incidencias técnicas graves. Solamente se tuvo que intervenir en dos situaciones: por un punto de interés cuyas coordenadas no representaban la posición real del POI; por una incidencia con la nube de cómputo. El primer problema se solucionó modificando la posición del POI en el repositorio de triplas. Para el segundo se tuvo que esperar hasta que la nube de cómputo volvió a estar operativa para volver a iniciar el servidor (se estuvo un par de horas sin servicio).

Las experiencias educativas en las que se utilizó Casual Learn sirvieron para mejorar la aplicación. Después de realizar cada una de ellas se les solicitaba a los estudiantes, de manera opcional, que realizaran un cuestionario para comprobar el grado de usabilidad. Los tres primeros pilotos se desarrollaron con la misma versión de la aplicación y después de que 49 estudiantes realizaran la encuesta se consiguieron 61,9 puntos sobre 100 en la escala SUS [Bro96] (*System*

*Usability Scale*) («OK» según [BKM09]).

A partir de la información que indicaron los usuarios se volvió a la etapa de análisis y diseño de la metodología ágil empleada y con lo que se realizaron cambios tanto en el cliente y el servidor de Casual Learn como en Casual Learn SPARQL. En Casual Learn SPARQL se agregó el concepto de POI: la información de las tareas informativas (las cuales no solicitaban realizar ninguna acción) pasó a formar parte de la descripción de los POI; las tareas que solicitaban ir del lugar donde se encontraba a otro relativamente cercano se eliminaron. El motivo de esta segunda eliminación estaba motivada porque a los usuarios les causaba confusión que una tarea les enviara a un punto donde ya habían estado. Además, la distancia entre estas tareas se medía en línea recta, por lo que en ocasiones desplazarse de un lugar a otro le suponía al usuario más problemas de los que originalmente se pensaron. Por ejemplo, Valladolid está dividida por una vía ferroviaria terrestre que solamente se puede atravesar por determinados puntos.

Las modificaciones en el punto SPARQL tuvieron como consecuencia que se agregara un nuevo recurso en el servidor para que el cliente solicitara esta nueva información. Con los POI el cliente tiene que gestionar menos información que en la versión anterior cuando se solicita la información de una zona. Por este motivo se mejoró el rendimiento de la pantalla de búsqueda activa. Otro cambio que se produjo en el cliente es que se pasó de notificar tareas educativas a puntos de interés cercanos para poder solicitar al servidor las mismas operaciones en la búsqueda activa y en la pasiva. Por último, se refinó la interfaz de usuario con el objetivo de modernizarla: se agregó el menú lateral y las tarjetas informativas de POI y se modificó la que existía de las tareas educativas. Con estos cambios, una vez finalizado el último piloto, se volvió a solicitar a los nuevos estudiantes que realizaran el mismo cuestionario que hicieron los anteriores. En este caso se obtuvo una puntuación de 68,0 sobre 100 puntos en la escala SUS respondiendo a la encuesta 76 estudiantes. Este valor sigue siendo de «OK» [BKM09], pero se ha comprobado que la metodología ágil utilizada ha servido para mejorar la usabilidad de la aplicación. Se espera que esta puntuación continúe aumentando a medida que se implementen las sugerencias de estos nuevos usuarios.

### 6.4. Conclusiones

De la información que se ha conseguido extraer de los distintos puntos del sistema se pueden obtener varias conclusiones. Una de ellas es que la promoción que se realizó en medios de comunicación locales y regionales y a través de los canales de la Universidad de Valladolid sirvió para tener una base de usuarios que hasta entonces no se había logrado. Otra conclusión es que una parte de los estudiantes que realizaron los pilotos eliminaron la aplicación a las pocas semanas. Por este motivo, si no se consigue dar un valor añadido a la aplicación, este tipo de usuarios verán a Casual Learn como una herramienta que solo se utiliza en clase, por lo que la eliminarán cuando terminen con el temario donde se utilice. Otros usuarios veían a Casual Learn como una aplicación que recomendarían a sus conocidos, sobretodo si saben que estos van a realizar una visita a otra ciudad (uso turístico). También hay que tener en cuenta que la información contenida en Casual Learn SPARQL está restringida a Castilla y León, por lo que sería deseable que este repositorio tuviera información de otros territorios para aumentar el número de potenciales usuarios de la aplicación.

Salvo los pequeños problemas técnicos que se han relatado en este capítulo Casual Learn ha sido capaz de soportar la carga de usuarios en todo momento. Los tiempos de respuesta que ofrece son lo suficientemente bajos como para que al utilizar el cliente no se aprecien todas las comunicaciones que se realizan. El peor resultado se ofrece cuando se hace uso del sistema de

recomendación externo a Casual Learn, por lo que se debería comenzar a buscar un sustituto. Además, el conjunto del sistema ha conseguido cumplir con el objetivo principal de este TFM, que los usuarios utilicen el contenido de Casual Learn SPARQL de forma totalmente transparente. Los tiempos de respuesta del servidor de las operaciones que más se realizan (solicitud de POI de una zona y tareas asociadas a un POI) son bajos, por lo que se garantiza que el usuario tenga una sensación de fluidez al utilizar el sistema.

Otra forma de medir el grado de validez del sistema es por los reconocimientos que ha recibido. La aplicación Casual Learn ha sido premiada con la *best demo* en la «*Fifteenth European Conference on Technology Enhanced Learning*» y en el «IV Concurso de Datos Abiertos de la Junta de Castilla y León» en la categoría a mejor recurso educativo<sup>6</sup>. La experiencia con Casual Learn ha servido para llevar a cabo las publicaciones [Rui+20; Rui+21d] y los artículos en revisión [Rui+21b; Rui+21a].

---

<sup>6</sup><https://datosabiertos.jcyl.es/web/es/premiados-concurso-datos-abiertos.html>



## Capítulo 7

# Conclusiones y líneas de trabajo futuro

En el capítulo final de este TFM se relaciona el objetivo principal que motivó su desarrollo con los resultados que se han obtenido, destacando el desarrollo de la aplicación Casual Learn. Casual Learn utiliza datos abiertos enlazados para permitir que sus usuarios realicen tareas educativas, busquen información de los monumentos de su entorno a través de un mapa interactivo, configuren el envío de avisos para que les notifique cuando se encuentren próximos a un punto de interés. . . Pero, como es lógico, a medida que se ha ido desarrollando la idea inicial se han podido detectar limitaciones que inicialmente no se habían tenido en cuenta. Para solventar los problemas encontrados se realiza una serie de propuestas.

### 7.1. Conclusiones del trabajo realizado

La conservación del patrimonio cultural es fundamental tanto a nivel económico como social. Las ciudades en las que se favoreció la destrucción de grandes monumentos arquitectónicos –como los antiguos monasterios y palacios– con promesas de modernización y rápidos beneficios económicos han visto como, a largo plazo, han salido perjudicadas. La pérdida de estos edificios no solo ha afectado al nivel de la riqueza cultural existente en esas ciudades, sino que influirá negativamente a largo plazo por todo el posible turismo que pudiera visitarlas y las actividades que se pudieran realizar alrededor de este. Un claro ejemplo de esta destrucción de patrimonio se produjo en la ciudad de Valladolid, puesto que, a pesar de su importancia histórica, no cuenta con la cantidad de monumentos y lugares históricos que debería poseer si se compara con las ciudades de su alrededor (algunas de ellas declaradas Ciudades Patrimonio de la Humanidad por la UNESCO).

Para que no vuelva a ocurrir la destrucción del patrimonio cultural los distintos gobiernos han legislado, en mayor o menor medida, para intentar protegerlo. La protección del patrimonio por parte de los gobiernos puede producirse a nivel administrativo (multando si se detecta alguna infracción o subvencionando la conservación y restauración del patrimonio) y a nivel educativo. Muchos países, entre los que se encuentra España, han incluido dentro del currículo educativo objetivos para que las nuevas generaciones conozcan y aprendan a respetar y conservar el patrimonio de su entorno.

En ese TFM se ha propuesto Casual Learn, una aplicación diseñada con el objetivo de contribuir a la educación relacionada con el patrimonio cultural que se basa en el aprendizaje

ubicuo. El aprendizaje ubicuo surge de la evolución que han realizado las metodologías educativas durante los últimos años gracias a la inclusión de las TIC (Tecnologías de la Información y la Comunicación) en todos los aspectos de la vida cotidiana. De esta omnipresencia se puede aprovechar la educación para ofrecer a los estudiantes contenidos en todo momento, dejando de estar limitada al aula. Utilizando el aprendizaje ubicuo los docentes pueden romper las fronteras del aula para que sus estudiantes puedan relacionar con más facilidad los elementos de su entorno con los conocimientos aprendidos en clase. El aprendizaje ubicuo precisa de la tecnología, pero la tecnología no queda limitada al elemento físico con el que interactúan los estudiantes. Para que se pueda producir este tipo de aprendizaje se necesita de aplicaciones con las que puedan interactuar los estudiantes.

Casual Learn se ha diseñado como el elemento que permite a los estudiantes obtener información sobre el patrimonio cultural de su entorno desde cualquier lugar a través de un dispositivo electrónico. La información que ofrece la aplicación desarrollada proviene de Casual Learn SPARQL. Casual Learn SPARQL permite la obtención de la información que almacena utilizando SPARQL, tecnología directamente relacionada con la Web Semántica. Toda la información que se obtenga de este repositorio se recupera a través de una serie de peticiones SPARQL.

Casual Learn no se limita a informar a los estudiantes sobre detalles y hechos relevantes de los puntos de interés, ya que además, permite la realización de tareas educativas. Al realizar estas tareas, los alumnos podrán compartirlas a través de un portafolio propio de Casual Learn o utilizando algunas de las redes sociales y entornos educativos más populares. La acción de compartir las respuestas servirá para cerrar el círculo educativo: los profesores pueden sugerir la utilización de Casual Learn a sus estudiantes y estos podrán compartir con los docentes las respuestas a las tareas de aprendizaje que hayan contestado.

El registro de las respuestas de los estudiantes en un lugar donde siempre se pudiera acceder a ellas y la utilización de tecnologías semánticas en un único lugar facilitó la decisión de estructurar Casual Learn como una aplicación distribuida. Las respuestas se almacenarán en una base de datos accesible a través de un servidor, que también ofrecerá una API RESTful para que se pueda recuperar el contenido de estas respuestas. Además, el servidor funcionará como una pasarela, encargándose de traducir las peticiones de información sobre POI y tareas educativas de los usuarios en solicitudes semánticas para que Casual Learn SPARQL las pueda comprender. Se ha comprobado que esta traducción se realiza rápidamente, pudiéndose utilizar en aplicaciones en tiempo real puesto que el servidor necesita –en promedio– de  $28ms$  para responder a solicitudes de los POI contenidos en una zona determinada y  $78ms$  para devolver las tareas de aprendizaje asociadas a un POI.

Estos tiempos promedios han sido calculados a partir de la información almacenada por el registro de eventos que utiliza el servidor en producción. La mayor parte de los registros reportados en el servidor habrán sido generados por los usuarios que utilicen el cliente de la aplicación Casual Learn (algunos usuarios –o aplicaciones– pueden haber realizado peticiones directamente al servidor sin utilizar el cliente diseñado). El cliente se ha diseñado como una aplicación nativa de Android, disponible desde el 4 de septiembre de 2020 en Google Play<sup>1</sup> y que ha sido adquirida desde esta tienda 555 veces. En el momento de la escritura de este TFM cuenta con una valoración de 4,6 sobre 5 estrellas. La crítica más negativa que un usuario ha realizado a través del sistema de valoraciones está relacionada con cómo se gestionan los mapas descargados y qué permisos se solicitan para almacenarlos en la memoria compartida del dispositivo. A este usuario se le contestó que el sistema de almacenamiento de mapas se realizaba así para evitar que la misma información se descargara más de una vez. Además, almacenando la información

---

<sup>1</sup><https://play.google.com/store/apps/details?id=es.uva.gsic.adolfinstro>



en la memoria secundaria del dispositivo se consigue mejorar la fluidez de la aplicación.

El cliente cuenta con dos modos de funcionamiento: búsqueda interactiva y búsqueda en segundo plano. En el primer caso, será el usuario el que explore la información disponible. En el segundo, el cliente de Casual Learn ofrecerá información al usuario en base a la ubicación donde se entre el usuario y qué tiene este a su alrededor. En ambos casos se proporciona información sobre puntos de interés. Asociados a los POI se encuentran las tareas educativas que permitirán al usuario demostrar sus conocimientos. Para el registro de las acciones que realiza el usuario con la aplicación se ha utilizado Google Analytics a través de Firebase. Gracias a la utilización de este servicio, se ha podido saber la cantidad de tareas completadas desde que Casual Learn se encuentra en producción. Este número asciende a 936 tareas realizadas por 134 usuarios. De estas 936 respuestas 640 fueron realizadas por los estudiantes de las diferentes experiencias educativas en las que se utilizó Casual Learn. Con estos valores se pueden obtener dos conclusiones: las pruebas piloto han sido muy importantes para alcanzar los resultados reportados en este TFM y que usuarios externos han utilizado Casual Learn para conocer los monumentos de su entorno realizando tareas educativas.

Con lo reportado hasta el momento se podría concluir que **se ha cumplido con el objetivo del TFM**: se ha desarrollado una aplicación que, utilizando el conjunto de datos contenidos en Casual Learn SPARQL, permite tanto la obtención de información relacionada con el patrimonio cultural de su entorno como la realización de tareas educativas y la publicación de las respuestas que proporcionen a los usuarios a estas. También hay que resaltar que con el diseño arquitectónico de separación de capas que se ha realizado para la aplicación, otros sistemas podrán utilizar el servidor como punto desde donde extraer información de tareas educativas y puntos de interés. Estas peticiones de información se programarán como peticiones HTTP, una tecnología mucho más extendida y conocida que las peticiones SPARQL. Por este motivo el servidor puede ayudar a aquellos desarrolladores e investigadores que no sepan construir consultas semánticas pero necesiten acceder al conjunto de datos almacenados en Casual Learn SPARQL.

La validación de la aplicación desarrollada, en vez de haberse producido mediante pruebas sintéticas, se ha realizado a través de varias experiencias educativas [Rui+21a]. Concretamente se han realizado **cuatro pruebas piloto** en los que han participado **más de 100 estudiantes** que cursaban el último año de Educación Secundaria Obligatoria y el primero de bachiller, que es el público objetivo de la aplicación. Después de realizar las primeras tres pruebas piloto, que transcurrieron entre diciembre de 2020 y enero de 2021, se solicitó que los estudiantes que lo desearan respondieran una encuesta sobre *usabilidad*, obteniendo una puntuación SUS de 60,6 puntos sobre 100. Para la cuarta experiencia educativa se utilizó la versión actualizada con la realimentación que proporcionaron los usuarios –versión que se ha reportado en este TFM– y después de su finalización se realizó la misma encuesta a los nuevos usuarios consiguiendo esta vez 68,8 puntos sobre 100 («OK» según [BKM09]). Por este motivo, aparte de que el conjunto de pruebas piloto sirviesen para validar el sistema y demostrar que se ha cumplido con el objetivo que motivó su realización, también han servido para mejorar la aplicación. Esto ha sido posible debido a la metodología usada en el desarrollo del TFM ya que, entre otras ventajas, permite volver a etapas anteriores para solucionar problemas que previamente no se habían planteado. La calidad de la propuesta de Casual Learn también viene avalada por los **premios concedidos** a la aplicación: *Best Demo* de ECTEL 2020 y recurso educativo del IV Concurso de Datos Abiertos organizado por la Junta de Castilla y León. Además, el desarrollo de esta aplicación ha tenido el respaldo del grupo de investigación donde se ha realizado a través de las **publicaciones** [Rui+20; Rui+21d] y los artículos en revisión [Rui+21b; Rui+21a].

## 7.2. Limitaciones y líneas de trabajo futuro

La versión presentada en este TFM ha cumplido con los objetivos propuestos al inicio de este. A pesar de ello se han detectado una serie de limitaciones a lo largo de su implementación y documentación. La primera de ellas está relacionada con la generación e incorporación de nueva información a Casual Learn SPARQL. En las experiencias educativas se ha comprobado que, aunque los docentes consideran interesantes las tareas base que se encuentran en el repositorio que utiliza la aplicación, desean agregar otras para que las realicen sus alumnos. Para suplir esta carencia, los profesores debían enviar la información de las tareas al grupo de investigación GSIC/EMIC. Una vez que se disponía de la información de los profesores, se adaptaba para que fuera entendida por el repositorio de triplas donde se almacenaba.

Otra limitación es que los profesores no pueden determinar el orden en el que sus estudiantes han realizado las tareas. Esto puede resultar interesante si el docente desea, por ejemplo, comparar dos estilos arquitectónicos diferentes, y su idea es que primero visitaran edificios de un estilo y después los del otro para que pudieran compararlos.

Los estudiantes pudieron diferenciar las tareas educativas y puntos de interés que sus profesores crearon gracias a que se adaptó el cliente para que asignara un color específico a cada experiencia educativa. Esta solución supone otra limitación puesto que, a medida que se incrementa el número de docentes que utilicen este sistema, se dificultará y ensuciará la interfaz gráfica del cliente ya que será necesario utilizar más colores para diferenciar cada uno de los pilotos.

Otra de las limitaciones de la versión actual de Casual Learn es que únicamente está disponible para dispositivos Android. Esto supuso que potenciales usuarios no pudieran utilizar la aplicación ya que su dispositivo móvil ejecutaba el sistema operativo iOS. Esta situación no es la ideal debido a que puede crear un ambiente de frustración alrededor del alumno ya que sus compañeros pueden utilizar una herramienta que él no puede utilizar. Otra forma de que los estudiantes se puedan frustrar y desanimar es debido a la imposibilidad de recibir una realimentación de las tareas que realizan por parte de su profesor a través de la versión actual de Casual Learn.

La última limitación de la que se va a hablar está relacionada con Recombee. Como se ha visto, Casual Learn utiliza este sistema para ordenar las tareas educativas de un POI notificado según los intereses del usuario al que se las esté mostrando. El primer problema es el tiempo que se necesita para obtener esta información: más de 0,5s. Este tiempo es lo suficientemente elevado como para que el usuario que utilice el sistema lo detecte (algo no deseable). El segundo es que Recombee es un programa comercial con una licencia privativa.

Las cuatro primeras limitaciones encontradas se podrían solventar con la creación de una herramienta de edición de información y una actualización de la ontología que se utiliza en Casual Learn SPARQL. Esta tarea no comenzaría de cero puesto que ya se ha comenzado a trabajar en LocalizARTE [Gar+21], una aplicación cuyo cliente se podría utilizar desde cualquier navegador moderno (solución a la **compatibilidad** con los diferentes sistemas). Esta aplicación permitirá que los profesores **agreguen y gestionen** POI, tareas educativas e **itinerarios** (tareas que se deben realizar en un determinado orden) a través de una interfaz sencilla que oculte las tecnologías relacionadas con la Web Semántica. La nueva información se integrará en un nuevo grafo dentro de Casual Learn SPARQL. Los estudiantes podrán realizar las tareas educativas de la misma forma que con Casual Learn, pero además podrán realizar otras que pertenezcan a otros espacios (como el espacio mapa virtual). Estas respuestas pasarán a formar parte de un listado en el que solamente tendrá acceso el estudiante y su docente. Este último podrá

**proporcionar una realimentación** a las respuestas que proporcione el estudiante.

Por otro lado, la limitación que plantea Recombee se podría plantear de otra forma. En la versión actual solo se reordenan las tareas ofrecidas al estudiante en función de las valoraciones que haya podido dar previamente (él y los usuarios que Recombee considere similares). Además las notificaciones de POI solo se basan en la distancia que exista entre la posición del usuario y la ubicación del POI. En ambos casos no se está aprovechando todo el potencial que puede ofrecer la información existente en Casual Learn SPARQL. Como se estudió en el capítulo 2, las clases que formaban su ontología eran tareas, espacios físicos y **temas**. Los temas indicaban la categoría con la que estaba relacionada cada tarea. De manera indirecta también indican las categorías con las que está relacionado cada POI (a través de la propiedad `hasContext` entre tareas y POI). Por este motivo se podría realizar un **sistema de recomendación basado en datos abiertos enlazados**. Sin hacer uso de soluciones externas este sistema podría, por ejemplo, recomendar tareas y lugares relacionados con una tarea educativa que el usuario acabara de completar.



# Referencias

- [AH11] D. Allemang y J. Hendler. *Semantic Web for the Working Ontologist*. 2nd ed. Morgan Kaufmann Publishers/Elsevier, 2011.
- [ANDVOL] Android Developers. *Cómo configurar RequestQueue* — *Desarrolladores de Android*. URL: <https://developer.android.com/training/volley/requestqueue#java> (última visita en 20 de septiembre de 2021).
- [BC95] T. Berners-Lee y D.W. Connolly. *Hypertext Markup Language - 2.0*. RFC 1866. Noviembre de 1995.
- [Bec+01] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Adrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland y Dave Thomas. *Manifesto for Agile Software Development*. 2001. URL: <http://agilemanifesto.org/iso/en/manifesto.html> (última visita en 20 de septiembre de 2021).
- [Ber06] Tim Berners-Lee. *Linked Data - Design Issues*. 27 de julio de 2006. URL: <https://www.w3.org/DesignIssues/LinkedData> (última visita en 20 de septiembre de 2021).
- [BFM05] T. Berners-Lee, R.T. Fielding y L.M. Masinter. *Uniform Resource Identifier (URI): Generic Syntax*. RFC 3986. Enero de 2005.
- [BHL01] T. Berners-Lee, J. Hendler y O. Lassila. «The semantic web». En: *Scientific american* 284.5 (2001), págs. 34-43.
- [BKM09] A. Bangor, P. Kortum y J. Miller. «Determining what individual SUS scores mean: Adding an adjective rating scale». En: *Journal of Usability Studies* 4.3 (2009), págs. 114-123.
- [BOE15] Agencia Estatal Boletín Oficial del Estado. *Real Decreto 1105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y del Bachillerato*. Boletín Oficial del Estado, enero de 2015. URL: <https://www.boe.es/eli/es/rd/2014/12/26/1105> (última visita en 20 de septiembre de 2021).
- [Bra14] T. Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 7159. Marzo de 2014.
- [Bro96] J. Brooke. «SUS: A quick and dirty usability scale». En: *Usability Evaluation in Industry* (1996).
- [Can15] Esteban Cano. *Dispositivos digitales móviles en educación : el aprendizaje ubicuo*. Madrid: Narcea, S.A. de Ediciones, 2015.
- [CB15] «Chapter Six - Query Processing». En: *RDF Database Systems*. Ed. por O. Curé y G. Blin. Boston: Morgan Kaufmann, 2015, págs. 145-167.

- [CL07] Trudi Cooper y Terence Love. «E-Portfolios in e-Learning». En: *Advanced principles of effective e-learning* (2007), págs. 267-292.
- [CWL14] Richard Cyganiak, David Wood y Markus Lanthaler. *RDF 1.1 Concepts and Abstract Syntax*. W3C, febrero de 2014. URL: <https://www.w3.org/TR/rdf11-concepts/> (última visita en 20 de septiembre de 2021).
- [DACYL] Junta de Castilla y León. *Municipios de Castilla y León — Datos Abiertos — Junta de Castilla y León*. URL: <http://datosabiertos.jcyl.es/web/jcyl/set/es/sector-publico/municipios/1284278782067> (última visita en 20 de septiembre de 2021).
- [DS05] M.J. Dürst y M. Suignard. *Internationalized Resource Identifiers (IRIs)*. RFC 3987. Enero de 2005.
- [DuC13] Bob DuCharme. *Learning SPARQL: Querying and Updating with SPARQL 1.1*. O'Reilly Media, junio de 2013.
- [EUOD17] European Union Open Data Portal. *Conformant SPARQL protocol service*. Abril de 2017. URL: <https://data.europa.eu/euodp/en/data/dataset/eu-open-data-portal-sparql-endpoint> (última visita en 20 de septiembre de 2021).
- [Fel+14] Alexander Felfernig, Michael Jeran, Gerald Ninaus, Florian Reinfrank, Stefan Reiterer y Martin Stettinger. «Basic Approaches in Recommendation Systems». En: *Recommendation Systems in Software Engineering*. Ed. por Martin P. Robillard, Walid Maalej, Robert J. Walker y Thomas Zimmermann. Diciembre de 2014, págs. 15-37.
- [Fie00] Roy T Fielding. «Architectural styles and the design of network-based software architectures». Tesis doct. University of California, Irvine, 2000. Cap. 5. Representational State Transfer (REST).
- [FIRANA] Firebase. *Google Analytics — Firebase*. URL: <https://firebase.google.com/docs/analytics> (última visita en 20 de septiembre de 2021).
- [FIRAUT] Firebase. *Firebase Authentication*. URL: <https://firebase.google.com/docs/auth> (última visita en 20 de septiembre de 2021).
- [Fou] Apache Software Foundation. *Maven — Welcome to Apache Maven*. URL: <https://maven.apache.org/index.html> (última visita en 20 de septiembre de 2021).
- [FR14] R.T. Fielding y J. Reschke. *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*. RFC 7231. Junio de 2014.
- [Gal16] Vanesa Gallego Lema. «El aprendizaje ubicuo en Educación Física en el Medio Natural: un estudio de caso». Tesis doct. Universidad de Valladolid, España, 2016. Cap. 1. Retos tecnológicos y educativos en el aprendizaje ubicuo.
- [Gar+21] P. García-Zarza, A. Ruiz-Calleja, M.L. Bote-Lorenzo, G. Vega-Gorgojo, E. Gómez-Sánchez y J.I. Asensio-Pérez. «Hacia la anotación y realización de tareas de aprendizaje ubicuo en el contexto de historia del arte». En: *XV Jornadas de Ingeniería Telemática*. En proceso de revisión. La Coruña, España, 2021.
- [Git] GitHub. *The State of the Octoverse — The State of the Octoverse explores a year of change with new deep dives into developer productivity, security, and how we build communities on GitHub*. URL: <https://octoverse.github.com/> (última visita en 20 de septiembre de 2021).

- [GSICa] Grupo de Sistemas Inteligentes y Cooperativos / Educación, Medios y Cultura (GSIC/EMIC). *Casual Learn SPARQL*. URL: <https://casuallearn.gsic.uva.es/sparql> (última visita en 20 de septiembre de 2021).
- [GSICb] Grupo de Sistemas Inteligentes y Cooperativos / Educación, Medios y Cultura (GSIC/EMIC). *Grupo de Sistemas Inteligentes y Cooperativos / Educación, Medios, Informática y Cultura (GSIC/EMIC)*. Grupo de Investigación. URL: <https://gsic.uva.es> (última visita en 20 de septiembre de 2021).
- [HS13] Steve Harris y Andy Seaborne. *SPARQL 1.1 Query Language*. W3C, marzo de 2013. URL: <https://www.w3.org/TR/sparql11-query/> (última visita en 20 de septiembre de 2021).
- [INE20] Instituto Nacional de Estadística. *Equipamiento y uso de TIC en hogares. Año 2020*. Noviembre de 2020. URL: [https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=estadistica\\_C&cid=1254736176741&menu=ultiDatos&idp=1254735976608](https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=estadistica_C&cid=1254736176741&menu=ultiDatos&idp=1254735976608) (última visita en 20 de septiembre de 2021).
- [Koe16] George. Koelsch. *Requirements Writing for System Engineering*. 1st ed. 2016. Berkeley, CA: Apress, 2016.
- [Lou13] Jérôme Louvel. *Restlet in action : developing RESTful web APIs in Java*. Shelter Island: Manning, 2013.
- [MM19] A. Meier y Kaufmann M. *SQL & NoSQL Databases Models, Languages, Consistency Options and Architectures for Big Data Management*. 1st ed. 2019. Wiesbaden: Springer Fachmedien Wiesbaden, 2019.
- [Mon] MongoDB. *The MongoDB 4.4 Manual — MongoDB Manual*. URL: <https://docs.mongodb.com/v4.4/> (última visita en 20 de septiembre de 2021).
- [Nie+99] H. Nielsen, J. Mogul, L.M. Masinter, R.T. Fielding, J. Gettys, P.J. Leach y T. Berners-Lee. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616. Junio de 1999.
- [OAS21] Miller, Darrel and Whitlock, Jeremy and Gardiner, Marsh and Ralphson, Mike and Ratovsky, Ron and Sarid, Uri. *OpenAPI Specification v3.0.0 – Introduction, Definitions, & More*. URL: <https://spec.openapis.org/oas/v3.0.0> (última visita en 20 de septiembre de 2021).
- [OSM] OpenStreetMap. *OpenStreetMap*. URL: <https://www.openstreetmap.org/about/> (última visita en 20 de septiembre de 2021).
- [OSMDR] osmdroid. *osmdroid*. URL: <https://osmdroid.github.io/osmdroid/index.html> (última visita en 20 de septiembre de 2021).
- [Pis+20] G. Pishtari, M.J. Rodríguez-Triana, E.M. Sarmiento-Márquez, M. Pérez-Sanagustín, A. Ruiz-Calleja, P. Santos, L.P. Prieto, S. Serrano-Iglesias y T. Våljätaga. «Learning design and learning analytics in mobile and ubiquitous learning: A systematic review». En: *British Journal of Educational Technology* 51.4 (2020), págs. 1078-1100.
- [RAE] Real Academia Española. *tesela — definición — Diccionario de la lengua española — RAE – ASALE*. URL: <https://dle.rae.es/tesela> (última visita en 20 de septiembre de 2021).
- [Re03] R. Ramakrishnan y J. Ehrke. *Database management systems*. 3rd. ed. McGraw-Hill Education, 2003.
- [Reca] Recombee. *API Documentation — Recombee Docs*. URL: <https://docs.recombee.com/toc.html> (última visita en 20 de septiembre de 2021).

- [Recb] Recombee. *What Makes Our Recommendation Technology Unique — Recombee*. URL: <https://www.recombee.com/technology.html> (última visita en 20 de septiembre de 2021).
- [Rob57] C. C. Robusto. «The Cosine-Haversine Formula». En: *The American Mathematical Monthly* 64.1 (1957), págs. 38-40.
- [Rui+20] A. Ruiz-Calleja, M.L. Bote-Lorenzo, G. Vega-Gorgojo, S. Serrano-Iglesias, P. García-Zarza, J.I. Asensio-Pérez y E. Gómez-Sánchez. «CasualLearn: A smart application to learn History of Art». En: *European Conference on Technology Enhanced Learning*. Springer. 2020, págs. 472-476.
- [Rui+21a] A. Ruiz-Calleja, M.L. Bote-Lorenzo, J.I. Asensio-Pérez, S. Villagrà-Sobrino, V. Alonso-Prieto, E. Gómez-Sánchez, P. García-Zarza, S. Serrano-Iglesias y G. Vega-Gorgojo. «Orchestrating Ubiquitous Learning Situations about Cultural Heritage with Casual Learn». En: *Computers & Education* (2021). en proceso de revisión.
- [Rui+21b] A. Ruiz-Calleja, P. García-Zarza, P. Vega-Gorgojo, M.L. Bote-Lorenzo, E. Gómez-Sánchez, J.I. Asensio-Pérez, S. Serrano-Iglesias y A. Martínez-Monés. «Casual Learn: A Semantic Mobile Application for Learning Local Cultural Heritage». En: *Semantic Web Journal* (2021). en proceso de revisión.
- [Rui+21c] A. Ruiz-Calleja, G. Vega-Gorgojo, M.L. Bote-Lorenzo, J.I. Asensio-Pérez, Y. Dimitriadis y E. Gómez-Sánchez. «Supporting contextualized learning with Linked Open Data». En: *Journal of Web Semantics* 70 (2021), pág. 100657.
- [Rui+21d] A. Ruiz-Calleja, S.L. Villagrà-Sobrino, M.L. Bote-Lorenzo, S. Serrano-Iglesias, P. García-Zarza, V. Alonso-Prieto y J.I. Asensio-Pérez. «Orchestrating an ubiquitous learning situation about Cultural Heritage with Casual Learn». En: *European Conference on Technology Enhanced Learning*. Ed. por Springer. 2021.
- [SS05] Alberto Sillitti y Giancarlo Succi. «Requirements Engineering for Agile Methods». En: *Engineering and Managing Software Requirements*. Ed. por Aybüke Aürum y Claes Wohlin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, págs. 309-326. DOI: 10.1007/3-540-28244-0\_14.
- [UNE] UNESCO. *Patrimonio cultural*. URL: <https://es.unesco.org/fieldoffice/santiago/cultura/patrimonio> (última visita en 20 de septiembre de 2021).
- [W3CSW] *Semantic Web - W3C*. URL: <https://www.w3.org/standards/semanticweb/> (última visita en 20 de septiembre de 2021).



## Anexo A

# Documentación de la API del servidor

En las siguientes páginas se ha insertado el fichero con el que se generó la documentación de la API del servidor de Casual Learn desarrollada según los principios REST [Fie00]. Una vez que se desarrolló el fichero del Listado A.1 se creó la documentación utilizando el programa **OpenAPI Generator**<sup>1</sup>. Si se desea interactuar visualizar la versión final de la documentación e interactuar con ella se encuentra accesible en [https://casuallearnapp.gsic.uva.es/server\\_api](https://casuallearnapp.gsic.uva.es/server_api).

Listado A.1: Fichero en formato YAML con el que se genera la documentación de la API RESTful del servidor

```
1 openapi: 3.0.0
2
3 info:
4   title: 'Casual Learn Server API.'
5   version: '3.0.0'
6   contact:
7     name: 'Pablo García-Zarza'
8     email: 'pablogz@gsic.uva.es'
9   license:
10    name: 'Apache 2.0'
11    url: 'https://www.apache.org/licenses/LICENSE-2.0'
12
13 servers:
14   - url: https://casuallearnapp.gsic.uva.es/app
15
16 tags:
17   - name: POI
18   - name: Tareas educativas
19   - name: Respuestas
20   - name: Portafolio
21
22 components:
23   schemas:
24     POIArray:
25       type: array
26       items:
27         type: object
28         required:
29           - latitud
30           - longitud
```

<sup>1</sup><https://openapi-generator.tech/>

```

31     - contexto
32     - label
33     - nTareas
34     - creadoPor
35     - comment
36 properties:
37     latitud:
38         type: number
39         format: double
40         example: 41.653947
41     longitud:
42         type: number
43         format: double
44         example: -4.722827
45     contexto:
46         type: string
47         description: 'Identificador del POI.'
48         example: 'https://casuallearn.gsic.uva.es/context/
49             Iglesia_de_Santa_María_La_Antigua/4722827/41653947'
50     label:
51         type: string
52         example: 'Iglesia de Santa María La Antigua'
53         description: 'Nombre del monumento o el lugar'
54     nTareas:
55         type: integer
56         format: int32
57         example: 1
58         description: 'Número de tareas educativas agrupadas en el punto de
59             interés.'
60     creadoPor:
61         type: string
62         example: 'https://casuallearn.gsic.uva.es/researchers'
63     comment:
64         type: string
65         example: 'La iglesia de Santa María de La Antigua se levanta [...]
66             aspecto original.'
67         description: 'Descripción del monumento o el lugar.'
68     enlaceWiki:
69         type: string
70         example: 'https://es.wikipedia.org/wiki/Iglesia_de_Santa_Marí
71             a_La_Antigua'
72         description: 'Enlace a Wikipedia para consultar más información.'
73     example:
74     - latitud: 41.653947
75     - longitud: -4.722827
76     - contexto: 'https://casuallearn.gsic.uva.es/context/
77         Iglesia_de_Santa_María_La_Antigua/4722827/41653947'
78     - label: 'Iglesia de Santa María La Antigua'
79     - nTareas: 1
80     - creadoPor: 'https://casuallearn.gsic.uva.es/researchers'
81     - comment: 'La iglesia de Santa María de La Antigua se levanta [...]
82         aspecto original.'
83     - enlaceWiki: 'https://es.wikipedia.org/wiki/Iglesia_de_Santa_Marí
84         a_La_Antigua'
85     - latitud: 41.652692
86     - longitud: -4.723384
87     - contexto: 'https://casuallearn.gsic.uva.es/context/
88         Catedral_de_Valladolid/4723384/41652692'
89     - label: 'Catedral de Valladolid'
90     - nTareas: 9
91     - creadoPor: 'https://casuallearn.gsic.uva.es/researchers'
92     - comment: 'Purista y sobria, [...] principal, obra de Churriguera.'
```

```

85     enlaceWiki: ''
86
87   TaskArray:
88     type: array
89     items:
90       type: object
91       properties:
92         id:
93           type: string
94           description: 'IRI de la tarea educativa.'
95           example: 'https://casuallearn.gsic.uva.es/Catedral_de_Valladolid/
           encuentraEstilos'
96         comment:
97           type: string
98           description: 'Nombre del monumento o el lugar en el que se encuentre
           la tarea de aprendizaje.'
99           example: 'Catedral de Valladolid'
100        tipoRespuesta:
101          type: string
102          example: 'textoCorto'
103        recursoAsociadoTexto:
104          type: string
105          example: 'En este monumento puedes encontrar más de un estilo
           arquitectónico. Indica al menos dos distintos.'
106          description: 'Enunciado de la tarea educativa.'
107        thumbnail:
108          type: string
109          example: 'http://commons.wikimedia.org/wiki/Special:FilePath/
           ISanMartinCuellar2.jpg?width=300'
110          description: 'Imagen descriptiva de la tarea en baja resolución.'
111        imagen:
112          type: string
113          example: 'http://commons.wikimedia.org/wiki/Special:FilePath/
           ISanMartinCuellar2.jpg'
114          description: 'Imagen descriptiva de la tarea.'
115        textoLicencia:
116          type: string
117          example: 'CC BY-SA 3.0 - Lourdes Cardenal'
118        respuestaEsperada:
119          type: string
120          example: 'Renacentista y Barroco'
121        creadoPor:
122          type: string
123          example: 'https://casuallearn.gsic.uva.es/researchers'
124        contexto:
125          type: string
126          example: 'https://casuallearn.gsic.uva.es/context/
           Catedral_de_Valladolid/4723384/41652692'
127          description: 'Identificador del lugar o monumento donde está
           agrupada la tarea.'
128        latitud:
129          type: number
130          format: double
131          example: 41.652692
132        longitud:
133          type: number
134          format: double
135          example: -4.723384
136        example:
137          - id: 'https://casuallearn.gsic.uva.es/Catedral_de_Valladolid/
           encuentraEstilos'
138            comment: 'Catedral de Valladolid'

```

```

139         tipoRespuesta: 'textoCorto'
140         recursoAsociadoTexto: 'En este monumento puedes encontrar más de un
           estilo arquitectónico. Indica al menos dos distintos.'
141         respuestaEsperada: 'Renacentista y Barroco'
142         creadoPor: 'https://casuallearn.gsic.uva.es/researchers'
143         contexto: 'https://casuallearn.gsic.uva.es/context/
           Catedral_de_Valladolid/4723384/41652692'
144         latitud: 41.652692
145         longitud: -4.723384
146         - id: 'https://casuallearn.gsic.uva.es/Catedral_de_Valladolid/
           fotografiaEstilos'
147         comment: 'Catedral de Valladolid'
148         tipoRespuesta: 'multiplesFotografias'
149         recursoAsociadoTexto: 'En este monumento hay elementos de distintos
           estilos arquitectónicos: Renacentista y Barroco. Por cada estilo
           que detectes toma una fotografía que muestre un elemento
           arquitectónico para ejemplificar dicho estilo.'
150         creadoPor: 'https://casuallearn.gsic.uva.es/researchers'
151         contexto: 'https://casuallearn.gsic.uva.es/context/
           Catedral_de_Valladolid/4723384/41652692'
152         latitud: 41.652692
153         longitud: -4.723384
154
155 RespuestaEnviada:
156   type: object
157   required:
158     - idUsuario
159     - idTarea
160     - instanteInicio
161     - instanteFin
162   properties:
163     idUsuario:
164       type: string
165     idTarea:
166       type: string
167       example: 'https://casuallearn.gsic.uva.es/Catedral_de_Valladolid/
           fotografiaEstilos'
168     puntuacion:
169       type: integer
170       format: int32
171       minimum: 1
172       maximum: 5
173       description: 'Estrellas que el usuario ha dado a la tarea educativa
           como valoración.'
174       example: 5
175     respuestaTextual:
176       type: string
177       example: 'Creo que [...]'
178     numeroMedia:
179       type: integer
180       format: int32
181       description: 'Número de elementos multimedia que ha creado el usuario
           para generar la respuesta.'
182       example: 3
183     instanteInicio:
184       type: integer
185       format: int64
186       description: 'Instante en el que el usuario comenzó a interactuar con
           la tarea educativa (pantalla de descripción de la tarea).'
187       example: 100
188     instanteFin:
189       type: integer

```

```

190         format: int64
191         description: 'Instante en el que el usuario finalizó la tarea
192             educativa.'
193         example: 110
194     instanteModificacion:
195         type: integer
196         format: int64
197         description: 'Instante en el que finalizó la edición de una respuesta
198             .'
199         example: 222
200     publico:
201         type: boolean
202         default: true
203         description: 'Respuesta pública en el portafolio.'
204         example: true
205     example:
206         idUsuario: 'aAbBcCdDeE'
207         idTarea: 'https://casuallearn.gsic.uva.es/Catedral_de_Valladolid/
208             fotografiaEstilos'
209         respuestaTextual: 'Creo que [...]'
210         numeroMedia: 3
211         instanteInicio: 10
212         instanteFin: 37
213         publico: false
214
215 IdentificadorRespuesta:
216     type: object
217     properties:
218         idToken:
219             type: string
220             example: idEx
221             description: 'Identificador único.'
222
223 NuevoPortafolio:
224     type: object
225     properties:
226         idUsuario:
227             type: string
228             description: 'Identificador único del usuario.'
229         publico:
230             type: boolean
231             example: true
232             description: 'Indica si el portafolio puede consultarlo cualquier
233                 persona o nadie.'
234         retardado:
235             type: boolean
236             example: true
237             description: 'Indica si las respuestas se hacen públicas en el momento
238                 en el que se reciben en el servidor o si se establece un tiempo
239                 de retardo aleatorio hasta su publicación.'
240     required:
241     - publico
242     - retardado
243     example:
244         idUsuario: 'aAbBcCdDeE'
245         publico: true
246         retardado: true
247
248 paths:
249     /contextos:

```

```

246 get:
247   summary: 'Obtención de los POI de una tesela rectangular.'
248   parameters:
249     - in: query
250       name: 'norte'
251       description: 'Coordenada que limita el norte de la tesela.'
252       required: true
253       schema:
254         type: number
255         format: double
256         minimum: -90
257         maximum: 90
258         example: 41.66
259     - in: query
260       name: 'sur'
261       description: 'Coordenada que limita el sur de la tesela.'
262       required: true
263       schema:
264         type: number
265         format: double
266         minimum: -90
267         maximum: 90
268         example: 41.655
269     - in: query
270       name: 'este'
271       description: 'Coordenada que limita el este de la tesela.'
272       required: true
273       schema:
274         type: number
275         format: double
276         minimum: -180
277         maximum: 180
278         example: -4.72
279     - in: query
280       name: 'oeste'
281       description: 'Coordenada que limita el oeste de la tesela.'
282       required: true
283       schema:
284         type: number
285         format: double
286         minimum: -180
287         maximum: 180
288         example: -4.725
289     - in: query
290       name: 'idUserario'
291       description: 'Identificador del usuario en Casual Learn.'
292       required: false
293       schema:
294         type: string
295         default: ''
296   responses:
297     '200':
298       description: 'Vector de POI en formato JSON. Puede estar vacío.'
299       content:
300         application/json:
301           schema:
302             $ref: '#/components/schemas/POIArray'
303     '400':
304       description: 'El usuario no ha enviado alguno de los parámetros
305         solicitados o los ha enviado en un formato incorrecto.'
306   default:
307     description: 'Error interno del servidor.'

```

```

307     tags:
308         - POI
309
310 /tareas:
311     get:
312         summary: 'Obtención de las tareas educativas asociadas a un POI.'
313         parameters:
314             - in: query
315               name: contextos
316               required: true
317               schema:
318                 type: string
319                 example: 'https://casuallearn.gsic.uva.es/context/
320                       Catedral_de_Valladolid/4723384/41652692'
321             - in: query
322               name: 'idUsuario'
323               description: 'Identificador del usuario en Casual Learn'
324               required: false
325               schema:
326                 type: string
327                 default: ''
328         responses:
329             '200':
330                 description: 'Un vector de objetos JSON con las tareas. El vector
331                             puede estar vacío.'
332                 content:
333                     application/json:
334                         schema:
335                             $ref: '#/components/schemas/TaskArray'
336             '400':
337                 description: 'No se ha enviado un POI para obtener sus tareas
338                             educativas asociadas'
339                 default:
340                     description: 'Error interno del servidor'
341         tags:
342             - Tareas educativas
343
344 /tareasCompletadas:
345     post:
346         summary: 'Registro de una respuesta por parte de un usuario.'
347         requestBody:
348             required: true
349             content:
350                 application/json:
351                     schema:
352                         $ref: '#/components/schemas/RespuestaEnviada'
353         responses:
354             '201':
355                 description: 'La respuesta ha sido almacenada.'
356                 headers:
357                     Location:
358                         schema:
359                             type: string
360                             example: 'https://casuallearnapp.gsic.uva.es/app/
361                                   tareasCompletadas/idRespuestaEx'
362                 content:
363                     application/json:
364                         schema:
365                             $ref: '#/components/schemas/IdentificadorRespuesta'
366             '400':
367                 description: 'El cliente no ha enviado todos los parámetros necesarios
368                             o los ha enviado en un formato incorrecto.'

```

```

364         '401':
365             description: 'El usuario no está registrado en Firebase Authentication
366             ,
367         default:
368             description: 'Error interno del servidor'
369         tags:
370             - Respuestas
371 /tareasCompletadas/{idRespuesta}:
372     parameters:
373         - name: idRespuesta
374           in: path
375           required: true
376           description: 'Identificador único que se dio a la respuesta del usuario
377           ,
378           schema:
379             type: string
380     put:
381         summary: 'Edición de una respuesta.'
382         requestBody:
383             required: true
384             content:
385                 application/json:
386                     schema:
387                         $ref: '#/components/schemas/RespuestaEnviada'
388         responses:
389             '200':
390                 description: 'Respuesta editada almacenada.'
391                 content:
392                     application/json:
393                         schema:
394                             $ref: '#/components/schemas/IdentificadorRespuesta'
395             '400':
396                 description: 'El cliente no ha enviado todos los parámetros necesarios
397                 o los ha enviado en un formato incorrecto.'
398             '401':
399                 description: 'El usuario no está registrado en Firebase Authentication
400                 ,
401             '404':
402                 description: 'No existe ninguna respuesta con ese identificador'
403         default:
404             description: 'Error interno del servidor.'
405         tags:
406             - Respuestas
407 /portafolio:
408     post:
409         summary: 'Creación del portafolio de un usuario.'
410         requestBody:
411             content:
412                 application/json:
413                     schema:
414                         $ref: '#/components/schemas/NuevoPortafolio'
415         responses:
416             '201':
417                 description: 'El portafolio se ha creado.'
418                 headers:
419                     Location:
420                         schema:
421                             type: string
422                             example: https://casuallearnapp.gsic.uva.es/app/portafolio/
423                                 idPortafolioUsuario

```



```

421         content:
422             application/json:
423                 schema:
424                     $ref: '#/components/schemas/IdentificadorRespuesta'
425         '400':
426             description: 'El cliente no ha enviado todos los parámetros necesarios
427                 o estaban en un formato incorrecto.'
428         '401':
429             description: 'El identificador de usuario no existe en Firebase
430                 Authentication.'
431         default:
432             description: 'Error interno del servidor.'
433     tags:
434         - Portafolio
435 /portafolio/{idUsuario}:
436     parameters:
437         - name: idUsuario
438           in: path
439           required: true
440           description: 'Identificador único que se dio al portafolio del usuario.'
441           schema:
442             type: string
443     get:
444         summary: 'Obtención de una de las páginas del portafolio de un usuario.'
445         parameters:
446             - name: page
447               in: query
448               required: false
449               description: 'Página del portafolio que se desea consultar.'
450               schema:
451                 type: integer
452                 format: int32
453                 default: 0
454         responses:
455             '200':
456                 description: 'Se envía el portafolio del usuario con las respuestas
457                     que tenga. Se envía hasta tres respuestas por página. La informaci
458                     ón enviada de cada respuesta está incompleta.'
459             '400':
460                 description: 'La página solicitada no existe o el formato enviado para
461                     la selección de la página no es válido.'
462             '404':
463                 description: 'No se ha encontrado el portafolio.'
464         default:
465             description: 'Error interno del servidor.'
466     tags:
467         - Portafolio
468     put:
469         summary: 'Actualización de la visibilidad del portafolio y el retardo de
470             publicación.'
471         requestBody:
472             content:
473                 application/json:
474                     schema:
475                         $ref: '#/components/schemas/NuevoPortafolio'
476         responses:
477             '200':
478                 description: 'Ajustes del portafolio actualizados.'
479             headers:
480                 Location:
481                     schema:

```

```

477         type: string
478         example: https://casuallearnapp.gsic.uva.es/app/portafolio/
           idPortafolioUsuario
479     content:
480         application/json:
481             schema:
482                 $ref: '#/components/schemas/IdentificadorRespuesta'
483     '400':
484         description: 'Falta alguno de los parámetros o su formato es
           incorrecto.'
485     '401':
486         description: 'El usuario no se encuentra en Firebase Authentication.'
487     '404':
488         description: 'El portafolio no se había registrado previamente en el
           sistema.'
489     default:
490         description: 'Error interno del servidor.'
491     tags:
492     - Portafolio
493
494
495 /portafolio/{idUsuario}/{idRespuesta}:
496     parameters:
497     - name: idUsuario
498       in: path
499       required: true
500       description: 'Identificador único que se dio al portafolio del usuario.'
501       schema:
502         type: string
503     - name: idRespuesta
504       in: path
505       required: true
506       description: 'Identificador único que se dio a la respuesta del usuario
           .'
507       schema:
508         type: string
509     get:
510         summary: 'Obtención de toda la información de la respuesta de un usuario.'
511         responses:
512             '200':
513                 description: 'Se representa en una página toda la información que se
           disponga de la respuesta del usuario.'
514             '404':
515                 description: 'El portafolio o la respuesta no existe. Alguno de los
           elementos puede que no esté visible.'
516         default:
517             description: 'Error interno del servidor.'
518         tags:
519         - Portafolio

```

## Anexo B

# Plantillas del portafolio

El servidor de Casual Learn utilizará la extensión del *framework* Restlet Velocity para generar de manera dinámica las páginas del portafolio de cada uno de los usuarios del sistema. En la versión actual se dispone de tres plantillas: Listado B.1 si el usuario todavía no ha completado ninguna tarea educativa; Listado B.2 para crear el listado de respuestas de un usuario; y, por último, Listado B.3 para generar la página con la respuesta que ha proporcionado un usuario a una tarea de aprendizaje.

Listado B.1: Plantilla para la respuesta de un portafolio en el que el usuario todavía no haya registrado ninguna respuesta

```
1 <!DOCTYPE html >
2 <html >
3 <head >
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link rel="stylesheet" href="https://casuallearnapp.gsic.uva.es/bootstrap-4.5.2-
  dist/css/bootstrap.min.css">
7 <link rel="stylesheet" href="https://casuallearnapp.gsic.uva.es/css/estiloTarea.
  css">
8 <link rel="apple-touch-icon" sizes="57x57" href="https://casuallearnapp.gsic.uva
  .es/media/apple-icon-57x57.png">
9 <link rel="apple-touch-icon" sizes="60x60" href="https://casuallearnapp.gsic.uva
  .es/media/apple-icon-60x60.png">
10 <link rel="apple-touch-icon" sizes="72x72" href="https://casuallearnapp.gsic.uva
  .es/media/apple-icon-72x72.png">
11 <link rel="apple-touch-icon" sizes="76x76" href="https://casuallearnapp.gsic.uva
  .es/media/apple-icon-76x76.png">
12 <link rel="apple-touch-icon" sizes="114x114" href="https://casuallearnapp.gsic.
  uva.es/media/apple-icon-114x114.png">
13 <link rel="apple-touch-icon" sizes="120x120" href="https://casuallearnapp.gsic.
  uva.es/media/apple-icon-120x120.png">
14 <link rel="apple-touch-icon" sizes="144x144" href="https://casuallearnapp.gsic.
  uva.es/media/media/apple-icon-144x144.png">
15 <link rel="apple-touch-icon" sizes="152x152" href="https://casuallearnapp.gsic.
  uva.es/media/apple-icon-152x152.png">
16 <link rel="apple-touch-icon" sizes="180x180" href="https://casuallearnapp.gsic.
  uva.es/media/apple-icon-180x180.png">
17 <link rel="icon" type="image/png" sizes="192x192" href="https://casuallearnapp.
  gsic.uva.es/media/android-icon-192x192.png">
18 <link rel="icon" type="image/png" sizes="32x32" href="https://casuallearnapp.
  gsic.uva.es/media/favicon-32x32.png">
19 <link rel="icon" type="image/png" sizes="96x96" href="https://casuallearnapp.
  gsic.uva.es/media/favicon-96x96.png">
```

```

20 <link rel="icon" type="image/png" sizes="16x16" href="https://casuallearnapp.
    gsic.uva.es/media/favicon-16x16.png">
21 <link rel="manifest" href="https://casuallearnapp.gsic.uva.es/media/manifest.
    json">
22 <meta name="msapplication-TileColor" content="#ffffff">
23 <meta name="msapplication-TileImage" content="https://casuallearnapp.gsic.uva.es
    /media/ms-icon-144x144.png">
24 <meta name="theme-color" content="#ffffff">
25 <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js
    "></script>
26 <script src="https://casuallearnapp.gsic.uva.es/bootstrap-4.5.2-dist/js/
    bootstrap.min.js"></script>
27 <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300&display=
    swap" rel="stylesheet">
28 <title>Portafolio general del usuario</title>
29 </head>
30 <body>
31     <nav class="navbar navbar-expand-md" style="background-color: #666b6e;">
32         <span class="nav-text">Casual Learn</span>
33         <button class="navbar-toggler" type="button" data-toggle="collapse" data
            -target="#estadisticas" aria-controls="estadisticas" aria-expanded="
            false" aria-label="Toggle navigation" style="border: 0; padding: 0;
            font-size: 1.5em;">
34             <span class="navbar-text"></span>
35         </button>
36         <div class="collapse navbar-collapse" id="estadisticas">
37             <ul class="navbar-nav ml-auto">
38                 <li class="nav-text">Tareas: 0</li>
39                 <li class="nav-text"> 0</li>
40                 <li class="nav-text"> 0</li>
41                 <li class="nav-text"> 0</li>
42                 <li class="nav-text"> 0</li>
43                 <li class="nav-text"> 0</li>
44                 <li class="nav-text"> 0</li>
45                 <li class="nav-text"> 0</li>
46                 <li class="nav-text"> 0</li>
47             </ul>
48         </div>
49     </nav>
50
51     <div class="sinColor">
52         <div class="container contenedorDatos">
53             <h1>El portafolio del usuario no tiene ninguna respuesta.</h1>
54             <br>

```

```

55     <h3 style="text-align: justify;">Si acabas de activar el portafolio
      y llevas utilizando la aplicación un tiempo, puede que tengas
      que seleccionar qué tareas deseas hacer públicas. Lo puedes
      realizar desde la <i>app</i> de Casual Learn en el menú de la
      pantalla de Tarea completada, accesible desde la lista de tareas
      completadas. Desde el momento que hagas público el portafolios
      todas tus nuevas respuestas se harán públicas. Siempre podrás
      eliminar las respuestas individuales que no quieras que sean pú
      blicas desde la <i>app</i>. También podrás eliminar el
      portafolio cuando lo desees desde los ajustes de Casual Learn.</
      h3>
56     </div>
57 </div>
58 </body>
59 </html>

```

Listado B.2: Plantilla para la respuesta de un portafolio en el que el usuario tenga al menos una respuesta en el servidor

```

1 <!DOCTYPE html >
2 <html >
3 <head >
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link rel="stylesheet" href="https://casuallearnapp.gsic.uva.es/bootstrap-4.5.2-
  dist/css/bootstrap.min.css">
7 <link rel="stylesheet" href="https://casuallearnapp.gsic.uva.es/css/estiloTarea.
  css">
8 <link rel="apple-touch-icon" sizes="57x57" href="https://casuallearnapp.gsic.uva
  .es/media/apple-icon-57x57.png">
9 <link rel="apple-touch-icon" sizes="60x60" href="https://casuallearnapp.gsic.uva
  .es/media/apple-icon-60x60.png">
10 <link rel="apple-touch-icon" sizes="72x72" href="https://casuallearnapp.gsic.uva
  .es/media/apple-icon-72x72.png">
11 <link rel="apple-touch-icon" sizes="76x76" href="https://casuallearnapp.gsic.uva
  .es/media/apple-icon-76x76.png">
12 <link rel="apple-touch-icon" sizes="114x114" href="https://casuallearnapp.gsic.
  uva.es/media/apple-icon-114x114.png">
13 <link rel="apple-touch-icon" sizes="120x120" href="https://casuallearnapp.gsic.
  uva.es/media/apple-icon-120x120.png">
14 <link rel="apple-touch-icon" sizes="144x144" href="https://casuallearnapp.gsic.
  uva.es/media/apple-icon-144x144.png">
15 <link rel="apple-touch-icon" sizes="152x152" href="https://casuallearnapp.gsic.
  uva.es/media/apple-icon-152x152.png">
16 <link rel="apple-touch-icon" sizes="180x180" href="https://casuallearnapp.gsic.
  uva.es/media/apple-icon-180x180.png">
17 <link rel="icon" type="image/png" sizes="192x192" href="https://casuallearnapp.
  gsic.uva.es/media/android-icon-192x192.png">
18 <link rel="icon" type="image/png" sizes="32x32" href="https://casuallearnapp.
  gsic.uva.es/media/favicon-32x32.png">
19 <link rel="icon" type="image/png" sizes="96x96" href="https://casuallearnapp.
  gsic.uva.es/media/favicon-96x96.png">
20 <link rel="icon" type="image/png" sizes="16x16" href="https://casuallearnapp.
  gsic.uva.es/media/favicon-16x16.png">
21 <link rel="manifest" href="https://casuallearnapp.gsic.uva.es/media/manifest.
  json">
22 <meta name="msapplication-TileColor" content="#ffffff">
23 <meta name="msapplication-TileImage" content="https://casuallearnapp.gsic.uva.es
  /media/ms-icon-144x144.png">
24 <meta name="theme-color" content="#ffffff">
25 <!--<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>-->
26 <script src="https://casuallearnapp.gsic.uva.es/bootstrap-4.5.2-dist/js/

```

```

bootstrap.min.js"></script>
27 <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300&display=
swap" rel="stylesheet">
28 <title>Portafolio general del usuario</title>
29 </head>
30 <body>
31 <nav class="navbar navbar-expand-md" style="background-color: #666b6e;">
32 <span class="nav-text">Casual Learn</span>
33 #if($estadistica.total > 0)
34 <button class="navbar-toggler" type="button" data-toggle="collapse" data
-target="#estadisticas" aria-controls="estadisticas" aria-expanded="
false" aria-label="Toggle navigation" style="border: 0; padding: 0;
font-size: 1.5em;">
35 <span class="navbar-text"></span>
36 </button>
37 <div class="collapse navbar-collapse" id="estadisticas">
38 <ul class="navbar-nav ml-auto">
39 <li class="nav-text">Tareas: $estadistica.total</li>
40 <li class="nav-text"> $estadistica.sinRespuesta
</li>
41 <li class="nav-text"> $estadistica.preguntaCorta
</li>
42 <li class="nav-text"> $estadistica.preguntaLarga
</li>
43 <li class="nav-text"> $estadistica.imagen</li>
44 <li class="nav-text"> $estadistica.
imagenMultiple</li>
45 <li class="nav-text"> $estadistica.
preguntaImagen</li>
46 <li class="nav-text"> $estadistica.preguntaImágenesMúltiples</li>
47 <li class="nav-text"> $estadistica.video</li>
48 <li class="nav-text"> $estadistica.
preguntaVideo</li>
49 </ul>
50 </div>
51 #end
52 </nav>
53
54 #foreach ($area in $respuestas)
55 #if ( $foreach.count % 2 != 0 )
56 <div class="sinColor">
57 #else

```

```

58     <div class="color">
59 #end
60     <div class="container contenedorDatos">
61         <h1>${tarea.titulo}</h1>
62         <p>${tarea.textoPregunta}</p>
63     </div>
64
65     <div class="container respuestaUsuario">
66         <h6>${tarea.fecha}</h6>
67         <p>${tarea.textoUsuario}</p>
68     </div>
69     <div class="container masInfo">
70         <a href="${tarea.masInfo}">Más información</a>
71     </div>
72 </div>
73 #end
74
75 #if( $siguiente.tieneAnterior || $siguiente.tieneSiguiente )
76 <div class="container" style="margin-top: 10px;">
77     <ul class="pagination justify-content-center">
78         #if($siguiente.tieneAnterior)
79         <li class="page-item"><a class="page-link" href="$siguiente.
80             anteriores">ANTERIORES</a></li>
81         #end
82         #if($siguiente.tieneSiguiente)
83         <li class="page-item active"><a class="page-link" href="$siguiente.
84             siguientes">SIGUIENTES</a></li>
85         #end
86     </ul>
87 </div>
88 #end
</body>
</html>

```

Listado B.3: Plantilla para la respuesta de una solicitud de respuesta individual de un usuario

```

1 <!DOCTYPE html >
2 <html >
3 <head >
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link rel="stylesheet" href="https://casuallearnapp.gsic.uva.es/bootstrap-4.5.2-
7     dist/css/bootstrap.min.css">
8 <link rel="stylesheet" href="https://casuallearnapp.gsic.uva.es/css/estiloTarea.
9     css">
10 <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
11     integrity="sha512-
12     xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/
13     sMzMZ19scr4PsZChSR7A==" crossorigin="" />
14 <link rel="apple-touch-icon" sizes="57x57" href="https://casuallearnapp.gsic.uva
15     .es/media/apple-icon-57x57.png">
16 <link rel="apple-touch-icon" sizes="60x60" href="https://casuallearnapp.gsic.uva
17     .es/media/apple-icon-60x60.png">
18 <link rel="apple-touch-icon" sizes="72x72" href="https://casuallearnapp.gsic.uva
19     .es/media/apple-icon-72x72.png">
20 <link rel="apple-touch-icon" sizes="76x76" href="https://casuallearnapp.gsic.uva
21     .es/media/apple-icon-76x76.png">
22 <link rel="apple-touch-icon" sizes="114x114" href="https://casuallearnapp.gsic.
23     uva.es/media/apple-icon-114x114.png">
24 <link rel="apple-touch-icon" sizes="120x120" href="https://casuallearnapp.gsic.
25     uva.es/media/apple-icon-120x120.png">
26 <link rel="apple-touch-icon" sizes="144x144" href="https://casuallearnapp.gsic.

```

```

    uva.es/media/media/apple-icon-144x144.png">
16 <link rel="apple-touch-icon" sizes="152x152" href="https://casuallearnapp.gsic.
    uva.es/media/apple-icon-152x152.png">
17 <link rel="apple-touch-icon" sizes="180x180" href="https://casuallearnapp.gsic.
    uva.es/media/apple-icon-180x180.png">
18 <link rel="icon" type="image/png" sizes="192x192" href="https://casuallearnapp.
    gsic.uva.es/media/android-icon-192x192.png">
19 <link rel="icon" type="image/png" sizes="32x32" href="https://casuallearnapp.
    gsic.uva.es/media/favicon-32x32.png">
20 <link rel="icon" type="image/png" sizes="96x96" href="https://casuallearnapp.
    gsic.uva.es/media/favicon-96x96.png">
21 <link rel="icon" type="image/png" sizes="16x16" href="https://casuallearnapp.
    gsic.uva.es/media/favicon-16x16.png">
22 <link rel="manifest" href="https://casuallearnapp.gsic.uva.es/media/manifest.
    json">
23 <meta name="msapplication-TileColor" content="#ffffff">
24 <meta name="msapplication-TileImage" content="https://casuallearnapp.gsic.uva.es
    /media/ms-icon-144x144.png">
25 <meta name="theme-color" content="#ffffff">
26 <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300&display=
    swap" rel="stylesheet">
27 <title>Respuesta individual</title>
28 </head>
29 <body>
30 <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"
31     integrity="sha512-XQoYMqMTK8LvdxXYG3nZ[...]3ynxwA=="
32     crossorigin="">
33 </script>
34
35 <nav class="navbar navbar-expand-md" style="background-color: #666b6e;">
36     <span class="nav-text mr-auto">Casual Learn</span>
37     <span class="nav-text">
38         <a href="../$volver.general" style="color: white; text-decoration: none
39             ;">
40             Portafolio
41         </a>
42     </span>
43 </nav>
44 <div class="sinColor">
45 #if ( $tarea.muestra )
46 <div class ="container row text-center contenedorFotosCabecera" id="
47     contenedorImagen">
48     <div class="contenedorDosFotos">
49         
51         <a href="$tarea.licencia" alt="Enlace a la licencia de la imagen
52             representativa." target="_blank">
53             <span class="contenedorFotoSuper textoLicencia">$tarea.autorFoto</
54                 span>
55             
58         </a>
59     </div>
60 </div>
61 #end
62
63 <div class="container contenedorDatos">
64     <h1>$tarea.titulo</h1>
65     <div id="mapaTarea" class="mapaTarea"></div>
66     <p>$tarea.textoPregunta</p>
67 </div>

```



```

61
62 #if( $tarea.respuestaUser )
63     <div id="respuestaUsuario" class="container respuestaUsuario">
64         <h6>$tarea.fecha</h6>
65         <p>$tarea.textoUsuario</p>
66     </div>
67 #end
68 </div>
69 </div>
70
71 <script>
72     var latitud = $tarea.latitud;
73     var longitud = $tarea.longitud;
74     var mymap = L.map('mapaTarea',
75     {
76         zoomControl: false,
77         boxZoom: false,
78         scrollWheelZoom: false,
79         doubleClickZoom: false,
80         dragging: false,
81         touchZoom: false,
82         keyboard: false,
83         tap: false
84     }).setView([latitud, longitud], 15);
85     L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
86         attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">
87             OpenStreetMap</a> contributors'
88     }).addTo(mymap);
89     var iconoTarea = L.icon({
90         iconUrl: 'https://casuallearnapp.gsic.uva.es/media/iconoTarea.png',
91         iconSize: [48, 48]
92     });
93     var marker = L.marker([latitud, longitud], {icon: iconoTarea}).addTo(mymap);
94     marker.bindPopup("<a href='https://www.openstreetmap.org/?zoom=18&mlat=" +
95         latitud + "&mlon=" + longitud + "'>Abrir en OpenStreepMap</a><br><br><a
96         href='https://www.google.com/maps/search/?api=1&query=" + latitud + ","
97         + longitud + "'>Abrir en Google Maps</a>", {
98         closeButton: false
99     });
100 </script>
101 </body>
102 </html>

```