



Trabajo de Fin de Grado

**ULL-Navigation. Tecnología de
realidad virtual en entornos
universitarios**

Alejandro Hernández Padrón

La Laguna, 22 de mayo de 2019

D. **Francisco de Sande González**, con DNI nº 42.067.050-G profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

C E R T I F I C A

Que la presente memoria titulada:

“ULL-Navigation. Tecnología de realidad virtual en entornos universitarios”

ha sido realizada bajo su dirección por D. **Alejandro Hernández Padrón**, con DNI nº 42221533L

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 22 de mayo de 2019

Agradecimientos

Mis agradecimientos al profesor Francisco de Sande González por su labor como tutor de este proyecto y orientando este trabajo.

XXX

Licencia



- © Esta obra está bajo una licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 4.0
Internacional.

Resumen

Este documento constituye el trabajo de investigación del alumno durante el proceso de desarrollo de una aplicación para dispositivos móviles Android mediante el uso de técnicas de realidad virtual(AR) combinadas con técnicas de geolocalización.

Partimos de los conocimientos de programación en Java adquiridos en la asignatura: “Diseño arquitectónico y patrones” cursada en el itinerario de Ingeniería del Software. Esta asignatura, impartida en el tercer curso del Grado en Ingeniería Informática de la Universidad de La Laguna, ha sido la que ha sentado los fundamentos a partir de los cuáles se ha desarrollado la aplicación.

Durante este proyecto, el alumno ha conseguido adquirir independencia en su trabajo, visión y planificación realizando tareas de investigación, desarrollo y documentación, que han dado como resultado la obtención de conocimientos durante el proceso de trabajo.

También se ha investigado sobre las posibles tecnologías de realidad virtual presentes en la actualidad y su uso e implementación en sistemas Android.

Palabras clave: Aplicaciones Android, Java, dispositivos móviles, programación, Realidad Aumentada . . . XXX

Abstract

MI RESUMEN en Inglés

Para hacer un espaciado grande XXX

Keywords: *Application for Android, Java, mobile devices, programming, Augmented Reality ... XXX*

Índice general

Introducción	1
1. Objetivos	2
2. Herramientas y Tecnologías	3
2.1. Introducción	3
2.1.1. Android Studio	3
2.1.2. LaTex	4
2.1.3. Github	5
2.2. Tecnologías utilizadas	5
2.2.1. El Sistema Operativo Android	5
2.2.2. Realidad Aumentada	6
2.2.3. Node.js	12
2.2.4. MongoDB	14
2.2.5. Heroku	15
2.2.6. mLab	16
2.2.7. Google Maps	16
3. AR en entornos universitarios	18
3.1. Aplicaciones móviles en entornos universitarios	18
3.1.1. Descarga automática de material	18
4. La aplicación ULL-Navigation	19
4.1. Especificación de requisitos	19
4.2. Especificación detallada de los requisitos	19
4.3. Ventanas de la aplicación	21
5. Back-end	26
5.1. Base de datos	26
5.2. Configuración del servidor	26
5.2.1. Instalación	27

5.2.2. Implementación	28
Bibliografía	33

Índice de figuras

2.1.	Android Studio, un IDE flexible e intuitivo.	4
2.2.	Google Glass. Demostración de RA y su uso en deportes	7
2.3.	Marker-bases AR. Demostración de su funcionamiento	8
2.4.	RA Volcán. Ejemplo de uso de RA en Educación	10
2.5.	Node.js	12
2.6.	Node.js. Ejemplo de funcionamiento	13
2.7.	Node.js. Ventajas	14
2.8.	MongodDB	14
2.9.	Ventajas de MongoDB	16
4.1.	Ventanas de iniciales de <i>ULL-Navigation</i>	21
4.2.	Ventana <i>Inicio</i> y el Menú de <i>ULL-Navigation</i>	22
4.3.	Mapa ULL.	23
4.4.	Ventanas de <i>Todo los centros ULL</i> e <i>Información del centro de ULL-Navigation</i>	24
4.5.	Ventanas de <i>Configuración</i> e <i>Información de ULL-Navigation</i>	25
5.1.	ull-site. Ejemplo de un centro de la ULL en la base de datos.	31

Introducción

Este documento comprende el trabajo de investigación y desarrollo realizado por el alumno en la consecución de su Trabajo de Fin de Grado (TFG), con el que culminará sus estudios del Grado en Ingeniería Informática cursados en la Escuela Superior de Ingeniería y Tecnología (ESIT) de la Universidad de la Laguna (ULL).

Capítulo 1

Objetivos

Este TFG tiene los siguientes objetivos principales:

- Por un lado se pretende ampliar los conocimientos en tecnologías móviles en el sistema operativo *Android* [1] y en el desarrollo de aplicaciones para este sistema operativo.
- Otro objetivo presente en este TFG es que el alumno investigue y profundice en las técnicas y tecnologías de realidad virtual presentes en la actualidad.
- Por otro lado, también se pretende que el alumno se familiarice con el uso de herramientas de control de versiones utilizando *Github* [2] y de edición de textos técnicos utilizando *LaTeX* [3].
- Por último, tras las labores de investigación y recopilación de información correspondientes, se espera que el alumno aplique los conocimientos adquiridos para desarrollar una aplicación funcional que cubra las necesidades propuestas.

Capítulo 2

Herramientas y Tecnologías

Este capítulo tiene como objetivo presentar las distintas herramientas software y tecnologías empleadas por el alumno en el desarrollo de **ULL-Navigation**.

2.1. Introducción

A continuación se explicarán brevemente las distintas herramientas software utilizadas en el proyecto.

2.1.1. Android Studio

Android Studio[4] es el IDE[5](Entorno de Desarrollo Integrado) oficial para el desarrollo de aplicaciones en Android, basado en IntelliJ IDEA[6]. Android Studio ofrece una serie de funcionalidades que han facilitado a la desarrolladora numerosas tareas, entre las cuales podemos destacar:

- Un sistema de compilación basado en Gradle[7] que ha simplificado tanto la inserción de dependencias de las distintas librerías que se han tenido que utilizar, como la compilación de la aplicación.
- Un emulador rápido y fácil de utilizar que ha ayudado a visualizar las distintas pantallas durante el desarrollo aunque no ha sido de mucha utilidad para probar el funcionamiento al ser dependiente la app de la tecnología Bluetooth.
- La facilidad para publicar cambios a aplicaciones ya funcionando sin tener que eliminar y volver a crear un nuevo APK parando la app.
- Un sistema de visualización de las diferentes pantallas muy completo, con soporte visual para añadir componentes y cambiar atributos fácilmente.

- Un sistema de depuración, con una interfaz sencilla e intuitiva.



Figura 2.1: Android Studio, un IDE flexible e intuitivo.

Se ha utilizado este IDE frente a otros como Eclipse + ADT [8] debido a que en la actualidad es el IDE oficial con soporte, mejoras y actualizaciones constantes de Google a las nuevas versiones de Android. Se ha preferido aprender a utilizar este entorno con vistas al futuro.

2.1.2. LaTex

LaTeX es un sistema de composición de textos, orientado a la creación de documentos que presenten una alta calidad tipográfica. Por sus características y posibilidades, es usado especialmente en la generación de artículos y publicaciones científicas que incluyen, entre otros elementos, expresiones matemáticas, gráficos o figuras.

LaTeX está formado por un gran conjunto de macros de TeX, escrito por Leslie Lamport en 1984, con la intención de facilitar el uso del lenguaje de composición tipográfica, creado por Donald Knuth. LaTeX es software libre bajo licencia LPPL.

Se ha decidido utilizar este sistema debido al carácter profesional que aporta a los documentos. Ha sido una buena oportunidad para aprender a usar un sistema de composición de texto como este, ya que en un futuro puede ser beneficioso el saber manejar esta herramienta.

Si bien es cierto, que el uso de esta herramienta frente a otros editores más familiares ha sido algo tedioso en el inicio, es verdad que una vez acostumbrada a su uso ha resultado ser muy eficaz.

2.1.3. Github

GitHub[2] es una forja (plataforma de desarrollo colaborativo) para alojar proyectos que utiliza el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por GitHub, Inc. (anteriormente conocida como Logical Awesome). Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

Se ha decidido crear un repositorio en esta plataforma para poder llevar un control y una trazabilidad del proyecto. El tutor y el alumno han trabajado en este repositorio de manera conjunta. En el caso del tutor, principalmente para revisar el seguimiento semanal y llevar un control de las tareas. En el caso del alumno, para tener un repositorio donde subir los distintos elementos que se han ido generando a lo largo del trabajo. Aparte de este repositorio, también se ha abierto un segundo repositorio [9] asociado a la oficina del software libre (OSL) para subir el código una vez terminado como parte del programa de apoyo a trabajos finales libres (PATFL) [10] de la ULL.

Mediante el uso de este repositorio, el alumno ha conseguido ampliar sus conocimientos en Git y familiarizarse con la interfaz de GitHub. Previamente se había utilizado como repositorios GitLab, SVN y RTC en otros proyectos, por lo que no ha sido una complicación mayor utilizar este sistema.

2.2. Tecnologías utilizadas

A continuación se revisan las distintas tecnologías utilizadas en el desarrollo de la aplicación.

2.2.1. El Sistema Operativo Android

Android es un sistema operativo que emplea Linux en la interfaz del hardware. Los componentes del SO subyacentes se codifican en C o C++ pero las aplicaciones se desarrollan en Java. De esta manera Android asegura una amplia operatividad en una gran variedad de dispositivos debido a dos hechos: la interfaz en Linux ofrece gran potencia y funcionalidad para aprovechar el hardware, mientras que el desarrollo de las aplicaciones en Java permite que Android sea accesible para un gran número de programadores conocedores del código.

Este SO fue diseñado principalmente para dispositivos móviles con pantalla táctil: smartphones, tablets y otros dispositivos como televisores o automóviles. Fue desarrollado inicialmente por Android Inc., empresa que fue respaldada económicamente por Google y más tarde adquirida por esta misma empresa.

Actualmente tiene una gran comunidad de desarrolladores creando aplicaciones para extender la funcionalidad de los dispositivos. A fecha de hoy existen más de un millón de aplicaciones disponibles para la tienda oficial de Apps de Android, Google Play[11] sin tener en cuenta las aplicaciones de otras tiendas no oficiales, como por ejemplo, la tienda de aplicaciones de Samsung Apps[12].

2.2.2. Realidad Aumentada

La Realidad Aumentada(RA) o Augmented Reality(AR) en inglés, es el término que se usa para definir la visión de un entorno físico del mundo real, a través de un dispositivo tecnológico. Este dispositivo, nos permite expandir nuestro mundo físico añadiendo capas de información digital generadas por un computador, como pueden ser imágenes, sonidos y videos a nuestra visión del entorno en tiempo real.

RA esta cambiando la manera en la que sus usuarios pueden ver el mundo, actualmente es una tecnología que se encuentra en auge debido a su enorme potencial, del cual podremos ver sus aplicaciones en nuestra vida cotidiana en un futuro no muy lejano.

Sus posibles aplicaciones no tienen límites, pueden llegar desde reconocer plantas e incluso monumentos y mostrar información sobre lo que se está viendo, hasta añadir información en tiempo real en una operación a un paciente, comprobar como queda un mueble en tu salón o sus aplicaciones para realizar videojuegos, como podemos ver con el reciente éxito de Pokemon Go!.

A continuación se explicará en detalle:

- ¿Como funciona esta tecnología?
- Tipos de Realidad Aumentada
- Diferencias entre Realidad Aumentada y Realidad Virtual
- Realidad Mixta
- Futuro y usos de la Realidad Aumentada
- Integración de Realidad Aumentada en Android Studio

¿Como funciona esta tecnología?

LaRA necesita de un dispositivo de visualización en el que mostrar esta unión del entorno real junto con la información digital. Esta unión puede ser visualizado en múltiples dispositivos: pantallas, gafas, dispositivos portátiles, smartphones, etc.

Además de estos sistemas de visualización, necesitaríamos de un sistema de computación que realice los cálculos y reciba los datos provenientes de múltiples sensores, es decir: una CPU, una GPU, RAM, GPS, WIFI, bluetooth, acelerómetro, giroscopio, cámara, etc. Gracias a estos elementos el sistema puede reconocer el entorno real[13].

Todo esto necesita una parte software. El software en una primera parte deberá de reconocer el terreno, ubicaciones, objetos e imágenes, mediante las imágenes obtenidas de la cámara y los datos de los sensores. Este proceso de transformación de diferentes conjuntos de datos a un sistema de coordenadas se llama registro de la imagen.

Posteriormente el software deberá reestructurar el mundo real en función del registro de imágenes, añadiendo y combinando la información correspondiente al entorno para generar nuestra imagen de RA. Existen múltiples maneras en las que se puede reestructurar este mundo.



Figura 2.2: Google Glass. Demostración de RA y su uso en deportes

Tipos de Realidad Aumentada

Existen hoy en día cuatro tipos de RA:

- Marker-bases AR. Se basa en el reconocimiento de imágenes conocidas como marker o marcadores. Los marker son imágenes distintivas que son reconocidas por nuestro dispositivo fácilmente ya que contienen puntos visuales únicos. Un buen ejemplo de este tipo de imágenes son los conocidos código QR[14], también tiene usos en el reconocimiento de textos para su traducción o para crear simulaciones de objetos 3D o arquitecturas sin llegar a construirlas de forma física. RR
- Markerless AR. Corresponde a la RA que recoge los datos de su posición, orientación para mostrar la información correspondiente a ese área. Estos

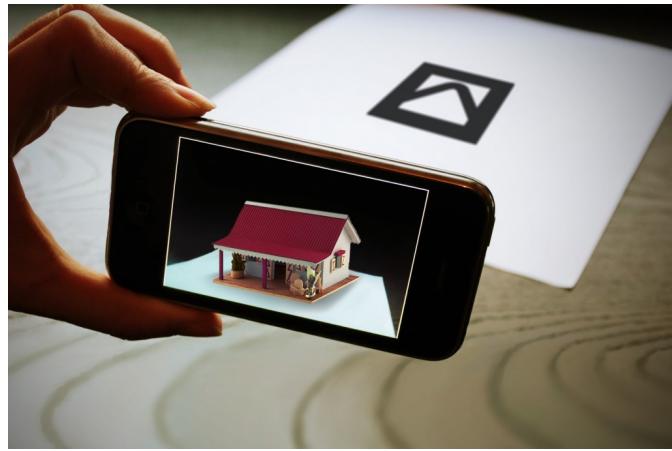


Figura 2.3: Marker-bases AR. Demostración de su funcionamiento

sistemas utilizan los datos obtenidos de la cámara, GPS, brújula, giroscopio y acelerómetro para establecer la ubicación del usuario. Utiliza técnicas para reconocer terreno o ambientes para calcular la posición y orientación de la cámara. Con este tipo de tecnología podemos por ejemplo probar un mueble en nuestro salón antes de llegar a comprarlo.

- Location-based AR. Es un tipo de RA parecida a la anterior, es decir no utiliza marcadores pero está centrada más en los cálculos de la posición y geolocalización. Este tipo de RA puede proveer de ayuda viajeros que necesiten una mano para moverse por la ciudad, ya que mediante el reconocimiento de su ubicación y la orientación pueden mostrarles la ruta para llegar a su destino o mostrarle información de los puntos de interés que les rodean.
- Projection-based AR. Esta modo de RA consiste en la proyección de luz en superficies y objetos en el mundo real. Existen muchos usos interesantes de esta tecnología, como aplicaciones para el uso de teclados virtuales proyectados que reconozcan cuando una "tecla" es pulsada. Esta proyección también se puede hacer en medio del aire con ayuda de la tecnología de láseres de plasma.
- Superimposition-based AR. Esta tecnología reemplaza la imagen original por una de RA, de forma completa o parcial. El reconocimiento de objetos juega un papel fundamental en este tipo de tecnología. Tiene gran utilidad el campo de la medicina, por ejemplo un doctor podría examinar a un paciente mientras ve la imagen de RA que se creó a partir de una visión de rayos X y la imagen real de paciente, así puede ver y entender mejor el daño en

un hueso.

Diferencias entre Realidad Aumentada y Realidad Virtual

En los últimos años la Realidad Virtual(RV)[15] o Virtual Reality(VR) en inglés y la Realidad Aumentada han empezado a recibir mucha más atención. Llevando a desarrolladores llevar a cabo integraciones de ambas tecnologías en numerosas industrias.

De acuerdo a un análisis de expertos, la RV iba tener el liderato por 2018 como tecnología pionera, sin embargo la RA va tener mucha más importancia y a largo plazo, llegando a formar parte de nuestro día a día.

Realidad Virtual es un entorno de escenas u objetos de apariencia real. Aleja al usuario del entorno real y da la sensación de estar inmerso en él. Dicho entorno es contemplado por el usuario a través de un dispositivo conocido como gafas o casco de realidad virtual. Este puede ir acompañado de otros dispositivos, como guantes o trajes especiales, que permiten una mayor interacción con el entorno así como la percepción de diferentes estímulos que intensifican la sensación de realidad.

La Realidad Aumentada no aísla al usuario del mundo exterior, sino que lleva al mundo real objetos no existentes al mundo real mediante la superposición de imágenes en tiempo real.

Realidad Mixta

Existe otro tipo de tecnología que nace de la unión de Realidad Aumentada y la Realidad Virtual, la Realidad Mixta(RM)[16]. Es un tipo de realidad similar a la Realidad Aumentada pero con una idea más ambiciosa de mezclar lo real con lo virtual. La RM es mucho más inmersiva que la realidad aumentada y requiere de mucho más poder de procesamiento.

Un ejemplo del uso de esta tecnología puede ser el mejorar los métodos de aprendizaje de los estudiantes, mediante simulaciones de tareas construidas virtualmente en un entorno real.

Futuro y usos de la Realidad Aumentada

Actualmente la Realidad Aumentada la tenemos más mano que en años anteriores. Dispositivos como los smartphones ya nos traen y nos enseñan las primeras muestras de esta tecnología, la cual está en sus primeros años de desarrollo pero ya se puede ver el potencial y la enorme importancia que va a tener en un futuro.

Los usos actuales de esta tecnología se acercan a todo los sectores conocidos:

- Realidad Aumentada en Educación. La llegada de la RA afectará a los

procesos convencionales de aprendizaje. La RA tiene la capacidad de cambiar el horario y el lugar en el que se estudia y la posibilidad de introducir nuevos métodos de enseñanza.

Actualmente gran parte de la población joven tiene un smartphone el cuál es un medio hábil para la RA. Por lo tanto se dan unas condiciones idóneas para que la RA profundice en el campo de la Educación y se hagan más descubrimientos ya que cada estudiante va a tener un dispositivo a mano capaz de reproducir la RA, lo cual puede ayudar a los estudiantes a tener contenidos mas accesibles sobre cualquier asignatura o conseguir que información compleja sea más fácil de entender. Un ejemplo claro sería la creación de libros interactivos que al ser apuntados con la cámara del móvil muestren el funcionamiento en 3D de un volcán o del latido de un corazón.

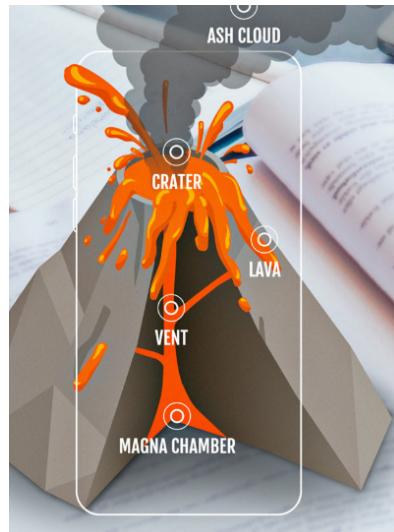


Figura 2.4: RA Volcán. Ejemplo de uso de RA en Educación

- Realidad Aumentada en los Videojuegos. Sin duda uno de los sectores en lo que más interés se tiene para desarrollar esta tecnología y quizás donde más se haya avanzado en realidad aumentada. Todas las grandes empresas de este sector tienen ya potentes desarrollos y lanzamientos de videojuegos que combinan la realidad física con la virtual, con múltiples posibilidades de personalización de cada juego. Un claro ejemplo es el éxito de Pokemón Go![17], una app para smartphones que utiliza técnicas de RA basadas en la localización, el cual a través de la cámara del dispositivo y modelos 3D para representar a los personajes de la saga Pokemón[18] los cuales se encuentran escondidos en ubicaciones del mundo real.

El desarrollo en este sector promete. Tras los beneficios económicos obtenidos con el juego de Pokemón Go!, muchas empresas han dado el salto a este sector. A medida que la potencia de los smartphones va creciendo cada año facilita la integración del RA. En un futuro esta tecnología revolucionará todo el sector de los videojuegos y cambiará la manera en la que interactuamos con ellos.

- Realidad Aumentada en la Industria. El desarrollo de apps de Realidad Aumentada en el ámbito industrial también está creciendo, ya que ayudan a mejorar la productividad de los ciclos de trabajo. Por ejemplo, algunas compañías están desarrollando aplicaciones que ayuden a los trabajadores de una cadena de montaje. De este modo, los empleados podrán obtener información adicional sobre las acciones que llevan a cabo. Este mismo sistema también se puede implementar en las reparaciones de vehículos o maquinaria industrial, ya que la app puede mostrarnos toda clase de avisos sobre las piezas deterioradas. Incluso nos puede llegar a mostrar contenido visual en 3D sobre como llevar a cabo la reparación o sustitución de esos elementos.
- Realidad Aumentada en Turismo. El negocio del turismo siempre ha intentado estar al día con la nueva tecnología para poder ofrecer nuevos servicios, formas de publicitarse, transporte y actividades de ocio. Por eso no es raro que la RA se haya hecho un hueco en este sector debido al potencial que tiene para mejorar la experiencia de los turistas.

Posibles usos en este sector son la puesta de guías turísticas que faciliten a los usuarios de estas moverse por la ciudad, señalando puntos de interés, datos históricos, restaurantes, hoteles, etc. El mismo concepto podría aplicarse en museos y zoos. Otro uso interesante sería el de romper la barrera del lenguaje gracias a la traducción inmediata de señales, textos y anuncios, al idioma del turista. Para los Juegos Olímpicos de Tokio 2020 se espera tener esta tecnología preparada para poder traducir en tiempo real todos los textos, señales y anuncios, a los visitantes, a través de nuestros móviles.

- Realidad Aumentada en Medicina. En cuanto a la Medicina, es interesante ver como avanza la tecnología en este campo pero a su vez es bastante difícil de prever que tipo de ventajas nos ofrecerá. Pero sin duda este sector apunta prometedor para mejorar nuestra calidad de vida y salud.

Los principales usos que se plantean se encuentran enfocados principalmente en los quirófanos, en los que el especialista o cirujano monte una especie gafas-pantalla que le permita realizar la operación sin la necesidad de apartar la vista del paciente para consultar información o ir monitorizando la operación.

Esto se traduce en operaciones mas rápidas y seguras sin que el cirujano se despiste. A su vez esto podría mostrar información anatómica sobre el paciente en tiempo real, es decir gracias a algoritmos de inteligencia artificial permitir identificar nervios, venas mayores y huesos, y que esto sean marcados con un distinto color, facilitando mucho las labores de los médicos.

2.2.3. Node.js

Node.js [19] es una librería y entorno de ejecución de E/S dirigida por eventos y por lo tanto asíncrona que se ejecuta sobre el intérprete de JavaScript creado por Google llamado Chrome V8[20]. Node.js es un entorno de JavaScript del lado del servidor, basado en eventos. Utiliza el motor de Chrome V8 motor permite a Node un entorno de ejecución que compila y ejecuta JavaScript a velocidades increíbles.



Figura 2.5: Node.js

¿Cómo funciona?

Node.js fue desarrollado con el objetivo que fuera un sistema escalable y que tuviera la consistencia de generar un elevado número de conexiones de forma simultanea con el servidor. La mayoría de las tecnologías que trabajan desde el lado del servidor tienden a accionar las peticiones de forma aislada y mediante hilos independientes. Esto se traduce en que a mayor número de solicitudes mayor es la cantidad de recursos necesarios para responderlas. Node.js se ha desarrollado para optimizar la gestión de estas solicitudes.

La solución que propone Node.js se basa en el tratamiento de la solicitudes de forma unificada en un único hilo complementado con un bucle de eventos (Event Loop[21]) de tipo asíncrono. De este modo cada petición que se recibe se trata como un evento y pertenecen a este único bucle. Este nuevo replanteamiento proporciona un lenguaje con una capacidad para gestionar una gran cantidad de solicitudes y conexiones con la máxima eficiencia.

Otro de sus puntos fuertes es su gestor de paquetes Node Package Manager (NPM). Este gestor de acceso en un enorme conjunto de librerías. Gracias a este

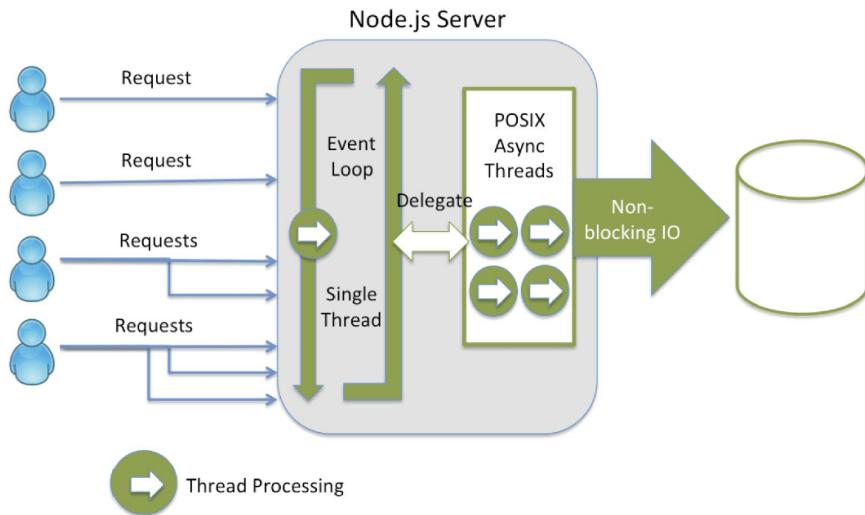


Figura 2.6: Node.js. Ejemplo de funcionamiento

gestor se pueden instalar paquetes, módulos y agregar dependencias de manera simple.

¿Por qué Node.js

- Fácil de integrar e instalar en cualquier servidor, sin importar el sistema operativo.
- JavaScript como base. Un lenguaje fácil de entender y aprender por cualquier programador.
- Gran rendimiento, permite generar a su vez arquitecturas potentes y sólidas.
- Su alta capacidad de escalabilidad. La creación de aplicaciones web con la capacidad de atender muchos miles de solicitudes en un único servidor de forma simultánea, sin la necesidad de tener que incrementar las infraestructuras de servidores.
- No sólo el funcionamiento de las aplicaciones resulta mucho más ágil y potente, sino que además el proceso de desarrollo y programación también resulta mucho más liviano y rápido.



Figura 2.7: Node.js. Ventajas

2.2.4. MongoDB

Descripción

MongoDB [22] es un sistema de base de datos multiplataforma NoSQL [23] orientado a documentos. Esta es de esquema libre, es decir, en un mismo documento podemos tener entradas o registros con diferentes atributos unos de otros sin tener que repetirse un registro a otro. Esto difiere de las tablas de bases de datos relacionales.



Figura 2.8: MongoDB

¿Cómo funciona?

Un registro en MongoDB es un documento, cuya estructura de datos se compone por el par campo y valor. Utilizan un formato para guardar estas estructuras que se llama BSON [24] también llamada JSON Binario. Este formato tiene una ventaja pese a que puede ocupar más espacio de lo que lo haría el formato JSON. BSON guarda de forma explícita las longitudes de los campos, los índices de los arrays,

y demás información útil para el escaneo de datos y así agilizar las búsquedas en estos documentos.

Estos documentos que a su vez incorporan una clave primaria como identificador, son la unidad básica de datos en MongoDB. Las colecciones contienen un conjunto de documentos y funciones equivalentes a las de las bases de datos relacionales. Estas colecciones pueden tener cualquier tipo de datos.

Beneficios

Dentro de sus cualidades mas destacables encontramos:

- El hecho de que sea una base de datos NoSQL, significa que no necesita esquemas predefinidos y puede guardar cualquier tipo de datos. Esto otorga una increíble flexibilidad para crear cualquier tipo de campos que se necesiten en un documento, permitiendo mayor escalabilidad en MongoDB comparado a las bases de datos tradicionales.
- A su vez el uso de documentos para guardar la información hace que sea mucho más fácil de mapear estos datos en los lenguajes nativos de programación.
- Es fácil de acceder a los documentos si lo indexamos. En MongoDb este indexado provee de rápidos tiempos de respuesta a las peticiones, esto le permite ser hasta 100 veces más rápida que los modelos de bases de datos relacionales.
- MongoDB es una base de datos distribuida, por lo que es fácil de usar y proporciona una elevada disponibilidad, escalabilidad horizontal y distribución geográfica.

2.2.5. Heroku

Heroku es una plataforma como servicio de computación (PaaS[25]) en la nube que soporta distintos lenguajes de programación.

Heroku es un plataforma en la nube que permite a desarrolladores y compañías construir, proporcionar, monitorizar y escalar aplicaciones. Es una forma sencilla de montar una infraestructura en la nube.

Heroku es conocida por correr las aplicaciones en Dynos. Un dyno es simplemente un ordenadores virtual que pueden ser encendido y o apagado en función al tamaño y requisitos de tu aplicación.

A cada uno de estos Dynos se le puede configurar mas espacio o capacidad de procesamiento, o se pueden añadir mas Dynos si tu aplicación lo necesita. Heroku a cada més realiza una cobro en función de los Dynos que tengas contratados.

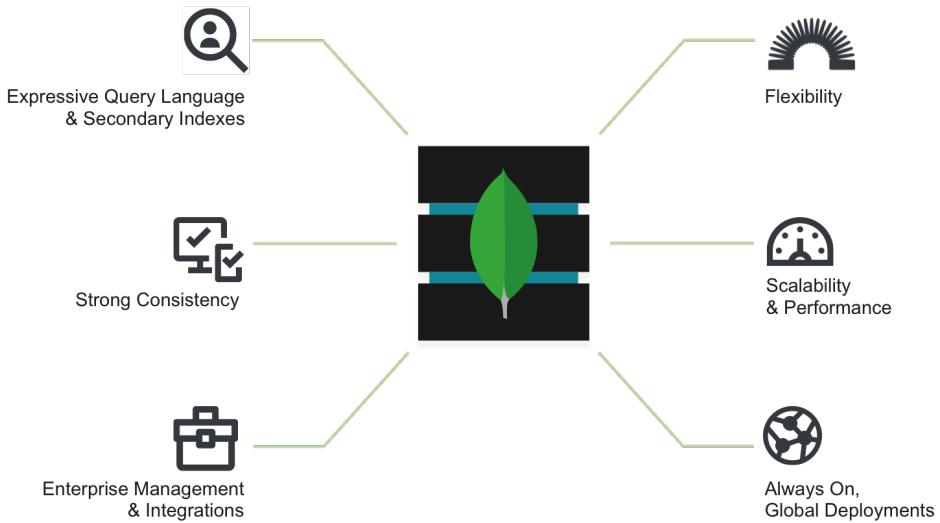


Figura 2.9: Ventajas de MongoDB

Se ha optado por utilizar el Dyno gratis que ofrece Heroku a cada repositorio. El cual tienen un límite de 1000 horas activas al mes y se pone en suspensión a los 30 minutos sin recibir tráfico. Una vez que vuelve a recibir tráfico se activa de nuevo.

2.2.6. mLab

mLab es un servicio de base de datos en la nube totalmente gestionado que ofrece aprovisionamiento y escalado automáticos de las bases de datos MongoDB, copia de seguridad y recuperación, monitoreo, herramientas de administración basadas en web y soporte experto. La plataforma de base de datos como servicio de mLab corre sus máquinas en proveedores de servicios en la nube como AWS[26], Azure[27] y Google Cloud[28].

En esta plataforma se ubicará la base de datos de nuestra aplicación. Dado las facilidades de acceso a ella e integración con el servidor de nuestra aplicación en node.js, es una opción simple, fácil de manejar y con posibilidad de escalar.

2.2.7. Google Maps

La API de Google Maps [29] fue publicada para Android en 2008 y en 2012 para IOS. Esta API permite utilizar mapas basados en datos de Google Maps en una aplicación Android, y además ofrece métodos para personalizar el mapa:

- Creación de marcadores, polígonos y superposiciones sobre el mapa para resaltar puntos o zonas.

- Permite cambiar la vista del usuario de modo que se muestre un área del mapa en particular.
- Ofrece la posibilidad de elegir el tipo de mapa: De carreteras, satélite, híbrido (mezcla de carretera y satélite) y de terreno.

Con esta API fácil de integrar y con soporte de Google, es la mejor opción para tener un mapa funcional y que nos permita ubicarnos en una aplicación Android.

Capítulo 3

AR en entornos universitarios

En este capítulo ...

3.1. Aplicaciones móviles en entornos universitarios

- XXX
- XXX

3.1.1. Descarga automática de material

Capítulo 4

La aplicación ULL-Navigation

En este capítulo explicaremos la aplicación ULL-Navigation utilizando la especificación de requisitos de la misma y explicando su funcionamiento.

4.1. Especificación de requisitos

Nos dispondremos a exponer los requisitos de la presente aplicación propuesta para el desarrollo de este TFG.

Se trata de una aplicación móviles, más concretamente a aquellos dispositivos que utilizan Android como sistema operativo. Es una aplicación diseñada para los estudiantes de la Universidad de La Laguna (ULL) la cual les permita ubicarse, detectar y reconocer los centros y edificios pertenecientes a la universidad mediante técnicas de Realidad Aumentada basadas en la geolocalización.

Los requisitos principales de la aplicación son:

- La aplicación se desarrollará para dispositivos con Android. Se utilizará Android Studio como entorno de desarrollo integrado(IDE) para su desarrollo.
- Se implementarán técnicas de Realidad Aumentada basadas en la geolocalización para mostrar al usuario el centro universitario al cuál apunta con la cámara.
- Los centros de la Universidad de La Laguna junto a su información correspondiente estarán ubicadas en un base datos en la nube y a su vez el servidor que se comunique con esta deberá estar en la nube.

4.2. Especificación detallada de los requisitos

A continuación se expondrán de forma más detallada cada uno de los requisitos de la aplicación.

La aplicación se iniciará una Splash Screen[30] o pantalla de inicio con el logo de la Universidad de La Laguna. Esta pantalla dará paso a una ventana de login.

Para poder utilizar la aplicación el usuario a ser alumno de la Universidad de La Laguna y poseer su respectiva cuenta de Google con el correo institucional de la universidad con el formato “*aluxxxxxxxxxx@ull.edu.es*”. Sin ella no se podrá acceder a la aplicación. Además se podrá cerrar la sesión de esta cuenta.

Una vez logueado accederemos a venta de inicio en la que aparecerá un acceso directo a las ventanas de *Mapa ULL* y *Navegación en modo RA* que explicaremos más adelante. Además dispondrá de los accesos directos a enlaces de interés de Universidad de La Laguna que se abrirán en un navegador externo.

Una vez dentro de la aplicación tendremos menú para movernos por las diferentes ventanas. Se utilizará un menú deslizante lateral o *Navigation Drawer*[31] ubicado en la parte superior izquierda de la aplicación. Este menú deberá ser simple e intuitivo.

Como accesos en este menú disponemos de las siguientes ventanas:

- **Mapa ULL:** Esta ventana contendrá un mapa de la universidad con todos los centros en la base de datos.
- **Navegación en modo RA:** En esta ventana mediante el uso de la cámara se identificarán los centros universitarios a los que esta apunte y permitirá mostrar una pestaña con información detallada del los mismos.
- **Centros de la ULL:** Contiene todas la ubicaciones y centros de la Universidad de La Laguna y permitirá la búsqueda de los mismos.
- **Configuración:** Permitirá acceder a los ajustes de la aplicación.
- **Cerrar sesión:** Cerrará la sesión actual y nos devolverá a la ventana de login
- **Info:** Información de aplicación así como su autor.

Cada centro de la universidad tendrá una ficha de información que será accesible desde una ventana de la aplicación con la siguiente información:

- **Id:** Con él identificaremos el centro.
- **Nombre:** Nombre oficial del centro
- **Ubicación:** La ubicación exacta en la que se encuentra el centro.
- **Descripción:** Descripción del centro con el objetivo y actividades que se desarrollan en él.

- **Imagen:** Imagen del centro.
 - **Lista de enlaces de interés:** Una lista con los enlaces a las instituciones, servicios, departamentos y grados que se imparten en este centro.

Esta información estará guardada en una base de datos en la nube. Para acceder a ésta se dispondrá de un servidor en la nube que conecte con la base de datos y envíe la información a la aplicación.

4.3. Ventanas de la aplicación

Nada más iniciar la aplicación nos encontramos con dos ventanas:

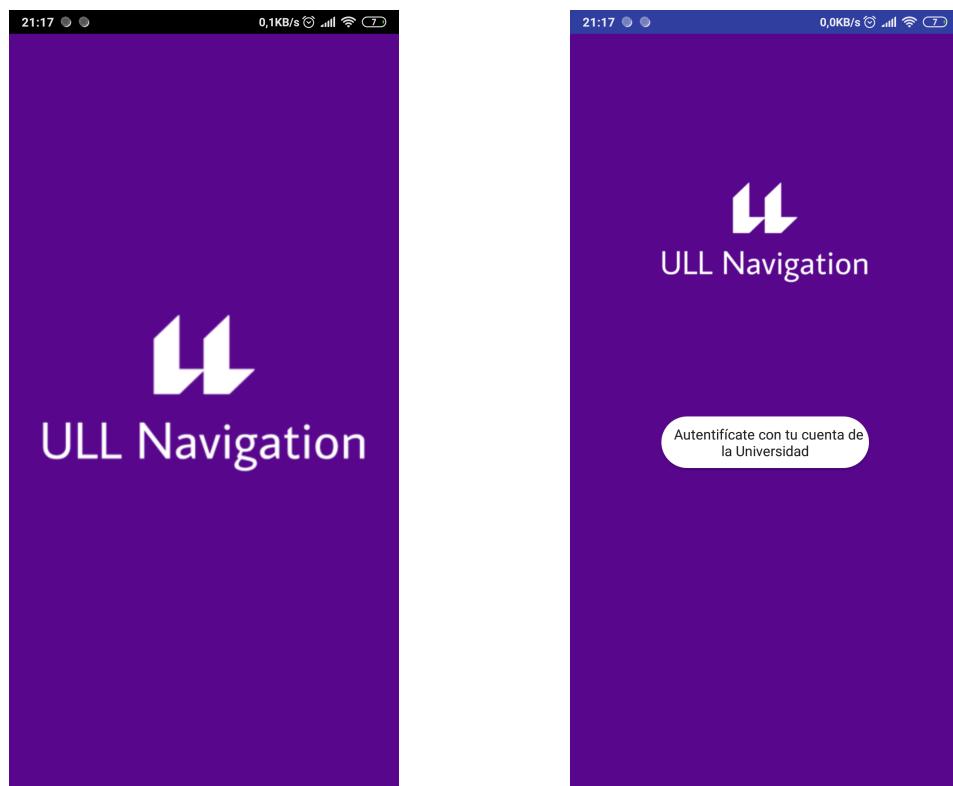


Figura 4.1: Ventanas de iniciales de *ULL-Navigation*.

La primera imagen (vease Figura 4.1a) es la pantalla de inicio que se aparece cuando ejecutamos la aplicación. Tras unos segundos, se carga la ventana del login(vease Figura 4.1b). En esta tendremos que tener una cuenta de Google de la

ULL. Una vez pinchamos en el botón del medio se nos abre una ventana de diálogo para que pongamos nuestra cuenta.

Cuando nos hayamos logueado con éxito se nos abrirá la venta de *Inicio* (vease Figura 4.2a) en la que tendremos una lista de accesos directos a las funcionalidad principales de la aplicación, como són *Navegación en modo RA*, *Mapa ULL* y enlaces a sitios web de la ULL.

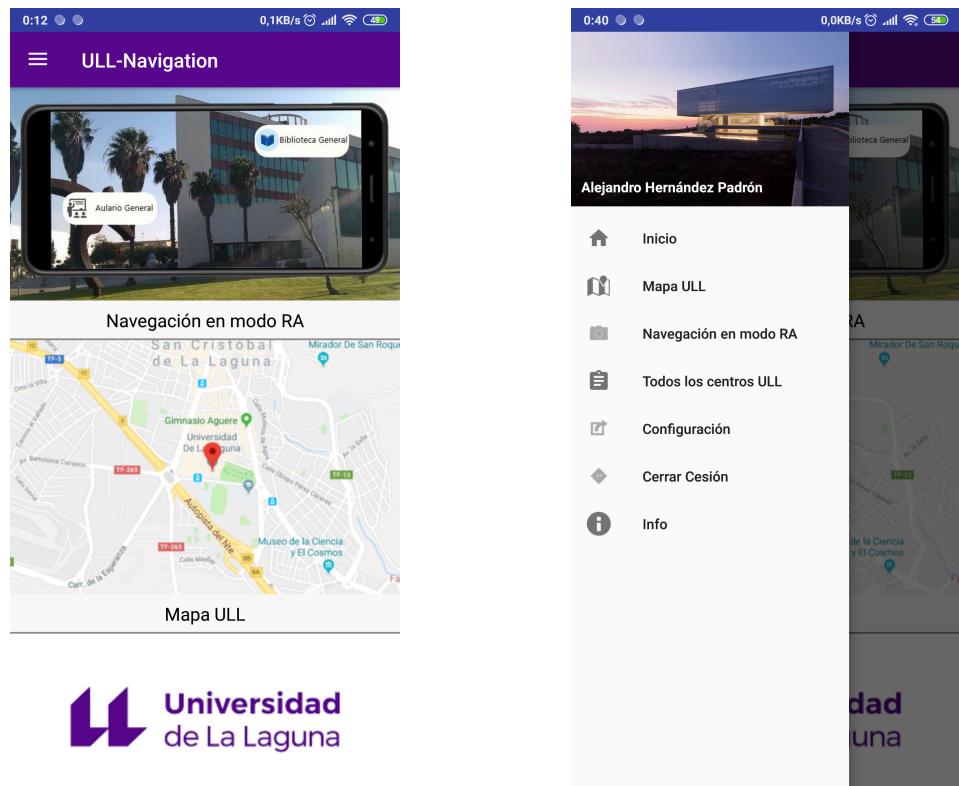


Figura 4.2: Ventana *Inicio* y el Menú de *ULL-Navigation*.

En la esquina superior izquierda de aplicación tenemos el acceso al menú *Navigation Drawer*. Si lo pulsamos se nos desplegará el menú que nos permite movernos por las distintas ventanas de la aplicación (vease Figura 4.2b).

Si nos movemos a la ventana de *Mapa ULL* (vease Figura 4.3) veremos el mapa de la API de Google Maps. En este mapa nos aparecerán en pinos azules las ubicaciones con sus respectivos nombres de la ULL que están guardadas en la base de datos. Cuando nuestro GPS encuentre la ubicación aparecerá un pin rojo que indicará nuestra posición. En la parte de abajo en el centro disponemos de un botón llamado “AR Mode” que nos llevará directo a la ventana de *Navegación en modo RA*.



Figura 4.3: Mapa ULL.

En la ventana de *Navegación en modo RA* se nos mostrará la cámara XXX

A través del menú de la aplicación podemos acceder a la ventana de *Todos los centros ULL* (vease Figura 4.4a). Aquí se nos mostrarán todos los centros de la ULL que se encuentran en la base de datos. Se podrá hacer un búsqueda de cualquier centro en la barra superior de la aplicación. Si pulsamos cualquiera de estos centros se nos desplegará una ventana con la información detallada del centro.

En esta ventana (vease Figura 4.4b) se mostrará la información referente al centro. Aquí se nos mostrará una imagen del mismo, nombre y descripción del centro y una lista de enlaces con los servicios, secretarías, grados y departamentos que podemos encontrar. Disponemos de un botón en la parte inferior de la imagen del centro que nos abrirá la ruta a la ubicación del centro en Google Maps para poder llegar a él.

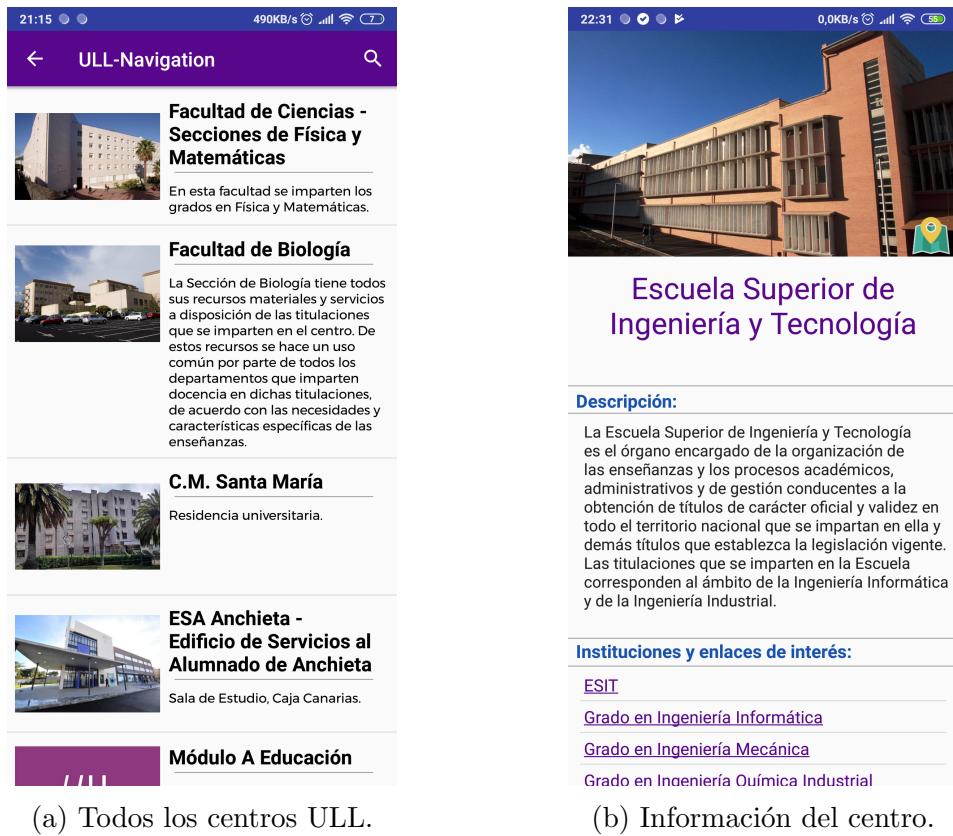


Figura 4.4: Ventanas de *Todo los centros ULL* e *Información del centro* de *ULL-Navigation*.

Por último desde el menú podemos acceder a las ventanas de *Configuración* y *Información*.

En la ventana de *Configuración* (vease Figura 4.5a) van los ajustes de la aplicación. En la ventana tenemos la opción para poder configurar si queremos encontrar los centros que se encuentran en el área entre dos circunferencias.

La ventana *Información* (vease Figura 4.5b) nos muestra información básica de la aplicación como el nombre, versión, correo de contacto, autor y objetivo e información de la app *ULL-Navigation*.

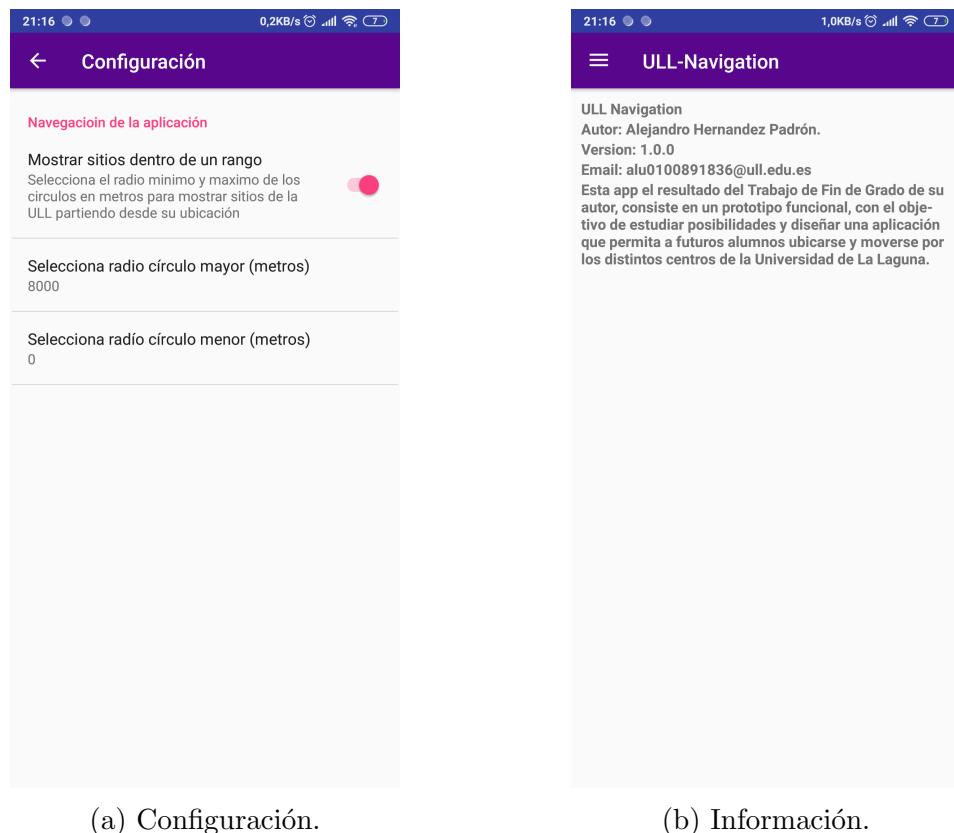


Figura 4.5: Ventanas de *Configuración* e *Información* de *ULL-Navigation*.

Capítulo 5

Back-end

Si recordamos en la capítulo 2 se habló sobre las tecnologías de Node.js y MongoDB, como servidor y base de datos respectivamente. Para poder implementar estas tecnologías se decidió por utilizar servicios en la nube para gestionarlos, facilitar su despliegue y con motivos de profundizar y adquirir conocimiento sobre este tipo de plataformas. Estos servicios son Heroku como servidor Node.js de nuestra aplicación y mLab como base de datos NoSQL que utiliza la tecnología de MongoDB. Ambos ofrecen una forma sencilla, accesible y escalable, para poder ejecutar nuestra aplicación y estudiar las posibilidades y beneficios que ofrecen este tipo de servicios en la nube en la actualidad.

5.1. Base de datos

Todo los datos necesarios para el funcionamiento de la aplicación estarán alojados en una base de datos en la nube. mLab será el proveedor de servicio escogido para alojar nuestra base de datos. Este proveedor nos ofrece bases de datos MongoDB.

Crear nuestra base de datos es muy sencillo con mLab. Simplemente nos registramos y creamos nuestra base de datos. Lo primero que necesitamos es elegir un plan de datos. Como nuestra base de datos no es necesario mucho espacio de almacenamiento se optó por el plan gratuito que dispone de 500 megas. Posteriormente fijamos el nombre de nuestra base de datos que será “bd-ull-navigation”.

5.2. Configuración del servidor

Necesitamos configurar nuestro servidor de Node.js para poder desplegarlo en Heroku y que funcione correctamente. Como requisito previo se instalara

GitHub en nuestra maquina para poder trabajar con el repositorio de Heroku que contendrá nuestro servidor.

5.2.1. Instalación

Para toda la instalación se ha utilizado Linux como sistema operativo.

GitHub

Será necesario disponer de una cuenta en GitHub primero para poder trabajar con el gestor de versiones. Para instalar GitHub en Linux utilizaremos el siguiente comando:

```
1 $ sudo apt install git-all
```

Instalación de Node.js

Para instalar Node.js en nuestra máquina con linux deberemos ejecutar estos comando:

```
1 $ curl -sL https://deb.nodesource.com/setup_11.x | sudo -E bash -
2 $ sudo apt install -y nodejs
```

Instalación de Heroku

Necesitamos tener una cuenta en Heroku para poder tener acceso y gestionar nuestro servidor en la nube.

Una vez hayamos entrado en nuestra cuenta. Creamos el repositorio en Heroku con el nombre de "server-ull-navigation". Este es el repositorio que Heroku desplegará con nuestro de servidor de Node.js en la nube.

A continuación instalaremos Heroku. Para ello necesitaremos instalar snap:

```
1 $ sudo apt install snapd
```

Una vez instalado snap podremos instalar Heroku:

```
1 $ sudo snap install heroku --classic
```

Después ejecutamos el comando el cual nos redireccionará a la página de Heroku en el navegador para que nos logueemos con nuestra cuenta y ya estará todo listo para empezar a configurar nuestra aplicación.

```
1 $ heroku login
```

5.2.2. Implementación

Una vez tengamos todo instalado podremos empezar a implementar nuestro servidor de Node.js.

Con el siguiente comando, ya tendremos preparado nuestro repositorio para empezar a trabajar:

```
1 $ git clone https://github.com/heroku/server-ull-navigation.git
```

Una vez dentro del directorio de nuestro repositorio necesitaremos ejecutar el comando:

```
1 $ npm init
```

Este comando resulta en la creación de un archivo llamado *package.json*. Este fichero se utiliza para administrar los paquetes disponibles en el Node Package Manager (npm) que se instalan localmente.

Ahora instalaremos los paquetes necesarios para el servidor:

- **ExpressJS:** Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles.
- **mongoose:** Mongoose es un Object Data Modeling (ODM) librería para trabajar MongoDB y Node.js.
- **body-parser:** Lo necesitamos para manejar las peticiones de JSON.
- **node-restful:** Maneja las peticiones recibidas del servidor.

Con un comando instalamos todos los paquetes y guardamos las dependencias utilizadas en *package.json*:

```
1 $ npm install --save express body-parser mongoose node-restful
```

El resultado del *package.json* sería el siguiente:

```
1 {
2   "private": true,
3   "name": "server-ull-navigation",
4   "version": "0.0.1",
5   "dependencies": {
6     "body-parser": "^1.18.3",
7     "express": "^4.16.3",
8     "mongoose": "^5.4.16",
9     "node-restful": "^0.2.6"
10   },
11   "engines": {
12     "node": ">=0.10.25"
13   }
14 }
```

Listado 5.1: Contenido del fichero *package.json*.

Inicializamos el servidor

Una vez ya instalados todos los paquetes creamos el fichero *server.js*. En este sera el que inicie el servidor y tenga las variables que configuran el mismo.

```

1 //Declaramos los paquetes que necesitamos
2 var express = require('express');
3 var app = express(); //Iniciamos nuestra app
4 //Ponemos en modo escucha la app
5 app.listen(3000, function(){
6     console.log("Express server listening on port 3000"
7 });

```

Listado 5.2: Configuración inicial del servidor.

Para comprobar que todo funciona utilizaremos el siguiente comando, que correrá nuestro servidor en local en el puerto 3000.

```
1 $ node app.js
```

Más adelante terminaremos de configurar este fichero. A continuación pasamos a explicar la estructura y organización de nuestro servidor.

Estructura de la aplicación

Dentro de las carpetas principales del repositorio creamos dos subdirectorios:

- **models**
- **routes**

A continuación crearemos el fichero *ullSites.js* dentro de la carpeta de **models**. Este fichero contiene el modelo que conecta con la colección de nuestra base de datos y maneja las respuestas como veremos más adelante.

```

1 //Declaramos la variable restful para manejar las peticiones
2 var restful = require('node-restful');
3 var mongoose = restful.mongoose; //Utilizamos mongoose para conectarnos a la BD
4
5 //Estructura de los sitios de la ull contenidos en la base de datos
6 var ullSitesSchema = new mongoose.Schema({
7     id: String,
8     name: String,
9     position: {
10         lat: String,
11         long: String
12     },
13     desc: String,
14     imageLink: String,
15     canFind: [
16         {
17             id: String,
18             link: String
19         }
20     ]
21 })

```

```

21 //Devolvemos el modelo para poder utilizarlo en otros ficheros
22 //Este modelo se conectara con colección de la base de datos "ull_sites"
23 module.exports = restful.model('ull_sites', ullSitesSchema);

```

Listado 5.3: Fichero *ullSites.js*.

Necesitaremos una ruta por la cual nuestro servidor responderá con la información que se solicite de la base de datos. Para ello en la carpeta **routes** crearemos el fichero *api.js*. Este fichero manejará las peticiones que lleguen a la ruta */api*.

```

1 var express = require('express');
2 var router = new express.Router();
3
4 //modelo que maneja la petición a la base de datos
5 var ullSites = require('../models/ullSites');
6
7 //Seleccionamos los métodos que puede responder
8 //Al ser una aplicación sencilla solo necesitamos manejar peticiones get
9 ullSites.methods(['get']);
10 //Indicamos al router la url que gestionara las peticiones
11 ullSites.register(router, '/ull-sites');
12
13
14 module.exports = router;

```

Listado 5.4: Fichero *api.js*.

En caso de acceder a la ruta “*/api/ull-sites*” de nuestra aplicación con una petición get, se enviará una respuesta al cliente con toda la información correspondiente de la base de datos.

Conexión con mLab

Con nuestra cuenta de mLab y base datos ya creada en mLab solo vamos a necesitar una url para poder acceder a ella y poder consultar, añadir y borrar datos. Esta url está disponible en la página principal de nuestra base de datos en mLab y tiene el siguiente formato:

```
1 mongodb://<dbuser>:<dbpassword>@ds235181.mlab.com:35181/ull-navigation
```

Donde “dbuser” es el usuario que usamos para crear la base de datos y “dbpassword” la contraseña. Para evitar que el usuario y contraseña queden expuestos públicamente en el repositorio vamos a utilizar variable de entorno. En nuestro caso vamos a configurar la variable de entorno para que funcione en Heroku. Para hacerlo necesitamos utilizar este comando en la terminal:

```
1 $heroku config:set PROD_MONGODB=mongodb://username:password@ds235181.mlab.com:35181/ull-navigation
```

Posteriormente la variable PROD_MONGODB se utilizará para conectarse a la base de datos una vez este desplegado en Heroku.

Solo vamos a necesitar crear una colección para el proyecto. Para ello accedemos a nuestra base de datos en el navegador y pinchamos en el botón de “Add collection” y la nombraremos “**ull_sites**”. Aquí tendremos la información de nuestros centros en instituciones de la ULL.

Cada centro de la ULL tendrá el siguiente formato e información:

```

1  {
2    "_id": {
3      "$oid": "5b5a004efb6fc07c4c24a5aa"
4    },
5    "id": "esit",
6    "name": "Escuela Superior de Ingeniería y Tecnología",
7    "position": {
8      "lat": "28.482965",
9      "long": "-16.322003"
10   },
11   "desc": "La Escuela Superior de Ingeniería y Tecnología es el órgano encargado de la organización de las enseñanzas y los procesos académicos, administrativos y de gestión conducentes a la obtención de títulos de carácter oficial y validez en todo el territorio nacional que se imparten en ella y demás títulos que establezca la legislación vigente.",
12   "imageLink": "https://www.ull.es/donde/assets/img/facultades/esit.jpg",
13   "canFind": [
14     {
15       "id": "ESIT",
16       "link": "https://www.ull.es/centros/escuela-superior-de-ingeneria-y-tecnologia/"
17     },
18     {
19       "id": "Grado en Ingeniería Informática",
20       "link": "https://www.ull.es/estudios-docencia/grados/ingenieria-informatica"
21     }
22   ]
23 }
```

Figura 5.1: ull-site. Ejemplo de un centro de la ULL en la base de datos.

Configuración final del servidor

Por último tenemos que acabar de configurar el fichero *server.js* que creamos anteriormente. Tenemos que preparar el servidor con la url de la base de datos de mLab y de disponer que las rutas de nuestra aplicación esté bien configurada para responder a las solicitudes.

```

1 var express = require('express');
2 var mongoose = require('mongoose');
3 var bodyParser = require('body-parser');
4
5 var app = express();
6 app.use(bodyParser.urlencoded({extended: false}));
7 app.use(bodyParser.json());
8
9 api = require('./routes/api');
10
11 //Realizamos la conexión con la base de datos con la url correctamente
12 //guardada en PROD_MONGODB
13 mongoose.connect(process.env.PROD_MONGODB, function (error) {
14   if (error) console.error(error);
15   else console.log('mongo connected');
16 });
```

```
17 app.get('/', function (req, res) { //Ruta inicial del servidor
18   res.send('ULL-Navigation server');//Respuesta por defecto
19 })
20 }
21
22 //Dejamos que el archivo /routes/api.js se encargue de las
23 //solicitudes que recibimos de /api
24 app.use('/api', api);
25
26 //Ponemos el servidor a escuchar
27 app.listen(process.env.PORT || 3000, function{
28   console.log("Express server listening on port %d in %s mode", this.address().port, app.settings
29   .env);
});
```

Listado 5.5: Configuración final del fichero *server.js*.

Despliegue en Heroku

Con nuestro servidor ya configurado su propósito ya podemos desplegarla en Heroku. Para ello necesitaremos primero crear un fichero *Procfile* en la raíz del repositorio que le dirá a Heroku cual es el fichero que inicia el servidor.

```
1 web: node server.js
```

Este es el fichero que Heroku ejecutará para iniciar la aplicación.

Para desplegar en Heroku necesitamos tres comandos:

```
1 $ git add .
2 $ git commit -m "Servidor final"
3 $ git push heroku master
```

Con estos pasos completados ya tendremos nuestro servidor desplegado y funcionando en la url: <https://server-ull-navigation.herokuapp.com>

Bibliografía

- [1] Android: **Android**. [<https://www.android.com/>] 2008, [[Disponible electrónicamente. Último acceso, julio de 2018]]. 2
- [2] Github: **Repositorio Github** 2008, [<https://github.com/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 2, 5
- [3] LaTeX3 Project: **LaTeX** 1985, [<https://www.latex-project.org/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 2
- [4] IntelliJ: **Android Studio** 2014, [<https://developer.android.com/studio/index.html>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 3
- [5] **IDE**. [https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 3
- [6] IntelliJ IDEA: **IntelliJ IDEA** 2001, [<https://www.jetbrains.com/idea/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 3
- [7] **Gradle** 2007, [<http://gradle.org/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 3
- [8] Google, Inc: **Eclipse con ADT plugin** 2012, [<https://marketplace.eclipse.org/content/android-development-tools-eclipse>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 4
- [9] **Repositorio de la aplicación ULL-NAVIGATION** [<https://github.com/alehdezp/TFG-ULL-NAVIGATION>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 5
- [10] **Plataforma del Programa de Apoyo a Trabajos Finales Libres** [<http://osl.ull.es/programa-apoyo-trabajos-finales-libres/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 5

- [11] **Google Play** [<https://play.google.com/store>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 6
- [12] **Samsung Apps** [<http://www.samsung.com/es/apps/mobile/galaxyapps/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 6
- [13] **Registro de Imagenes** [https://es.wikipedia.org/wiki/Registro_de_la_imagen]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 7
- [14] **CodigoQR.** [https://es.wikipedia.org/wiki/Codigo_QR]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 7
- [15] **Realidad Virtual.** [https://es.wikipedia.org/wiki/Realidad_virtual]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 9
- [16] **Relidad Mixta.** [https://es.wikipedia.org/wiki/Realidad_mixta]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 9
- [17] **Pokemón Go!** [<https://pokemongolive.com/es/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 10
- [18] **Pokemón.** [<https://es.wikipedia.org/wiki/Pokemon>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 10
- [19] **Node.js.** [<https://nodejs.org/en/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 12
- [20] **Chrome V8.** [https://es.wikipedia.org/wiki/Chrome_V8]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 12
- [21] **Event Loop.** [<https://nodejs.org/de/docs/guides/event-loop-timers-and-nexttick/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 12
- [22] **MongoDB.** [<https://www.mongodb.com/what-is-mongodb?lang=es-es>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 14
- [23] **Tecnología NoSQL** [<https://es.wikipedia.org/wiki/NoSQL>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 14
- [24] **Formato BSON.** [<https://es.wikipedia.org/wiki/BSON?lang=es-es>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 14
- [25] **PaaS.** [https://en.wikipedia.org/wiki/Platform_as_a_service]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 15

- [26] **Amazon Web Services.** [*https://aws.amazon.com/es/*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 16
- [27] **Azure.** [*https://azure.microsoft.com/es-es/*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 16
- [28] **Google Cloud.** [*https://cloud.google.com/?hl=es*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 16
- [29] **Google Maps API.** [*https://developers.google.com/maps/documentation/android-api*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 16
- [30] **Splash Screen.** [*https://en.wikipedia.org/wiki/Splash_screen*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 20
- [31] **Navigation Drawer.** [*https://material.io/design/components/navigation-drawer.html*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 20