



Trabajo de Fin de Grado

**ULL-AR. Tecnología de realidad
aumentada en entornos
universitarios**

Alejandro Hernández Padrón

La Laguna, 31 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-AR. Tecnología de realidad aumentada 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 31 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. compartiendo su conocimiento y exigiendo siempre lo mejor. A mis padres, hermano y amigos por apoyarme en el transcurso de mis estudios.

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 aumentada (RA) combinadas con técnicas de geolocalización.

Partimos de los conocimientos de programación en Java adquiridos en la asignatura: “Programación de Aplicaciones Interactivas” cursada en el itinerario de Ingeniería de Computación, también los conocimientos obtenidos en JavaScript, Node.js y MongoDB en la asignatura: “Prácticas Externas” realizada en la empresa “Itop Consulting”. Estas asignaturas, impartidas en el tercer y cuarto curso del Grado en Ingeniería Informática de “La Universidad de La Laguna”, han sido las que han sentado los fundamentos a partir de los cuales se ha desarrollado el trabajo.

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.

Palabras clave: Aplicaciones Android, Java, La Nube, dispositivos móviles, programación, Realidad Aumentada, Node.js, MongoDB, Google Maps.

Abstract

This document constitutes the research work of the student during the development process of an application for Android mobile devices through the use of augmented reality (AR) techniques combined with geolocation techniques.

Based on the knowledge of *Java* programming obtained in the subject: “*Programming of Interactive Applications*” studied in the itinerary of Computer Engineering, also the knowledge acquired in *JavaScript*, *Node.js* and *MongoDB* in the subject: “*External Practices*” carried out in the company “*Itop Consulting*”. These subjects, carried out in the third and fourth year of the degree in Computer Engineering of “*La Universidad de La Laguna*”, have been the ones who have laid the foundations from which the work has been developed.

Moreover, the student has learned independence in his work and gained vision and scheduling aptitudes, developing different labors of research, development and documentation that have come to give her a wide knowledge during the development of this project.

Keywords: *Application for Android, Java, Cloud Computing, mobile devices, programming, Augmented RealityNode.js, MongoDB, Google Maps.*

Índice general

1. Objetivos	1
2. Herramientas y Tecnologías	2
2.1. Herramientas de Desarrollo	2
2.1.1. Android Studio	2
2.1.2. LaTex	3
2.1.3. Github	3
2.2. Tecnologías utilizadas	4
2.2.1. El Sistema Operativo Android	4
2.2.2. Realidad Aumentada	4
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. RA en entornos universitarios	17
3.1. Posibles aplicaciones	18
3.1.1. Prácticas en laboratorios	18
3.1.2. Trabajos de campo	19
3.1.3. Eventos	19
3.1.4. Libros	19
3.1.5. Visitas	19
3.1.6. Aprendizajes experimentales	20
3.1.7. ejemplos	20
3.2. Aplicaciones móviles en entornos universitarios	21
3.2.1. Descarga automática de material	21
4. La aplicación ULL-AR	22
4.1. Especificación de requisitos	22
4.2. Especificación detallada de los requisitos	23

4.3. Ventanas de la aplicación	24
5. Back-end de ULL-AR	29
5.1. Base de datos	29
5.2. Configuración del servidor	29
5.2.1. Requisitos previos	30
5.2.2. Implementación	30
Bibliografía	35

Índice de figuras

2.1.	Android Studio, un IDE flexible e intuitivo.	3
2.2.	Ejemplo de uso de la RA en servicios de mantenimiento.	5
2.3.	Google Glass. Desmostración del uso de RA en deportes.	6
2.4.	Marker-bases AR. Demostración de su funcionamiento.	7
2.5.	RA Volcán. Ejemplo de uso de RA en Educación.	9
2.6.	Funcionamiento de Node.js.	13
2.7.	Ventajas de Node.js.	13
2.8.	Ventajas de MongoDB.	15
4.1.	Ventanas de iniciales de <i>ULL-AR</i>	24
4.2.	Ventana <i>Inicio</i> y el Menú de <i>ULL-AR</i>	25
4.3.	Mapa ULL.	26
4.4.	Ventanas de <i>Todas las instalaciones ULL</i> e <i>Información de la instalación de ULL-AR</i>	27
4.5.	Ventanas de <i>Configuración</i> e <i>Info</i> de <i>ULL-AR</i>	28
5.1.	Ejemplo de una instalación de la ULL en la base de datos.	33

Capítulo 1

Objetivos

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).

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.
- 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].
- Otro objetivo presente en este TFG es que el alumno investigue y profundice en las técnicas y tecnologías de realidad aumentada presentes en la actualidad.
- A su vez, se busca que el alumno adquiera conocimientos sobre las capacidades y ventajas del uso de servicios de computación en la nube.
- 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-AR.

2.1. Herramientas de Desarrollo

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 el desarrollador 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, el cual ha sido utilidad en las primeras ventanas de la aplicación.
- La facilidad y velocidad para aplicar cambios a aplicaciones ya funcionando.
- 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.

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 usar esta herramienta debido a la calidad profesional de los documentos que genera. Otra de las ventajas de esta herramienta es que te permite separar el formato y el contenido del documento, esto permite concentrarte más en la creación del contenido sin tener que distraerte con su diseño.

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.

Mediante el uso de este repositorio, el alumno ha conseguido ampliar sus conocimientos en Git y familiarizarse con la interfaz de GitHub.

Para instalar GitHub en Linux utilizaremos el siguiente comando:

¹ `$ sudo apt install git-all`

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: dispositivos móviles, 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.

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.

La RA está 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. Empresas de numerosos sectores ya han estado invirtiendo en su desarrollo debido a que los beneficios que traerá esta tecnología y las posibles aplicaciones por descubrir son prometedoras.

En un futuro la RA estará integrada en nuestro día a día y formará parte de la vida cotidiana. 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 cómo queda un mueble en tu salón o sus aplicaciones para realizar videojuegos, como podemos comprobar con el reciente éxito de Pokémon Go!.



Figura 2.2: Ejemplo de uso de la RA en servicios de mantenimiento.

A continuación, se explicará en detalle:

- ¿Cómo 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.

¿Cómo funciona esta tecnología?

La RA 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, dispositivos móviles, 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.

Todo esto necesita una parte software. El software en una primera parte deberá de reconocer el terreno, ubicaciones, objetos e imágenes, mediante los datos de los sensores y la cámara. Este proceso de transformación de diferentes conjuntos de datos a un sistema de coordenadas se llama registro de la imagen [8].

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 2.3: Google Glass. Desmostración del uso de RA 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, debido a que contienen puntos visuales únicos. Un buen ejemplo de este tipo de imágenes son los conocidos códigos QR [9]. Una vez se ha reconocido un marcador, se puede agregar información virtual, esta información puede ser la incorporación de animaciones o videos en una página de un libro de texto, simulaciones de objetos 3D o arquitecturas sin llegar a construirlas de forma física (véase Figura 2.4).



Figura 2.4: Marker-bases AR. Demostración de su funcionamiento.

- Markerless AR. Corresponde a la RA que recoge los datos de su posición y orientación para mostrar la información correspondiente a esa área. Estos sistemas utilizan los datos obtenidos de la cámara, GPS, brújula, giroscopio y acelerómetro para establecer la ubicación del usuario en el entorno. Utiliza técnicas para reconocer terreno o ambientes para calcular la posición y orientación de la cámara. Con este tipo de tecnología podemos probar un mueble en nuestro salón antes de llegar a comprarlo.
- Projection-based AR. Este 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.

A parte de los cuatro tipos de RA, existen subtipos dentro de cada uno de ellos.

“Location-based AR” es un tipo de “Markerless AR” que se centra más en los cálculos de la orientación y geolocalización del dispositivo. Esta técnica 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. Esta será la técnica de RA implementada en la aplicación a desarrollar en este TFG.

Diferencias entre Realidad Aumentada y Realidad Virtual

En los últimos años la realidad virtual (RV) o virtual reality (VR [10]) en inglés y la realidad aumentada han empezado a recibir mucha más atención. Llevando a desarrolladores llevar acabo integraciones de ambas tecnologías en numerosas industrias.

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

La 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 de mundo exterior, sino que lleva al mundo real objetos virtuales 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 [11]). Es un tipo de realidad similar a la realidad aumentada, pero con una idea más ambiciosa de mezclar lo real con lo virtual.

En la realidad mixta se trata de llevar el mundo real al mundo virtual. La idea es generar un modelo 3D de la realidad y sobre él superponer información virtual. De esta forma, se podrán combinar ambas realidades para agregar contenido adicional de valor para el usuario de RM. La RM es mucho más inmersiva que la realidad aumentada y requiere de mucho más poder de procesamiento.

Futuro y usos de la Realidad Aumentada

Actualmente la realidad aumentada la tenemos más mano que en años anteriores. Dispositivos como los dispositivos móviles 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 todos 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 dispositivo móvil el cual 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 más 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 (véase Figura 2.5) o del latido de un corazón.

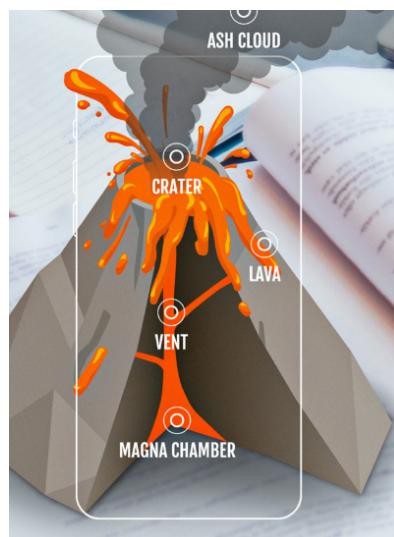


Figura 2.5: RA Volcán. Ejemplo de uso de RA en Educación.

- 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 dentro de cada juego. En un futuro esta tecnología revolucionará todo el sector de los videojuegos y cambiará la manera en la que interactuamos con ellos.

Un claro ejemplo del potencial de la RA en este sector, está en el éxito de Pokemón Go! [12]. Una aplicación para dispositivos móviles 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 [13], los cuales se encuentran escondidos en ubicaciones del mundo real.

- Realidad Aumentada en la Industria. El desarrollo de aplicaciones 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 aplicación puede mostrarnos toda clase de avisos sobre las piezas deterioradas. Incluso nos puede llegar a mostrar contenido visual en 3D sobre cómo 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.

Un posible uso en este sector es 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 a los visitantes en tiempo real todos los textos, señales y anuncios, a través de sus dispositivos móviles.

- Realidad Aumentada en Medicina. En cuanto a la Medicina, es interesante ver cómo avanza la tecnología en este campo que sin duda 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 más 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 estos sean marcados con un distinto color, facilitando mucho las labores de los médicos.

Android Studio y la RA

Para la integración de realidad aumentada en Android, se han probado y estudiado los kits de desarrollo de software (SDK [14]) de realidad aumentada disponibles para Android Studio. Estos son:

- **Vuforia** es un SDK de realidad aumentada que permite a los desarrolladores de aplicaciones crear rápidamente experiencias de RA inmersivas, de alta fidelidad y centradas en el móvil. El SDK de Vuforia aprovecha la tecnología de visión por computadora para identificar y rastrear imágenes y objetos 3D en tiempo real. Esta funcionalidad permite orientar y colocar objetos virtuales, incluidos modelos 3D y otros contenidos, en relación con el entorno del mundo real. Los modelos 3D y la información digital se pueden superponer sobre la escena del mundo real y verlos en relación con el entorno a través de un dispositivo móvil.
- **Kudan AR SDK** es una plataforma diseñada para desarrolladores de RA como una plataforma preparada para admitir tanto en el reconocimiento basado en marcadores como sin marcadores y el seguimiento de ellos. El motor Kudan SDK principal, se desarrolla completamente en C++ y posee optimizaciones específicas de la arquitectura desarrolladas para proporcionar un rendimiento más rápido y sólido sin afectar negativamente el espacio de memoria.
- **MaxST SDK** de realidad aumentada proporciona un motor RA integral multiplataforma equipado con todas las características requeridas por los desarrolladores para crear experiencias y aplicaciones de RA. El MaxST AR SDK proporciona las siguientes funcionalidades: seguimiento instantáneo, SLAM [15] (utiliza la cámara del teléfono inteligente para crear un "mapa virtual" del área circundante), rastreo de objetos, reconocimiento de imágenes y de marcadores.

Se ha optado por utilizar "Kudan AR SDK" para el desarrollo de la aplicación

de este TFG debido a problemas de compilación y flexibilidad en las opciones gratuitas de los otros dos SDK en Android Studio.

2.2.3. Node.js

Node.js [16] 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 [17]. Node.js es un entorno de JavaScript del lado del servidor, basado en eventos. Utilizar el motor de Chrome V8 permite a Node.js un entorno de ejecución que compila y ejecuta JavaScript a velocidades increíbles.

¿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 simultánea 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 las solicitudes de forma unificada en un único hilo complementado con un bucle de eventos (Event Loop [18]) de tipo asíncrono. De este modo cada petición que se recibe se trata como un evento y pertenecen a este único bucle (véase Figura 2.6). Este nuevo replanteamiento proporciona un lenguaje con una capacidad para gestionar una gran cantidad de solicitudes y conexiones con la máxima eficiencia.

Node.js utiliza un E/S de tipo asíncrono. En los modelos de tipo asíncronos, las tareas que efectúa el servidor se realizan de manera simultánea y repartidas entre los hilos del procesador. Este procedimiento evita que se produzcan bloqueos y proporciona una mayor potencia y velocidad de procesamiento.

Otro de sus puntos fuertes es su gestor de paquetes Node Package Manager (NPM). Gracias a este gestor se pueden instalar paquetes, módulos y agregar dependencias de manera simple.

¿Porqué 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.

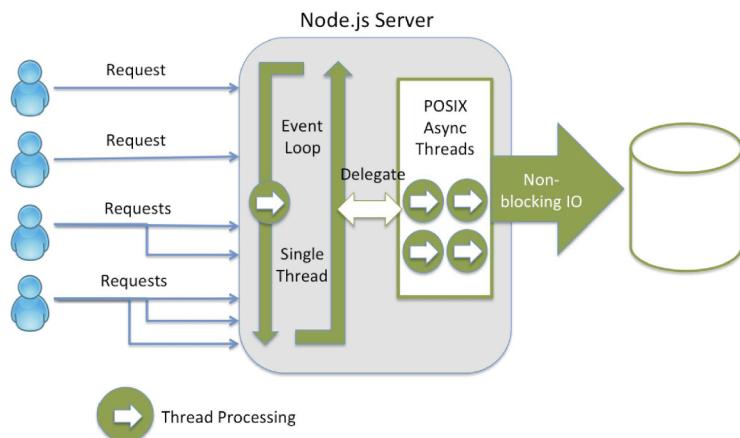


Figura 2.6: Funcionamiento de Node.js.

- 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.
- Un proceso de desarrollo y programación de aplicaciones rápido y liviano.

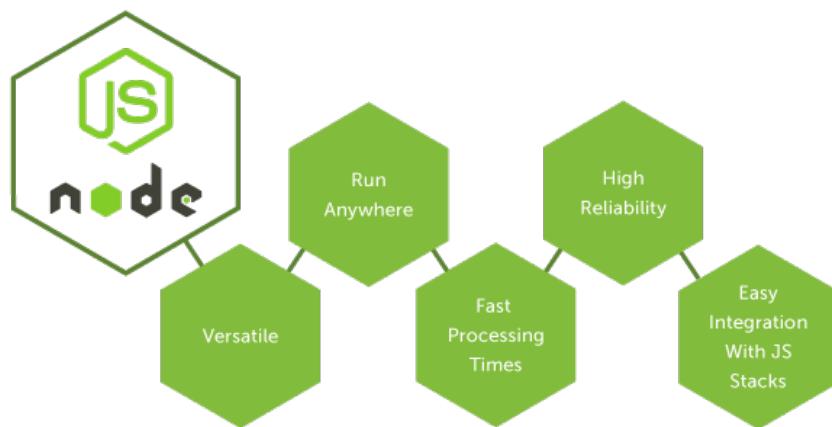


Figura 2.7: Ventajas de Node.js.

Para instalar Node.js en nuestra máquina con Linux deberemos ejecutar estos comandos:

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

2.2.4. MongoDB

Descripción

MongoDB [19] es un sistema de base de datos multiplataforma NoSQL [20] orientado a documentos. Esta base de datos 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. Este esquema difiere de las tablas de bases de datos relacionales.

¿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 [21] también llamada JSON [22] 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 en MongoDB contienen un conjunto de documentos y funciones equivalentes a las de las bases de datos relacionales.

Ventajas de MongoDB

- 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 los indexamos. En MongoDB este indexado provee de rápidos tiempos de respuesta a las peticiones, esto le permite ser hasta 100 veces más rápido 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.

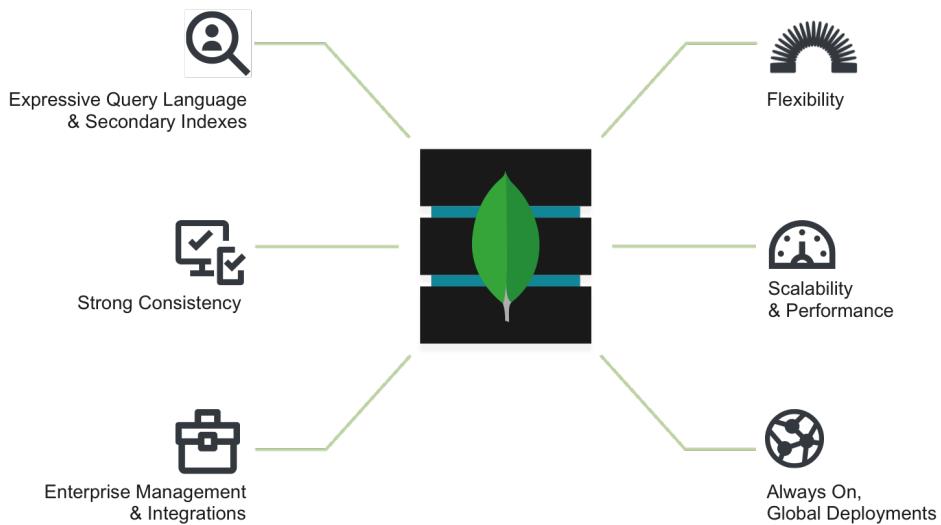


Figura 2.8: Ventajas de MongoDB.

2.2.5. Heroku

Heroku es una plataforma como servicio de computación (PaaS [23]) en la nube que soporta distintos lenguajes de programación. Heroku 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 ordenador 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 más espacio o capacidad de procesamiento, o se pueden añadir más dynos si tu aplicación lo necesita. Heroku a cada mes realiza un cobro en función de los dynos que tengas contratados.

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.

Para instalar Heroku en nuestro dispositivo solo tenemos que seguir los pasos de la web [24].

2.2.6. mLab

mLab es un servicio de base de datos en la nube totalmente gestionado que ofrece aprovisionamiento y escalados 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 [25], Azure [26] y Google Cloud [27].

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 [28] 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 (fusión 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

RA en entornos universitarios

En este capítulo trataremos los usos y ventajas de la integración de la realidad aumentada en entornos universitarios.

La realidad aumentada se presenta en el ámbito educativo como una tecnología capaz de aportar transformaciones significativas en la forma en que los estudiantes perciben y acceden a la realidad física, proporcionando así experiencias de aprendizaje más ricas e inmersivas.

Con la continua implantación de nuevas tecnologías en las aulas, junto al incremento de dispositivos móviles en la población, sitúa a la RA en una posición destacada en la actualidad. En la actualidad en educación, la realidad aumentada rara vez se usa, pero cada vez más docentes, investigadores y desarrolladores están comenzando a moverse hacia nuevos métodos de enseñanza más interactivos.

La tecnología de la RA permite cambiar la forma de entender los contenidos de aprendizaje, puesto que aporta nuevas formas de interacción con el mundo real a través de capas digitales de información que amplían, completan y transforman en cierto modo la información inicial.

Las aplicaciones que tiene la RA, en lo referente a la creación de materiales didácticos y actividades de aprendizaje son múltiples, directas y fáciles de imaginar en prácticamente todas las disciplinas, sobre todo, las relacionadas con las ciencias aplicadas (ingeniería, química y física, biología), pero también en el campo del diseño industrial, la cirugía, la arqueología, etc.

Los beneficios potenciales de la RA aplicados a la educación incluyen:

- Aumentar o enriquecer la información de la realidad para hacerla más comprensible al estudiante.
- Permite múltiples formas de visualización de conceptos teóricos difíciles.
- El uso de una interfaz tangible para la manipulación de objetos, que permite observar un objeto desde diferentes puntos de vista, seleccionando,

el estudiante, el momento y posición de observación.

- Potenciar el aprendizaje ubicuo [29].
- Crear escenarios “artificiales” seguros para los estudiantes como pueden ser laboratorios o simuladores.
- Enriquecer los materiales impresos para los estudiantes con información adicional en diferentes soportes.
- Facilita la colaboración efectiva y discusión entre los estudiantes.

Hay que tener en cuenta una serie de requisitos a la hora de diseñar un sistema educativo de RA. Estos requisitos son:

- Ser sencillo y robusto.
- Permitir que el educador ingrese información de manera simple y efectiva.
- Proporcionar al alumno información clara y concisa.
- Permitir una fácil interacción entre estudiantes y educadores.
- Realizar procedimientos complejos transparentes para los alumnos.
- Ser rentable y fácilmente extensible.

3.1. Posibles aplicaciones

3.1.1. Prácticas en laboratorios

Los laboratorios, poseen instrumental de aprendizaje que engloba más información de la que por su apariencia aporta, lo que hace que sea un escenario ideal para el uso de tecnología como la realidad aumentada. A todos aquellos elementos que podemos encontrar en un laboratorio se les puede asociar video tutoriales con información de uso, información en texto, archivos audibles, etc., que pueden ser accesibles de forma sencilla desde un dispositivo móvil. Permite la realización de prácticas en las que el profesor añada la información a los elementos del laboratorio y los alumnos sean los que consulten esa información, pueden ser los propios alumnos los que integren la información en el laboratorio, pueden crear varios puestos con información a modo de instrucciones de una práctica, etc.

3.1.2. Trabajos de campo

Al igual que en el caso de los laboratorios cualquier experiencia o práctica que hagamos es susceptible del uso de la realidad aumentada. Se podrá asociar información a un entorno objeto de estudio tanto por parte del alumnado como el profesorado para su trabajo de forma experimental de una forma muy sencilla. De esta manera, objeto de conocimiento y conocimiento se dan en el mismo tiempo y lugar. Un par de ejemplos pueden ser la realización de rutas por ciudad visitando lugares emblemáticos y descubriendo la información asociada a esos sitios, estatuas, edificios, monumentos, etc., o por zonas rurales, de montaña en las que podríamos identificar especies, accidentes geográficos, etc.

3.1.3. Eventos

En este tipo de ejemplo de uso cabrían las exposiciones, seminarios, jornadas, encuentros, etc. A través de la documentación que se realiza para los asistentes, ponentes y a modo de publicidad se pueden incluir códigos QR en posters informativos, en folletos, catálogos o en las webs de los eventos. Si utilizamos una aplicación específica de igual manera puede incluirse información adicional. Es un recurso muy interesante ya que es un modo de incluir gran cantidad de información asociada al evento accesible con cualquier soporte móvil en cualquier sitio y lugar debido a la ubicuidad de estos dispositivos.

3.1.4. Libros

A los libros electrónicos o en formato papel se añade realidad aumentada utilizando como activador de la información los textos, ilustraciones, encabezados, pies de página, etc., y como información adicional en muchos casos se incluye la biografía del autor, los pies de página, vídeos que desarrollan la acción más ampliada, textos adicionales y audios. Se denominan libros aumentados. El libro enmarcado en el proyecto: “HUSO DIGITAL: LA CIUDAD UNIVERSITARIA EN REALIDAD AUMENTADA, “El libro aumentado de Eduardo Torroja” es un claro ejemplo.

3.1.5. Visitas

En muchos casos, a lo largo del curso académico se realizan salidas fuera del aula y se visitan lugares como complemento educativo a las clases regladas. Los museos, galerías, fábricas, empresas, incorporan la realidad aumentada en sus recorridos proporcionando una información completa y audiovisualmente muy atractiva a los visitantes. Los estudiantes además de aprender la materia objeto de la visita

desarrollan las destrezas que el manejo de esta tecnología les proporciona.

3.1.6. Aprendizajes experimentales

prácticamente todas las disciplinas tienen una parte experimental que pueden realizarse con realidad aumentada facilitando en gran medida el aprendizaje y el desarrollo de destrezas transversales. Ejemplos claros pueden ser en medicina, donde el uso de las google glass de forma experimental hace un par de años fue muy mediático, en arquitectura e ingenierías, la posibilidad de realizar y ver modelos en 3D de diferentes edificios y construcciones es muy útil en el aprendizaje del alumno. En química o física con aplicaciones como las que aparece en el bloque 3 dedicado a programas y aplicaciones, también en ramas como la Gabinete de Tele-Educación 24 Universidad Politécnica de Madrid Realidad Aumentada en Educación biología, arte, historia, diseño, idiomas, geografía, matemáticas, urbanismo, música, geometría, etc.

3.1.7. ejemplos

Desde la Escuela Técnica Superior de Ingenieros en Topografía, Geodesia y Cartografía de la Universidad Politécnica de Madrid y de una manera muy sencilla nos presentan una práctica extrapolable a cualquier disciplina interesada en la observación del relieve de los mapas. Consiste en un simulador con un cajón de arena moldeable y tras ser capturado por una cámara con un sensor infrarrojo calcula un modelo digital tridimensional del terreno, además se proyectan sobre la arena las curvas de nivel, coloreadas con tintas hipsométricas. Permite la simulación de agua y de volcanes a modo de realidad aumentada. Sobre la arena se verá la representación cartográfica.

<http://idav.ucdavis.edu/> okreylos/ResDev/SARndbox/

Varios científicos de la Universidad Carlos III de Madrid han diseñado unas gafas inteligentes que permiten conectar a profesores y alumnos en tiempo real en el aula. Telmo Zarraonandia, Ignacio Aedo, Paloma Díaz y Álvaro Montero a través del artículo, An augmented lecture feedback system to support learner and teacher communication explican su funcionamiento. Con tan solo ponerse las gafas en cuestión, el docente obtendrá información del alumno al mirar tras ellas. Notas y comentarios que lanzarán los alumnos al docente los podrá recibir con tan solo observar a su grupo de clase.

El feedback estará asegurado en estas aulas, de nuevo utilizando como recurso tecnológico la realidad aumentada.

Collaborative work of students within the Augmented Reality application Construct3D

Libros de texto aumentados

Modelos 3D utilizados en arquitectura para la visualización de dispositivos
Casos de usos.

3.2. Aplicaciones móviles en entornos universitarios

- XXX
- XXX

3.2.1. Descarga automática de material

Capítulo 4

La aplicación ULL-AR

En este capítulo explicaremos la aplicación ULL-AR utilizando la especificación de requisitos de esta 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 para 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, la cual les permita ubicarse, detectar y reconocer las instalaciones 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 IDE para su desarrollo.
- Se implementarán técnicas de realidad aumentada basadas en la geolocalización para mostrar al usuario la instalación de la ULL a la cual apunte con la cámara.
- Las instalaciones, junto a su información correspondiente, estarán ubicadas en un base datos en la nube. El servidor que se conecte con esta base de datos también deberá estar en la nube.

4.2. Especificación detallada de los requisitos

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 ha de ser alumno de la Universidad de La Laguna y poseer su respectiva cuenta de Google con el correo institucional de la ULL. Este correo ha de tener el siguiente 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 una ventana de *Inicio* en la que aparecerá un acceso directo a las ventanas de *Mapa ULL* y *Navegación en modo RA* que explicaremos más adelante. A su vez, 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 un 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:

- Inicio. Ventana inicial con la que se abre la aplicación.
- Mapa ULL. Esta ventana contendrá un mapa de la universidad con todas las instalaciones en la base de datos.
- Navegación en modo RA. En esta ventana, mediante el uso de la cámara, se identificarán las instalaciones universitarias a los que el usuario apunte y permitirá mostrar una pestaña con información detallada de las mismas.
- instalaciones de la ULL. Contiene todas la ubicaciones e instalaciones de la Universidad de La Laguna y permitirá la búsqueda de estas.
- 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 de su autor.

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

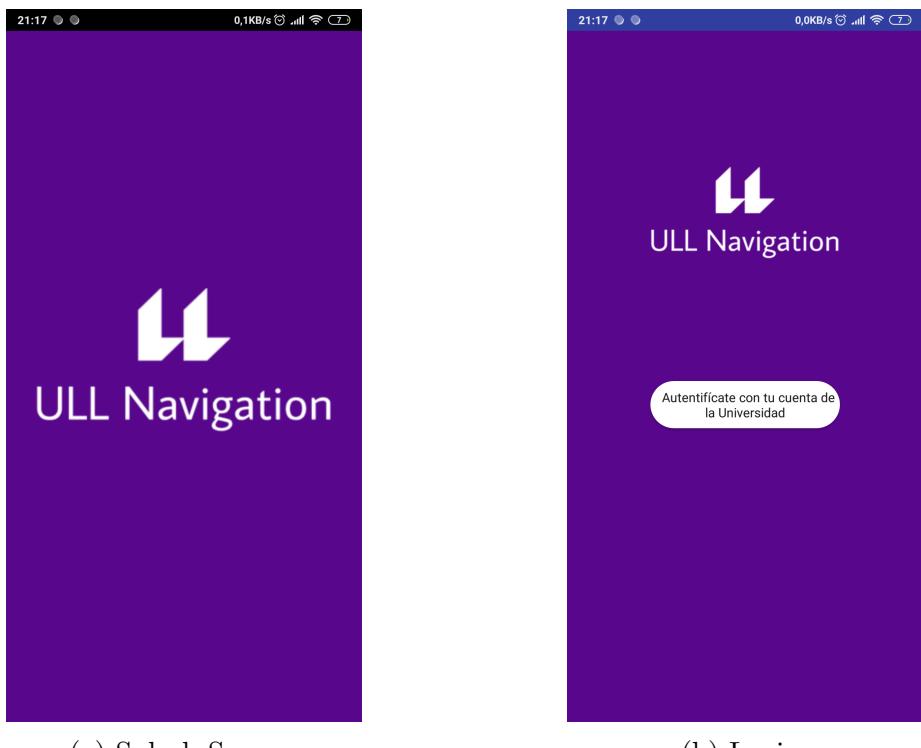
- **Id:** campo para identificar la instalación.
- **Nombre:** nombre oficial de la instalación.

- **Ubicación:** la ubicación exacta en la que se encuentra la instalación.
- **Descripción:** descripción de la instalación con el objetivo y actividades que se desarrollan en él.
- **Imagen:** imagen de la instalación.
- **Lista de enlaces de interés:** una lista con los enlaces a las instituciones, servicios, departamentos y grados que se imparten en esta instalación.

Esta información estará guardada en una base de datos en la nube. Para acceder a esta, 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.



(a) Splash-Screen.

(b) Login.

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

La primera ventana (véase Figura 4.1a) es la pantalla de inicio o Splash Screen que aparece cuando ejecutamos la aplicación. Tras unos segundos, se carga la

ventana de *Login* (véase Figura 4.1b). Para poder loguearnos deberemos tener una cuenta de Google de la ULL. Una vez que presionamos en el botón del medio, se nos abrirá una ventana de diálogo para que pongamos nuestra cuenta.

Cuando nos hayamos logueado con éxito, se nos abrirá la venta de *Inicio* (véase Figura 4.2a). En esta tendremos una lista de accesos directos a las funcionalidades principales de la aplicación, como son *Navegación en modo RA*, *Mapa ULL* y una serie de enlaces a sitios web relacionados con la ULL.

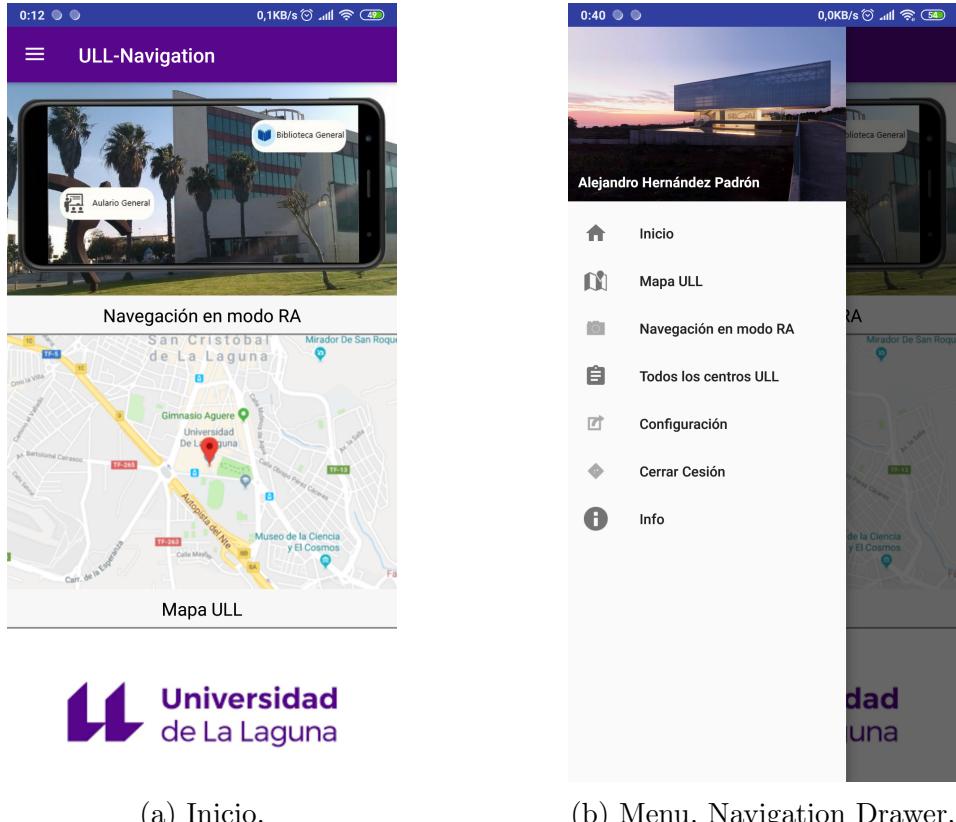


Figura 4.2: Ventana *Inicio* y el Menú de *ULL-AR*.

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 (véase Figura 4.2b).

Si nos movemos a la ventana de *Mapa ULL* (véase 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 el GPS del dispositivo 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.

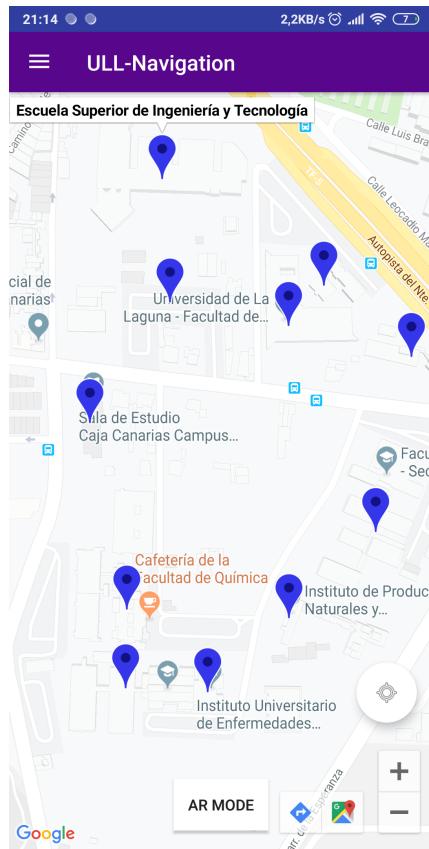


Figura 4.3: Mapa ULL.

En la ventana de *Navegación en modo RA* se nos mostrará la vista de la cámara de nuestro dispositivo

XXX

A través del menú de la aplicación podemos acceder a la ventana de *Todas las instalaciones ULL* (véase Figura 4.4a). Aquí se nos mostrarán todas las instalaciones de la ULL que se encuentran en la base de datos. Se podrá hacer una búsqueda de cualquier instalación en la barra superior de la aplicación. Si pulsamos cualquiera de estas instalaciones se nos desplegará una ventana con la información detallada de la instalación.

En la ventana *Información de la instalación* (véase Figura 4.4b) se dispondrá la información de esta. Aquí se nos mostrará una imagen de este, nombre y descripción de la instalación 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 de la instalación que nos abrirá la ruta a su ubicación

en la aplicación de Google Maps para poder llegar a ella.

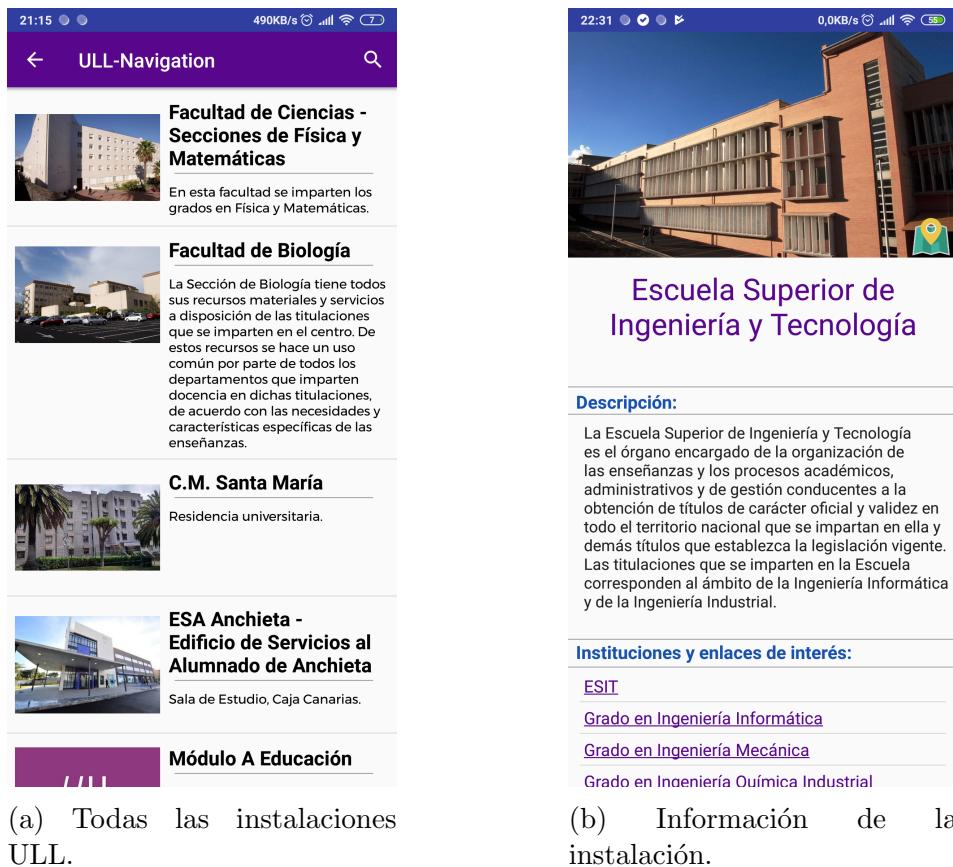


Figura 4.4: Ventanas de *Todas las instalaciones ULL* e *Información de la instalación* de *ULL-AR*.

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

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

La ventana *Info* (véase Figura 4.5b) nos muestra información básica de la aplicación como el nombre, versión, correo de contacto, autor y objetivo e información del desarrollo de la aplicación *ULL-AR*.

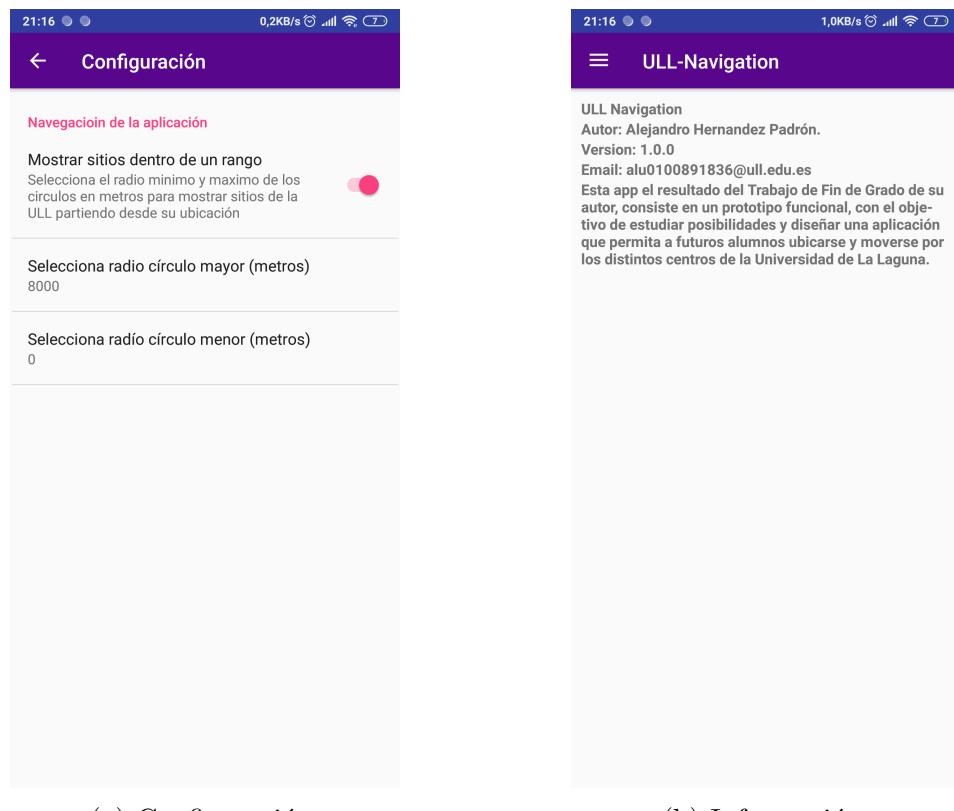


Figura 4.5: Ventanas de *Configuración* e *Info* de *ULL-AR*.

Capítulo 5

Back-end de ULL-AR

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 de nuestra aplicación y mLab como base de datos. 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

Todos 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 NoSQL que utilizan la tecnología 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 en 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-AR”.

5.2. Configuración del servidor

Necesitamos configurar nuestro servidor de Node.js para poder desplegarlo en Heroku y que funcione correctamente. A continuación se explicarán los pasos seguidos para la implementación nuestro servidor explicando su funcionamiento y su conexión con la base de datos.

5.2.1. Requisitos previos

Para toda la instalación e implementación del servidor Node.js se ha utilizado Linux como sistema operativo.

Previo a la implementación necesitaremos crearnos cuentas en las plataformas de Heroku y mLab. A su vez, debemos tener instalado GitHub, Node.js y Heroku. Para poder ejecutar los comandos que nos permitan creación y despliegue del servidor en la nube.

Una vez hayamos entrado en nuestra cuenta de Heroku. Creamos un repositorio con el nombre de "server-ull-AR" que será el servidor de nuestra aplicación en la nube.

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-AR.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 que se instalan localmente.

Ahora instalaremos los paquetes necesarios para el servidor, que son:

- **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 una librería para trabajar MongoDB y Node.js.
- **bodyparser:** Lo necesitamos para manejar las peticiones de JSON.
- **node-restful:** Sirve para manejar las peticiones recibidas del servidor y conectarse con MongoDB.

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-AR",
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 (véase Listado 5.2).

```

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 server.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 la carpeta principal 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 el variable restful para manejar las peticiones
2 var restful = require('node-restful');
3 //Utilizamos mongoose para conectarnos a la BD
4 var mongoose = restful.mongoose;
5
6 //Estructura de los sitios de la ull contenidos en la base de datos
7 var ullSitesSchema = new mongoose.Schema({
8     id: String,
9     name: String,
10    position: {
11        lat: String,
12        long: String
13    },
14    desc: String,
15    imageLink: String,
16    canFind: [
17        {
18            id: String,
19            link: String
20        }
21    }
22
23 //Devolvemos el modelo para poder utilizarlo en otros ficheros
24 //Este modelo se conectara con colección de la base de datos "ull_sites"
25 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* (véase Listado 5.4). Este fichero manejará las peticiones que lleguen al servidor a través de la ruta */api/* y se encargará principalmente de conectarse con la base de datos para responder a estas peticiones.

```

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 gestionará 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-AR
```

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-AR
```

Posteriormente la variable “PROD_MONGODB” se utilizará para conectarse al 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 las instalaciones de la ULL. Los cuales añadiremos manualmente desde la página de mLab.

```
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-ingineria-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: Ejemplo de una instalación 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. Necesitaremos indicar al servidor la url de la base de datos de mLab y de disponer que las rutas de nuestra aplicación estén bien configuradas para responder a las solicitudes.

```

1 var express = require('express'); //Servidor
2 var mongoose = require('mongoose');//Para conectar con BD
3 var bodyParser = require('body-parser');//Manejar peticiones JSON
4
5 var server = express(); //Instanciamos el servidor
6 server.use(bodyParser.urlencoded({extended: false}));
7 server.use(bodyParser.json());
8
9 //Utilizamos el fichero que se encuentra en ./routes/api.js
10 api = require('./routes/api');
11
12 //Realizamos la conexión con la base de datos con la url correctamente
13 //guardada en PROD_MONGODB
14 mongoose.connect(process.env.PROD_MONGODB, function (error) {
15   if (error) console.error(error);
16   else console.log('mongo connected');
17 });
18
19 server.get('/', function (req, res) { //Ruta raíz del servidor
20   res.send('ULL-AR server');//Respuesta por defecto
21 })
22
23 //Decimos al servidor que el archivo /routes/api.js se encargue de las
24 //solicitudes que recibimos de /api
25 server.use('/api', api);
26
27 //Ponemos el servidor a escuchar
28 server.listen(process.env.PORT || 3000, function(){
29   console.log("Express server listening on port %d in %s mode", this.address().port, server.
30   settings.env);
31 });

```

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

Despliegue en Heroku

Con nuestro servidor ya configurado, podemos realizar el despliegue en Heroku. Para ello necesitaremos primero crear un fichero *Procfile* en la raíz del repositorio, que le dirá a Heroku cuál es el fichero que inicia el servidor.

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

Para desplegarlo 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-AR.herokuapp.com>

Bibliografía

- [1] Android: **Android**. [<https://www.android.com/>] 2008, [[Disponible electrónicamente. Último acceso, julio de 2018]]. 1
- [2] Github: **Repositorio Github** 2008, [<https://github.com/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 1, 3
- [3] LaTeX3 Project: **LaTeX** 1985, [<https://www.latex-project.org/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 1
- [4] IntelliJ: **Android Studio** 2014, [<https://developer.android.com/studio/index.html>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 2
- [5] **IDE**. [https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 2
- [6] IntelliJ IDEA: **IntelliJ IDEA** 2001, [<https://www.jetbrains.com/idea/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 2
- [7] **Gradle** 2007, [<http://gradle.org/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 2
- [8] **Registro de Imagenes** [https://es.wikipedia.org/wiki/Registro_de_la_imagen]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 6
- [9] **CodigoQR**. [https://es.wikipedia.org/wiki/Codigo_QR]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 6
- [10] **Realidad Virtual**. [https://es.wikipedia.org/wiki/Realidad_virtual]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 8
- [11] **Relidad Mixta**. [https://es.wikipedia.org/wiki/Realidad_mixta]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 8

- [12] **Pokémon Go!** [<https://pokemongolive.com/es/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 10
- [13] **Pokémon.** [<https://es.wikipedia.org/wiki/Pokemon>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 10
- [14] **Kit de desarrollo de software.** [https://es.wikipedia.org/wiki/Kit_de_desarrollo_de_software]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 11
- [15] **SLAM.** [https://es.wikipedia.org/wiki/Localizaci\penalty\OM\hskip\z@skip\unhbox\voidb@x\bgroup\let\unhbox\voidb@\x\setbox@\tempboxa\hbox{o\global\mathchardef\accent@\spacefactor\spacefactor}\accent19o\egroup\spacefactor\accent@\spacefactor\penalty\OM\hskip\z@skip\spacefactor\sfcode'on_y_modelado_simult\penalty\OM\hskip\z@skip\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox@\tempboxa\hbox{a\global\mathchardef\accent@\spacefactor\spacefactor}\accent19a\egroup\spacefactor\accent@\spacefactor\penalty\OM\hskip\z@skip\spacefactor\sfcode'aneos]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 11
- [16] **Node.js.** [<https://nodejs.org/en/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 12
- [17] **Chrome V8.** [https://es.wikipedia.org/wiki/Chrome_V8]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 12
- [18] **Event Loop.** [<https://nodejs.org/de/docs/guides/event-loop-timers-and-nexttick/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 12
- [19] **MongoDB.** [<https://www.mongodb.com/what-is-mongodb?lang=es-es>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 14
- [20] **Tecnología NoSQL** [<https://es.wikipedia.org/wiki/NoSQL>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 14
- [21] **Formato BSON.** [<https://es.wikipedia.org/wiki/BSON?lang=es-es>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 14
- [22] **El formato de texto JSON** [<https://es.wikipedia.org/wiki/JSON>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 14

- [23] **PaaS.** [*https://en.wikipedia.org/wiki/Platform_as_a_service*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 15
- [24] **Instalación Heroku.** [*<https://devcenter.heroku.com/articles/heroku-cli>*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 16
- [25] **Amazon Web Services.** [*<https://aws.amazon.com/es/>*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 16
- [26] **Azure.** [*<https://azure.microsoft.com/es-es/>*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 16
- [27] **Google Cloud.** [*<https://cloud.google.com/?hl=es>*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 16
- [28] **Google Maps API.** [*<https://developers.google.com/maps/documentation/android-api>*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 16
- [29] **Aprendizaje ubicuo.** [*https://es.wikipedia.org/wiki/Aprendizaje_ubicuo*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 18
- [30] **Splash Screen.** [*https://en.wikipedia.org/wiki/Splash_screen*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 23
- [31] **Navigation Drawer.** [*<https://material.io/design/components/navigation-drawer.html>*]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 23