

Aplicaciones para dispositivos móviles utilizando frameworks de software libre: Caso de estudio IONIC y CAPACITOR

Ángel Salvador López-Vásquez^{1*}, José Francisco Delgado-Orta¹, Jorge Ochoa-Sommano¹, Omar Antonio Cruz-Maldonado¹, Ángel Antonio Ayala-Zúñiga¹, María Alejandra Menéndez-Ortiz¹, Juan Mario Martínez-Ruiz², Imelda Perales-Ambrosio², Karla Cristina Pacheco-De la Paz² & Marco Polo Reyes-Barrera²

Resumen

El gran auge que tiene el mundo tecnológico ha llevado a vivir un futuro sin igual que hace unos pocos años ni siquiera podíamos imaginar. Actualmente las personas cuentan con un dispositivo inteligente, además, todo lo que se ve físicamente alrededor puede cumplir ciertos criterios tecnológicos. Lo que se tiene hoy en día va más allá de una simple aplicación, se trata de aplicaciones híbridas. El desarrollo de aplicaciones híbridas poco a poco ha incursionado en esta era digital. Las ventajas de usar estas aplicaciones son significativas, así como su producción, proceso en el que se requiere hoy en día un tiempo menor debido a que pueden usarse marcos de trabajo (frameworks) para ello, facilitando su diseño y construcción, como por ejemplo Ionic, el cual recientemente ha salido a la luz. Sobre la disponibilidad que puede ofrecer Ionic, se logran identificar las características principales que lo convierten en una alternativa de programación adaptable para los usuarios, ya que puede tener una gran ventaja en la creación de aplicaciones móviles y web, además esta tecnología puede ser implementada en cualquier sistema operativo móvil que se esté utilizando en el mercado laboral, y ofrecer la disminución de tiempos de construcción, la reutilización de códigos de programación, es sin duda el futuro para los desarrolladores.

Palabras clave: Apps, Ionic, Aplicaciones híbridas, fuente abierta, Apache, Cordova.

Recibido: 19 de enero de 2022.

Abstract

The great boom in the technological world has led to a future without the same as a few years ago we couldn't even imagine. Nowadays many people has a smart device, applying the technology to perform physical tasks according to some technological criteria. What you have today goes beyond a simple application, it's hybrid applications. The development of hybrid applications has been gradually entered this digital age. The advantages of using these applications are significant, as well as their production, a process in which less time is required today because frameworks can be used to facilitate their design and construction, such as Ionic, which has recently launched. Availability that can Ionic deliver allows to identify the main features that make it an adaptive programming alternative for users who may have a great prominence in the construction of mobile and web applications. Additionally, the use of this technology can be implemented on any mobile operating system used in the labor market, giving place to lower programming costs and the reuse of programming code, being undoubtedly the future for software developers.

Key words: Ionic, framework, hybrid applications, open source, Apache, Cordova.

Aceptado: 21 de abril de 2022.

¹ Instituto de Industrias, Universidad del Mar campus Puerto Escondido. Ciudad Universitaria, Vía Sola de Vega km 1.5 Carretera Puerto Escondido- Oaxaca. San Pedro Mixtepec-Juquila, Oaxaca, México, 71980.

² Licenciatura en Informática, Universidad del Mar campus Puerto Escondido. Ciudad Universitaria, Vía Sola de Vega km 1.5 Carretera Puerto Escondido- Oaxaca. San Pedro Mixtepec-Juquila, Oaxaca, México, 71980.

* **Autor de correspondencia:** angel.vasquez@zicatela.umar.mx (ASLV)

Introducción

En la actualidad, la tecnología móvil ha crecido considerablemente. Es por eso que las aplicaciones se deben adaptar a los cambios que se presenten en los teléfonos inteligentes (smartphones), haciendo énfasis en los sistemas operativos más populares como Android e iOS.

Una solución para la mejora en el desarrollo de aplicaciones para los sistemas operativos antes mencionados, consiste en utilizar herramientas que permitan reducir el tiempo, mantener ordenada la información y reutilizar el código fuente; es por ello que se pueden encontrar aplicaciones nativas, web y por último, una combinación entre éstas, las denominadas “aplicaciones híbridas”.

Las aplicaciones híbridas, también conocidas como web apps, están optimizadas para tener una mejor apariencia como parte de su diseño de interfaz a través de las pantallas de los dispositivos móviles. Una vez que se implementan estas aplicaciones en el smartphone tienen acceso a las capacidades de cada dispositivo, como pueden ser la cámara, el micrófono, la geolocalización, el navegador, entre otras. La comunicación entre la aplicación y los recursos del dispositivo, se realiza mediante plugins o marcos de trabajo (frameworks) que sirven como interfaz entre ambos, dando como resultado la apariencia de estar usando una aplicación nativa, con casi los mismos tiempos de respuesta entre las acciones.

Lo que se destaca de las aplicaciones híbridas es que trabajan con tecnologías web como *HTML5* (*HyperText Markup Language*), *CSS* (*Cascading Style Sheets*) y *JavaScript*. Para el acceso a una aplicación híbrida se utiliza un navegador web, y en caso de no contar con el servicio de Internet, ésta puede ser ejecutada

offline, siendo otra ventaja de este tipo de aplicaciones.

Para instalarlas en los dispositivos móviles, se puede utilizar el mismo procedimiento que en las aplicaciones nativas (a través de una tienda de aplicaciones). Sin embargo, gran parte de estas aplicaciones se diseñan utilizando páginas web (Goicochea 2013). Por lo que se puede decir que las web apps son aquellas aplicaciones desarrolladas con el entorno nativo del dispositivo móvil y con el lenguaje de marcado de hipertextos (*HTML5*).

Existen varios marcos de trabajo que permiten el desarrollo de aplicaciones híbridas, siendo *Ionic* el que ha tenido mayor auge, ya que se basa actualmente en Capacitor y el soporte de Cordova para el desarrollo multiplataforma de aplicaciones.

Capacitor permite crear aplicaciones nativas web, proporcionando un enfoque moderno de contenedor nativo para los equipos que desean construir primero en la web sin sacrificar el acceso completo a los *SDK* (*Software Development Kit*) nativos cuando lo necesitan.

Cordova, antes *PhoneGap*, es una plataforma que permite la ejecución de aplicaciones híbridas mediante el desarrollo web, éstas pueden ser implementadas en distintas plataformas operativas entre las más conocidas se encuentran: *Android*, *iOS* y *Windows Phone*.

Cabe resaltar que todas las herramientas y recursos que utiliza *Ionic* son gratuitos, se pueden encontrar manuales y sitios oficiales para las personas que quieran incursionar en el desarrollo de aplicaciones móviles híbridas con *Ionic*, *Capacitor* y *Cordova*.

El framework Ionic

Ionic fue creado por Max Lynch, Ben Sperry y Adam Bradley de Drifty Co en el año 2013, es open source por lo cual es completamente gratis, además ofrece una librería ideal para desarrollo móvil, empleando HTML5, CSS y componentes JavaScript con la finalidad de construir aplicaciones muy interactivas y agradables para los clientes o usuarios finales (Ionic, 2013).

Este *framework* es atractivo y amigable para los programadores, y sobre todo para los usuarios finales, ya que permite crear apariencias llamativas e innovadoras durante la construcción de las interfaces gráficas de las aplicaciones. También existe una comunidad de desarrolladores que cada día trabaja en mejorar este framework con el objetivo de mejorar la experiencia en el desarrollo de aplicaciones, para ello se emplea Angular, React y Vue, todas estas herramientas ajustadas en un mismo paquete, logrando una buena arquitectura de desarrollo, diseño e integración con dispositivos con Android o iOS.

¿Cómo trabaja IONIC para aprovechar los servicios de los dispositivos móviles?

Ionic permite trabajar de forma óptima aprovechando de la mejor manera las capacidades que posee un dispositivo móvil. Lo anterior mediante la disposición de *plugins* que ofrece *Capacitor* y *Cordova*, desarrollados por la misma empresa y la comunidad. En la tabla I se muestran las características que pueden aprovechar de las tecnologías de desarrollo.

De la Tabla I, se observa que una aplicación híbrida posee casi las mismas características de una aplicación nativa, excepto en que una aplicación nativa posee un acceso completo a la API del sistema operativo. Aún así, una aplicación híbrida

posee el acceso al resto de las características del dispositivo móvil. En lo que respecta a los entornos de ejecución de los tres tipos, (Denko *et al.* 2021) establecen que una aplicación híbrida que trabaja con un framework como Ionic requiere de un menor factor de utilización de los recursos del hardware de cómputo (la CPU o unidad central de procesamiento y la memoria principal del dispositivo móvil) que una aplicación nativa para las tareas elementales de desarrollo móvil, como son compilación, ejecución y despliegue de las interfaces gráficas de usuario.

Una aplicación construida usando Ionic puede realizar diversas tareas, por ejemplo: acceder a las propiedades de la cámara del dispositivo, leer códigos QR o de barras, enviar documentos a imprimir, permitir el servicio de geolocalización, entre muchas más. Hay páginas oficiales en las que se pueden encontrar *plugins* desarrollados para Capacitor (Capacitor 2021, Github 2021) o Cordova (Rivas 2015), según las necesidades que se tengan.

Tabla I. Características de las aplicaciones para teléfonos móviles inteligentes.

Característica	Web	Híbrida	Nativa
Conexión de red	✓	✓	✓
Implementa en App store		✓	✓
Completo acceso a la API del sistema operativo			✓
Imagen/video de la cámara	✓	✓	✓
GPS	✓	✓	✓
Acceso a contactos		✓	✓
Comunicación entre aplicaciones	✓	✓	✓
Almacenamiento local	✓	✓	✓
Notificaciones		✓	✓

Herramientas y plataformas del framework IONIC

Capacitor

Capacitor es un ejecutor nativo multiplataforma que facilita la creación de aplicaciones web modernas que se ejecutan de forma nativa en *iOS*, *Android* y la *Web*, representando la próxima evolución de las aplicaciones híbridas. Lo anterior sirve de ayuda a quienes desean construir primero en la web cuando lo necesitan sin sacrificar el acceso completo a los SDK nativos.

Apache CORDOVA

Es una plataforma para el desarrollo de aplicaciones nativas para móviles, la cual usa *HTML*, *JavaScript* y *CSS*. Cuenta con un conjunto de APIs (*Application Programming Interfaces*) que permiten la construcción de aplicaciones híbridas para distintos sistemas operativos, como *Android*, *iOS*, *Windows Phone* o *Blackberry*.

¿Cómo trabaja Cordova?

Apache Cordova trabaja como en una capa que se comunica por medio de sus propias APIs con el dispositivo móvil y, como resultado se puede acceder a la tecnología que está al alcance.

Una aplicación que se construya usando Cordova queda finalmente empaquetada, es decir, con una extensión que se instala en el dispositivo, por ejemplo, en el caso de *Android* un *APK*. La función de Cordova consiste en traducir el código de los plugins utilizados a código nativo que entiende el sistema operativo.

Para acotar, lo más importante en el desarrollo de una aplicación con Cordova e Ionic es que su implementación es única para cada plataforma operativa, se trabaja de una forma y las librerías de Cordova se encargan de convertir la aplicación para

los distintos sistemas operativos móviles. Entonces, se puede decir que se realiza un solo trabajo para N sistemas.

Angular

Es un framework de código abierto desarrollado por *Google* para la construcción de aplicaciones web sobre el lenguaje *JavaScript* con programación del lado del cliente (Basalo 2014).

Angular utiliza el patrón *MVC* (Modelo-Vista-Controlador) para el desarrollo de aplicaciones web dinámicas, extendiendo la sintaxis de *HTML* para hacerlo más funcional. El servidor provee el contenido estático, también conocido como plantillas, de igual forma la información que se va a mostrar (modelos) y, el cliente se encarga de unir esa información y contenido, dando como resultado la vista.

Angular cuenta con un gestor de paquetes denominado *Bower*, el cual permite instalar dependencias de otros desarrolladores a la aplicación que se esté desarrollando, de manera segura y actualizada. Ayuda al desarrollador en la parte front-end de la aplicación web.

Bower se ejecuta sobre *Git* (*software* de control de versiones), dependiente de *npm* (gestor de paquetes) y de *Node.js*.

Node.JS

Es un *framework* que nació en el 2009, que permite implementar operaciones de entrada/salida en sistemas informáticos. *Node.js* es *JavaScript* que se ejecuta en el lado del servidor, y trabaja de manera asíncrona para dar respuestas rápidas desde la petición de un cliente a un servidor, sus características se pueden ver en la tabla II.

Tabla II. Características de *AngularJS* en el diseño de aplicaciones web.

Característica	Descripción
Escalabilidad	Con simples aplicaciones web se pueden implementar muchas funciones en cualquier momento.
Tiempo	Permite desarrollar de manera ágil aplicaciones web, portables y con atributos de movilidad.
Multiplataforma	Construcción de aplicaciones dinámicas que sean compatibles con cualquier plataforma operativa.
Menos código	Las aplicaciones son diseñadas en modo single-page application (SPA), utilizando <i>HTML</i> , <i>CSS</i> y <i>JavaScript</i> .
Soporte	Existe una amplia comunidad y saben las necesidades que deben incluirse en el nivel de <i>AngularJS</i> .

El código JavaScript que corre en el backend es ejecutado gracias a la máquina virtual de Google, llamada V8 que es utilizada por el navegador Google Chrome. En resumen, Node.js se compone de un entorno de ejecución y de una librería.

Además de ser un framework de código abierto, se ejecuta en distintos sistemas operativos, posee un gran rendimiento y permite la escalabilidad de las aplicaciones que se construyan con las tecnologías del lado del servidor.

GIT

Es un *software* que permite controlar las versiones de un proyecto, fue desarrollado por *Linus Torvalds*. Básicamente *Git* ayuda a los desarrolladores de sistemas a mantener los archivos de las aplicaciones organizados, incluso se puede trabajar de manera distribuida con integrantes de un equipo de programadores.

Las funciones de *Git* permiten auditar el código fuente, es decir, controlar quién ha realizado cambios, en que instancia de tiempo y que línea se ha modificado. Si es necesario, *Git* permite volver a una versión anterior para restaurar el proyecto antes de algún error o cambio.

Contar con esta tecnología ayuda a gestionar todo el contenido de una aplicación, administración o actualización de los paquetes que se requieran para un proyecto.

Razones para probar el framework Ionic

1- Si se está acostumbrado al desarrollo *web* y por cuestiones técnicas se pide que la misma aplicación sea realizada para una implementación en un dispositivo móvil con un diferente lenguaje de programación o incluso para *Android* nativo; para evitar esos inconvenientes, se desarrolla una sola vez y con la ayuda de *Ionic* todas las aplicaciones que se creen correrán en diferentes plataformas operativas.

2- Las principales herramientas con la que un desarrollador debe estar familiarizado son *JavaScript*, *HTML* y *CSS*. Lo mencionado anteriormente con las mismas características como si se construyera una aplicación *web*, la única diferencia señalada será la utilización de componentes de *Ionic*, denominados tags, los cuales resultan muy intuitivos para ser utilizados y emulados en cualquier sistema operativo móvil existente.

3- En la realización de aplicaciones híbridas, cuando un cliente pide que la aplicación sea ejecutada y visualizada en cualquiera de los sistemas operativos móviles (*Android* e *iOS*), con *Ionic*, *Capacitor* y *Cordova* se podrán reducir los costos y tiempos de desarrollo del *software* a partir de la reutilización del código fuente, para que el desarrollador pueda entregar su trabajo respetando el cronograma establecido, con lo cual, se beneficiará al cliente a la hora de estimar o presupuestar los costos.

4- Una de las grandes ventajas que tiene, es que no exige tantos recursos en cuanto a rendimiento y disponibilidad para su instalación, además, el rendimiento que ofrecen las aplicaciones creadas con *Ionic* permite que sean muy estables al momento de ser ejecutadas.

Ventajas de framework Ionic

- Tiene un número de componentes propios muy importante, estos se deben de tener en cuenta en su utilización sobre las aplicaciones móviles nativas desarrolladas en *HTML5* que se obtendrá o utilizará en las aplicaciones híbridas.
- El MVC (*Model-View-Controller* o *Modelo-Vista-Controlador*), es un patrón de diseño que separa los datos, la lógica y las interfaces de usuario (Culoccioni 2015).
- Desarrollar aplicaciones híbridas en un único código propicia un menor tiempo de desarrollo y hace que su mantenimiento y escalado sea más sencillo. El desarrollo de una sola aplicación con un código único para distintas plataformas resulta menos costoso que el desarrollo de una aplicación nativa.
- Las Aplicaciones que se desarrollan son multiplataforma con un único desarrollo.

- Gracias a los componentes que se describieron anteriormente, permite realizar aplicaciones escalables, funcionales, construir código más rápido, interactuar con recursos del dispositivo y, sin costo al adquirir las herramientas de desarrollo con las cuales trabajar.

- El uso de *Ionic* permite crear, construir, y compilar aplicaciones en cualquier plataforma, todo con un solo comando. Por eso se considera una potente CLI (Command Line Interface).

- Al ser software libre, cuenta con una extensa comunidad para la mejora continua a los posibles cambios que se tenga en los próximos años.

Resultados

Con los conceptos que se tienen sobre *Ionic* y los componentes necesarios, a continuación, se enlistan los elementos que se tienen que utilizar para desarrollar una aplicación híbrida:

- Android SDK (Google, 2019).
- Java JDK 1.8 (Oracle, 2021).
- Node/NPM v14.16.0 (MIT, 2021).
- Apache Cordova 10.0.0 (Apache, 2021).
- Capacitor 2.0 (Capacitor, 2021b).
- Ionic Framework 5.6.3 (Drifty, 2021).

Es muy importante resaltar que todas las herramientas son open source, por lo tanto, se puede consultar en línea todo sobre ellas: documentación, instalación, configuración, etc. De igual manera el uso de un sistema operativo libre.

El editor de texto utilizado para trabajar en el código de la aplicación puede ser el de la elección del desarrollador, siempre y cuando sea gratuito. Entre los editores

más populares se encuentran *Sublime*, *Atom*, *Visual Studio Code* y *Brackets*.

Instalación del framework Ionic

La instalación del marco de trabajo *Ionic* se realiza con la siguiente instrucción:

```
npm install -g @ionic/cli
```

Toda la documentación se puede encontrar en el portal oficial de Ionic Framework (Ionic 2021).

Construcción de una aplicación con el framework Ionic

Al finalizar la instalación de todas las herramientas, se debe crear un directorio en donde se alojarán todos los proyectos que se desarrollen bajo Ionic, para ello, es suficiente con escribir el siguiente comando:

```
ionic start [nombre del proyecto] [plantilla] [type=angular || react || vue] [-capacitor]
```

Nombre de plantillas aceptados en *Ionic 5* y sus *framework* o bibliotecas, excepto los últimos 2 no aceptados en *Vue*

- *tabs*
- *blank*
- *sidemenu*
- *list*
- *my-first-app*
- *conference*

El proyecto se deberá crear usando una estructura muy ordenada, creando las librerías y los archivos necesarios para compilar la aplicación. Luego, se procede a agregar las plataformas para cada sistema operativo móvil, *Android* e *iOS* son los principales sistemas con los que trabaja *Ionic*.

```
ionic cap add ios  
ionic cap add android
```

Con estas instrucciones se le comunica a *Ionic* que prepare las configuraciones necesarias para compilar una aplicación en un determinado sistema operativo (caso de *iOS* y *Android* para los comandos previos *ionic cap*). Una vez que se ejecutan ambos comandos, se generará una estructura de la aplicación similar a la de la figura 1.

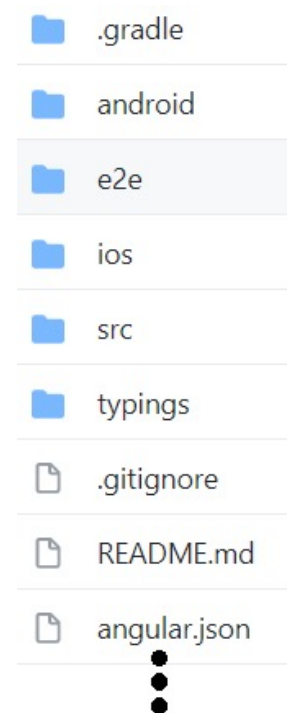


Figura 1. Estructura de un proyecto en *Ionic*.

Para hacer uso de una base de datos u otros *scripts*, es necesario implementar los diferentes plugins que ofrece *Capacitor* y *Cordova* así como *APIs* existentes.

Instalación de los complementos necesarios

Antes que nada, se debe tener instalado el *software* de control de versiones Git (disponible en: <https://git-scm.com/>

downloads), el cual permite la descarga de los archivos que la aplicación va a requerir para su funcionamiento.

Compilación de la aplicación

El framework Ionic ofrece varias formas para visualizar una aplicación híbrida, las cuales pueden ser: mediante su propio servidor web y desde un dispositivo móvil que esté conectado a la computadora en donde se encuentre el proyecto.

- Servidor web

Se puede visualizar el avance del desarrollo de la aplicación en tiempo real mediante un navegador con el comando:

```
ionic serve
```

Al contar con su propio servidor web, se asigna un puerto, el cual se puede cambiar, o bien, mantener el que se establece por defecto (<http://localhost:8100>).

- Compilación

Se debe compilar la aplicación para generar las carpetas de los sistemas operativos, con el comando:

```
ionic build
```

Con *Live Reload* se puede ver la versión final o de prueba de una app y al mismo tiempo modificarla, los cambios se verán reflejados como si se estuviera trabajando en un navegador *web*.

```
ionic cap run android || ios -l --external
```

El dispositivo debe estar en modo desarrollador, por lo que en caso de no ser así, el sistema operativo Android permite activarlo a través de la opción Información del dispositivo del menú Ajustes, en donde se encuentra el número de compilación, se pulsa 7 veces (*Google* 2021).

Conclusiones

El desarrollo de aplicaciones móviles híbridas recoge lo mejor de cada paradigma de programación: nativa y *web*. El aprovechar tecnologías como *Ionic*, *Capacitor* y *Cordova*, permite crear proyectos funcionales, optimizando recursos que casi siempre no se pueden controlar como: tiempo, costo, calidad, entre otros. El adquirir nuevos conocimientos con base en esta investigación, será de gran importancia y aporte a las comunidades que se encuentran trabajando para popularizar el uso del framework Ionic en teléfonos inteligentes.

La documentación, herramientas, librerías, manuales y demás recursos, se encuentran en las páginas oficiales de las plataformas, por lo que haciendo uso del Internet, se pueden despejar dudas, recoger experiencias de otros desarrolladores y, entre otras cosas, estar preparado para cualquier eventualidad que se presente a la hora de desarrollar una aplicación móvil híbrida, cualquiera que ésta sea.

Es posible desarrollar aplicaciones para distintos dispositivos móviles de forma ágil y rápida utilizando un lenguaje que muchos de los programadores web conocen como *HTML* y *JavaScript*, además de que también está implementado *Bootstrap*, *CSS* y *SASS* (*Syntactically Awesome Style Sheets*), lo cual es una ventaja que se ofrece para la construcción de un diseño *web* y móvil.

Agradecimientos

Al Programa para el Desarrollo Profesional Docente para el Tipo Superior (PRODEP), por el apoyo y las facilidades brindadas en el marco del proyecto de fortalecimiento de cuerpos académicos (UMAR-CA-38), convocatoria 2020. A los estudiantes de

la Licenciatura en Informática, colaboradores del cuerpo académico de Sistemas Inteligentes (UMAR-CA-38), por su esfuerzo y dedicación en la elaboración del presente trabajo. A un árbitro anónimo quien realizó comentarios y sugerencias para enriquecer nuestro manuscrito.

Referencias

- Apache. 2021.** Portal oficial de Apache Cordova. Obtenido de: <https://cordova.apache.org/>.
- Basalo, A. 2014.** Obtenido de <http://www.desarrolloweb.com/articulos/que-es-angularjs-descripcion-framework-javascript-conceptos.html>
- Capacitor. 2021a.** Official Plugins. Obtenido de <https://capacitorjs.com/docs/apis>.
- Capacitor. 2021b.** Portal oficial de Capacitor. <https://capacitorjs.com/docs/getting-started/with-ioni>.
- Culoccioni, S. 2015.** Desarrollo de aplicaciones móviles con Ionic Framework. Obtenido de <http://www.solvetic.com/tutoriales/article/1607-desarrollo-de-aplicaciones-m%C3%B3viles-con-ionic-framework/>.
- Denko, B., S. Pecnik & I. Fister. 2021.** A Comprehensive Comparison of Hybrid Mobile Application Development Frameworks. International Journal of Security and Privacy in Pervasive Computing. Vol. 13, no.1, pp:78-90.
- GitHub. 2021.** Capacitor Community. Obtenido de <https://github.com/capacitor-community>
- Google. 2019.** Portal oficial de descarga del SDK de Android. Obtenido de: <https://developer.android.com/studio>.
- Google. 2021.** Portal comunidad Android. (Junio de 2021). Cómo activar las opciones de desarrollo en Android. Obtenido de <https://developer.android.com/studio/debug/dev-options?hl=es-419>.
- Goicochea, A. 2013.** Obtenido de <http://anibalgoicochea.com/2013/07/26/aplicaciones-nativas-aplicaciones-web-y-aplicaciones-hibridas/>.
- Ionic Drifty, C. 2013.** Ionic. Obtenido de <http://ionicframework.com>.
- Ionic Drifty, C. 2021.** Portal oficial de Ionic Framework. Obtenido de: <https://ionicframework.com/docs/intro/cli>
- MIT. 2021.** Portal oficial de descarga de Node/NPM. Obtenido de <https://nodejs.org/es/download/>.
- Oracle. 2021.** Portal oficial de Java JDK. Obtenido de: <https://www.oracle.com/java/technologies/javase-downloads.html>.
- Rivas, J. J. 2015.** Cómo empezar con Ionic Framework. Obtenido de <http://www.phonegapSpain.com/que-es-y-como-empezar-con-ionic-framework/>.