



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: “APLICACIÓN MÓVIL PARA INFORMAR, UBICAR Y DIRIGIR  
HACIA SITIOS DISPONIBLES DE APARCAMIENTO PARA LOS  
INTEGRANTES DE LA COMUNIDAD UNIVERSITARIA ESPE”**

**AUTORES: REASCOS GONZÁLEZ, JORGE ISRAEL  
TORRES LÓPEZ, JONATHAN ALEJANDRO**

**DIRECTOR: ING. MARCILLO PARRA, DIEGO MIGUEL**

**SANGOLQUÍ**

**2020**



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, ***“APLICACIÓN MÓVIL PARA INFORMAR, UBICAR Y DIRIGIR HACIA SITIOS DISPONIBLES DE APARCAMIENTO PARA LOS INTEGRANTES DE LA COMUNIDAD UNIVERSITARIA ESPE”*** fue realizado por los señores ***Reascos González, Jorge Israel y Torres López, Jonathan Alejandro***, el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 26 de febrero del 2020

-----  
**ING. MARCILLO PARRA, DIEGO MIGUEL**

C. C. 171080292-5



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Nosotros, *Reascos González, Jorge Israel y Torres López, Jonathan Alejandro*, declaramos que el presente trabajo de titulación “**APLICACIÓN MÓVIL PARA INFORMAR, UBICAR Y DIRIGIR HACIA SITIOS DISPONIBLES DE APARCAMIENTO PARA LOS INTEGRANTES DE LA COMUNIDAD UNIVERSITARIA ESPE**” es de nuestra autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 26 de febrero del 2020

Reascos González, Jorge Israel

C.C. 100376251-3

Torres López, Jonathan Alejandro

C.C. 172234013-8



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**AUTORIZACIÓN**

Nosotros, **Reascos González, Jorge Israel y Torres López, Jonathan Alejandro**, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación “**APLICACIÓN MÓVIL PARA INFORMAR, UBICAR Y DIRIGIR HACIA SITIOS DISPONIBLES DE APARCAMIENTO PARA LOS INTEGRANTES DE LA COMUNIDAD UNIVERSITARIA ESPE**” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 26 de febrero del 2020

**Reascos González, Jorge Israel**

C.C. 100376251-3

**Torres López, Jonathan Alejandro**

C.C. 172234013-8

## DEDICATORIA

*Dedico este trabajo a mi papá, ya que gracias a su ejemplo, enseñanzas y consejos a lo largo de mi vida me han permitido crecer personal y académicamente, inculcándome con su ejemplo que con esfuerzo, valentía y trabajo duro todo es posible.*

*A mi mamá, por confiar y cuidar de mí en cada decisión que he tomado a lo largo de este tan arduo proceso, por saberme guiar con su sabiduría en virtud de valores como el respeto, humildad, responsabilidad y honradez valores que me han ayudado a cumplir un sueño más.*

*A mis hermanas, tíos y primos, y a toda mi familia ya que con su cariño inmenso y apoyo incondicional en este proceso que he conseguido culminarlo con éxito, ya que con sus consejos y palabras de aliento lograron hacer de mí una mejor persona.*

*Finalmente quiero dedicar este trabajo a todos mis amigos, ya que siempre estuvieron cuando más los he necesitado brindándome su apoyo, cariño y consejos para sobrellevar la carga que implica este proceso.*

*Jorge Reascos*

*A mi papá por el gran ejemplo que día a día me brinda, ha sido el pilar fundamental para seguir adelante en busca de cumplir tan anhelado objetivo, que gracias a sus oportunos consejos pude salir adelante y superarme en la vida. Por los esfuerzos que realiza para sacar adelante a nuestra familia.*

*A mi madre por brindarme su amor y apoyo incondicional, quién a pesar de todas mis fallas y errores siempre me dio la mano ayudándome a ser una mejor persona, ha sido la motivación e inspiración para seguir en este proceso de formación y así poder superarme tanto en mi vida personal como profesional.*

*A mis hermanos, tíos, primos, abuelos y toda mi familia que con sus consejos y palabras de motivación me impulsaron y me animaron a culminar este proceso de preparación académica.*

*Finalmente, a todos mis amigos que siempre estuvieron brindándome su apoyo en los momentos necesarios para no decaer en la búsqueda de conseguir este logro tan anhelado.*

*Alejandro Torres*

## AGRADECIMIENTO

*Al finalizar este trabajo, quiero utilizar este espacio para agradecer principalmente a Dios, por bendecirme a mí y a toda mi familia, ayudándonos a superar cada problema que surge frente a nosotros con la sabiduría y fortaleza necesaria para tomar las decisiones correctas.*

*A mis padres Jorge y Germania, ya que con su ejemplo y cariño incondicional supieron formarme y guiarme por el camino correcto haciendo de mí una mejor persona y un gran profesional en virtud del cumplimiento de valores, principios y ética, además de su infinito amor, cariño y paciencia que me brindaron en los momentos buenos y malos que atravesé durante todo el trayecto universitario.*

*A mis hermanas, tíos y primos que siempre me apoyaron y creyeron en mí, brindándome siempre palabras de aliento que me impulsaban a continuar esforzándome cada día más para conseguir mis objetivos y metas propuestas.*

*A mis abuelitos Jorge y Pacífico, quienes ya partieron de este mundo, pues siempre me enseñaron que toda meta y objetivo propuesto es posible, con trabajo duro, honrado, honesto y humilde. A mis abuelitas Rosa y Marina, quienes siempre me enseñaron que toda recompensa es satisfactoria si se la realiza con amor, cariño y paciencia.*

*A mi enamorada Vanessa Salazar, quien siempre me acompañó a lo largo de toda mi trayectoria universitaria, siendo un apoyo fundamental e incondicional en mi vida profesional y personal, quien con su cariño y afecto me enseñó que todo lo que me planteo a realizar lo puedo lograr.*

*A todos mis amigos con los que compartí valiosos momentos y que han sido una compañía invaluable para alcanzar esta meta, agradecer especialmente a Santiago y Diego de quienes sin esperar nada a cambio siempre estuvieron cuando los necesité compartiendo sus conocimientos, alegrías y tristezas.*

*Finalmente agradecer de manera especial por su esfuerzo, colaboración, sabiduría y apoyo total, a mi tutor de tesis, el Ing. Diego Marcillo, quien nos brindó una guía oportuna para el desarrollo y culminación de este trabajo.*

*Jorge Reascos*



*En primer lugar, agradecer a Dios por la oportunidad de culminar con éxito una etapa más de formación académica, por darme la sabiduría suficiente para tomar decisiones importantes en mi vida que con perseverancia y esfuerzo ha sido posible lograr este objetivo.*

*A mis padres Luis y Emma por inculcarme principios y valores los cuáles son fundamentales en la formación de un ser humano, por el amor recibido, la dedicación y paciencia con la que cada día se preocuparon durante el avance en mi etapa de formación, y quienes han sido el principal motor de mis sueños, gracias a ellos por confiar en mí y creer en mis expectativas.*

*A mis hermanos Dillan y Jhosua quienes han estado siempre para mí, apoyándome en cada decisión tomada, alentándome a culminar mis metas y sobre todo acompañándome día a día en esta etapa de formación.*

*Un agradecimiento especial a mi enamorada Michelle quien desde el primer día que la conocí ha sido una persona fundamental en este proceso de formación, brindándome su apoyo incondicional y cariño en los momentos que más he necesitado, dándome empujones necesarios en los momentos claves para poder culminar con éxito este sueño tan anhelado.*

*A mi tutor de tesis, el Ing. Diego Marcillo, quien me brindo consejos y apoyo durante la etapa de formación académica y desarrollo de esta tesis, que gracias a sus conocimientos académicos brindados fue posible el desarrollo y finalización de este trabajo.*

*A mis abuelos, tíos, primos, amigos y demás personas que confiaron en mí, gracias a sus consejos, palabras de aliento, apoyo incondicional que de una u otra manera han contribuido en el logro de esta meta.*

*Alejandro Torres*

## ÍNDICE

CERTIFICADO DEL DIRECTOR .....	i
AUTORIA DE RESPONSABILIDAD.....	ii
AUTORIZACIÓN.....	iii
DEDICATORIA .....	iv
AGRADECIMIENTO .....	vi
ÍNDICE.....	x
ÍNDICE DE TABLAS.....	xviii
ÍNDICE DE FIGURAS .....	xix
RESUMEN .....	xxii
ABSTRACT .....	xxiii
<b>CAPÍTULO I</b>	
<b>INTRODUCCIÓN</b>	
1.1 Antecedentes.....	1
1.2. Problemática.....	2
1.3. Justificación .....	3
1.4. Objetivos .....	3
1.4.1. Objetivo General.....	3
1.4.2. Objetivo Específico .....	4

1.5.	Alcance.....	4
1.6.	Hipótesis.....	5
1.7.	Metodología de la investigación.....	5

## **CAPÍTULO II**

### **MARCO TEÓRICO**

2.1.	Introducción.....	9
2.2.	Dispositivos Móviles .....	9
2.2.1.	Definición.....	9
2.2.2.	Categorías de dispositivos móviles.....	9
2.2.3.	Tipos de dispositivos móviles .....	10
2.2.3.1.	Smartphone.....	10
2.2.3.2.	Feature Phones .....	11
2.2.3.3.	Tablets.....	12
2.3.	Sistemas operativos para dispositivos móviles .....	12
2.3.1.	Definición.....	12
2.3.2.	Tipos de sistemas operativos .....	13
2.4.	Sistema Operativo - Android.....	14
2.4.1.	Antecedentes.....	14
2.4.2.	Definición.....	14
2.4.3.	Componentes .....	14

2.4.4.	Características.....	17
2.4.5.	Evolución.....	19
2.5.	Metodologías de desarrollo móvil - Mobile D .....	20
2.5.1.	Definición.....	20
2.5.2.	Ciclo de vida de Mobile-D.....	20
2.6.	Patrones de arquitectura.....	22
2.6.1.	Modelo cliente - servidor.....	22
2.6.2.	Arquitectura orientada a servicios SOA .....	23
2.7.	Geolocalización .....	24
2.7.1.	Definición.....	24
2.7.2.	Tipos de geolocalización.....	25
2.7.2.1.	GPS.....	25
2.7.2.2.	GSM .....	25
2.7.3.	Archivos KMZ .....	27
2.8.	Lenguaje de marcado KML.....	28
2.8.1.	Antecedentes.....	28
2.8.2.	Definición.....	28
2.8.3.	Estructura .....	29
2.8.4.	Funcionalidad .....	30

2.9.	Base de datos No SQL .....	31
2.9.1.	Definición.....	31
2.9.2.	Bases NoSQL en comparación con NoSQL.....	34
2.9.3.	Clasificación .....	37
2.9.4.	MongoDB .....	38
2.9.4.1.	Antecedentes.....	38
2.9.4.2.	Definición.....	38
2.9.4.3.	Características.....	39
2.9.4.4.	Funcionamiento .....	40
2.10.	Framework de desarrollo móvil.....	40
2.10.1.	Definición.....	40
2.10.2.	Tipos.....	41
2.10.2.1.	IONIC.....	41
2.10.2.2.	Android Studio .....	42
2.10.2.3.	Xamarin .....	43
2.11.	Google Maps .....	45
2.11.1.	Definición.....	45
2.11.2.	Tipos de Geolocalización.....	45
2.11.3.	Ventajas y Desventajas .....	46

2.11.4. Generación de datos de la Geolocalización .....	47
2.12. Servicios REST (Web Services) .....	49
2.12.1. Definición.....	49
2.12.2. Métodos HTTP .....	49
2.12.3. Características.....	50
2.12.4. Ventajas .....	51
2.13. Metodología para crear una encuesta.....	52
2.13.1. Definición.....	52
2.13.2. Características.....	53
2.13.3. Tipos de encuestas.....	54
2.13.4. Ciclo de una encuesta .....	55

### **CAPÍTULO III**

#### **DESARROLLO**

3.1. Introducción .....	58
3.2. Fase I: Análisis de la situación actual .....	58
3.2.1. Búsqueda de trabajos relacionados con la detección de plazas disponibles para el parqueo de vehículos.....	59
3.2.2. Revisión sistemática de literatura con los trabajos obtenidos de los repositorios digitales académicos .....	60

3.2.3.	Definición de la estructura (software y hardware) del prototipo para el sistema de información .....	62
3.3.	Fase II: Evaluar herramientas hardware y software para la solución planteada.....	63
3.3.1.	Búsqueda de lenguajes de programación y modelado de entornos virtuales para desarrollo de aplicaciones móviles .....	63
3.3.2.	Ejecutar pruebas de rendimiento en cada una de las herramientas encontradas.....	65
3.3.3.	Selección de la herramienta de desarrollo móvil que sea adecuada para la construcción del prototipo .....	68
3.4.	Fase III: Diseño del prototipo .....	70
3.4.1.	Desarrollo de la interfaz de ingreso de los datos relacionados con el usuario ..	70
3.4.2.	Digitalización del espacio de parqueo seleccionado para la construcción del prototipo .....	77
3.4.3.	Programación para la detección y asignación de la plaza más cercana al área de trabajo del usuario .....	79
3.4.4.	Programación de la ruta que dirige al área asignada para el usuario.....	81
3.5.	Fase IV: Implementación del prototipo.....	81
3.5.1.	Primer plan de pruebas internas ejecutadas por el grupo de trabajo .....	81



3.5.2. Alojar en las tiendas de aplicaciones disponibles para los diferentes dispositivos móviles .....	82
3.6. Fase V: Validación del prototipo mediante pruebas .....	86
3.6.1. Ejecución de pruebas por parte de los usuarios del área de parqueo de vehículos que posee la Universidad ESPE .....	86
3.6.2. Aplicación de una encuesta para conocer el grado de satisfacción al utilizar el sistema de información alojado en una tienda móvil .....	86
3.6.2.1. Objetivos: .....	86
3.6.2.2. Población o Universo seleccionado: .....	87
3.6.2.3. Identificación de la información a recolectar .....	87
3.6.2.4. Diseño del cuestionario .....	88
3.6.2.5. Cálculo de la muestra .....	90

## **CAPÍTULO IV**

### **PRUEBAS Y RESULTADOS**

4.1. Introducción .....	93
4.2. Protocolo de validación .....	93
4.3. Primer plan de pruebas internas ejecutadas por el grupo de trabajo .....	93
4.4. Ejecución de pruebas por parte de los usuarios del área de parqueo de vehículos que posee la Universidad ESPE .....	109
4.5. Resultados .....	116

4.5.1. Análisis de resultados sobre la satisfacción del usuario.....116

4.5.2. Discusión de los resultados obtenidos en la fase de pruebas .....122

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

5.1. Introducción .....124

5.2. Conclusiones .....124

5.3. Recomendaciones .....125

5.4. Trabajo Futuro .....126

**BIBLIOGRAFÍA .....128**

**ANEXOS.....132**

## ÍNDICE DE TABLAS

<b>Tabla 1</b> <i>Características de Android</i> .....	17
<b>Tabla 2</b> <i>Versiones de Android y sus características</i> .....	19
<b>Tabla 3</b> <i>Comparación de los diferentes tipos de base de datos NoSQL</i> .....	32
<b>Tabla 4</b> <i>Base de datos NoSQL en comparación con base de datos SQL</i> .....	34
<b>Tabla 5</b> <i>Ventajas y desventajas de la Geolocalización</i> .....	46
<b>Tabla 6</b> <i>Modos de generación y recolección de datos de Geolocalización</i> .....	47
<b>Tabla 7</b> <i>Trabajos seleccionados de la plataforma SpringerLink</i> .....	60
<b>Tabla 8</b> <i>Prueba de normalidad en memoria IONIC vs Apache Cordova</i> .....	66
<b>Tabla 9</b> <i>Prueba de normalidad en CPU IONIC vs Apache Cordova</i> .....	67
<b>Tabla 10</b> <i>Tabla de rendimiento de memoria Android Studio vs Xamarin</i> .....	67
<b>Tabla 11</b> <i>Tabla de rendimiento CPU Android Studio vs Xamarin</i> .....	68
<b>Tabla 12</b> <i>Consumo de recursos RAM y CPU</i> .....	69
<b>Tabla 13</b> <i>Indicadores de evaluación para el framework IONIC</i> .....	69

## ÍNDICE DE FIGURAS

<b>Figura 1</b> Metodología Ad Hoc .....	8
<b>Figura 2.</b> Sistemas operativos móviles – Iconos.....	13
<b>Figura 3.</b> Arquitectura interna de Android.....	15
<b>Figura 4.</b> Ciclo de desarrollo Mobile-D.....	21
<b>Figura 5.</b> Diagrama Modelo Cliente-Servidor .....	23
<b>Figura 6.</b> Diagrama de arquitectura orientada a servicios .....	24
<b>Figura 7.</b> Estructura de la red GSM .....	27
<b>Figura 8.</b> Archivo KML creado desde Google Earth.....	31
<b>Figura 9.</b> Actuación de Xamarin en iOS y Android.....	44
<b>Figura 10.</b> Estructura básica de la arquitectura REST.....	49
<b>Figura 11.</b> Ciclo de una encuesta.....	55
<b>Figura 12.</b> Diagrama de arquitectura del prototipo propuesto .....	63
<b>Figura 13.</b> Prototipo pantalla inicial de la aplicación EspePark .....	71
<b>Figura 14.</b> Registro de datos del usuario .....	72
<b>Figura 15.</b> Prototipo de ventana de confirmación de datos de usuario.....	73
<b>Figura 16.</b> Ventana para el registro de datos del vehículo .....	74
<b>Figura 17.</b> Interfaz para el registro del lugar a visitar .....	75
<b>Figura 18.</b> Interfaz para el registro de la jornada laboral .....	76
<b>Figura 19.</b> Interfaz de inicio de sesión de usuarios .....	77
<b>Figura 20.</b> Herramienta Sitios de Google Maps .....	78
<b>Figura 21.</b> Digitalización de las entradas principales de la Universidad ESPE.....	79

<b>Figura 22.</b> Digitalización de los lugares de aparcamiento vehicular .....	79
<b>Figura 23.</b> Pantalla principal de la aplicación Espe Park.....	95
<b>Figura 24.</b> Registro de datos personales del usuario .....	96
<b>Figura 25.</b> Registro de datos personales del usuario .....	97
<b>Figura 26.</b> Registro de datos del vehículo.....	98
<b>Figura 27.</b> Confirmación de registro exitoso de los datos del vehículo.....	99
<b>Figura 28.</b> Registro del área, edificio, bloque o departamento a visitar .....	100
<b>Figura 29.</b> Registro de la jornada laboral del usuario .....	101
<b>Figura 30.</b> Confirmación de datos de usuario .....	102
<b>Figura 31.</b> Mensaje de bienvenido al usuario luego de iniciar sesión.....	103
<b>Figura 32.</b> Solicitar una plaza disponible de aparcamiento .....	104
<b>Figura 33.</b> Usuario solicita una plaza disponible de aparcamiento .....	105
<b>Figura 34.</b> Ruta emitida por la aplicación para llegar a la plaza disponible de aparcamiento.....	106
<b>Figura 35.</b> Usuario llega a la plaza de aparcamiento que fue asignada .....	107
<b>Figura 36.</b> El usuario confirma ocupa la plaza de aparcamiento asignada.....	108
<b>Figura 37.</b> El usuario desocupa la plaza de aparcamiento asignada .....	109
<b>Figura 38.</b> Icono de la aplicación móvil EspePark.....	110
<b>Figura 39.</b> El usuario solicita una plaza de aparcamiento disponible .....	111
<b>Figura 40.</b> Gráfica de la ruta hacia la plaza asignada .....	112
<b>Figura 41.</b> Ubicación de la aplicación en el coliseo "Gral. Iturralde" .....	113
<b>Figura 42.</b> Ubicación con el vehículo en el coliseo de la Universidad .....	113
<b>Figura 43.</b> Ubicación de la aplicación en la entrada principal de la Universidad .....	114

<b>Figura 44.</b> Ubicación con el vehículo en la entrada principal de la Universidad .....	114
<b>Figura 45.</b> Ubicación de la plaza de aparcamiento asignada .....	115
<b>Figura 46.</b> Usuario llega con el vehículo a la plaza de aparcamiento.....	116
<b>Figura 47.</b> Calificación de la iniciativa de crear la aplicación.....	117
<b>Figura 48.</b> Número de usuarios dispuestos a probar la aplicación .....	118
<b>Figura 49.</b> Tiempo empleado para la búsqueda de una plaza disponible por medio de la aplicación.....	119
<b>Figura 50.</b> Nivel de utilidad de la aplicación móvil.....	120
<b>Figura 51.</b> Usuarios que recomendaría el uso de la aplicación .....	120
<b>Figura 52.</b> Nivel de dificultad de la navegación en la aplicación móvil .....	121
<b>Figura 53.</b> Software móvil preferido por los usuarios de los parqueaderos de la Universidad .....	122

## **RESUMEN**

En Ecuador, la adquisición de un vehículo se ha convertido en un trámite fácil lo que conlleva al crecimiento del parque automotor generando un gran congestionamiento vehicular en la ciudad y repercute en la pugna por la obtención de un puesto libre de aparcamiento especialmente durante las horas pico. Este problema se ha evidenciado día tras día en la Universidad de las Fuerzas Armadas ESPE, situación que se ha presentado esencialmente a la apertura de la jornada laboral/académica, es decir en la mañana a las 07:00 y en la tarde a las 15:00. Razón importante para realizar un sistema de información enfocado a dispositivos móviles mediante la utilización de diferentes tecnologías de la información y la comunicación la cual recomienda un lugar disponible de aparcamiento cercano al lugar donde se desarrollan las actividades laborales. Las herramientas empleadas en el sistema de información son: IONIC, archivos KML y KMZ, Servicios REST. El prototipo presentó resultados positivos en cuanto a la aceptación del uso de aplicativos móviles, así como la optimización del tiempo de aparcamiento del vehículo. Finalmente, el uso del sistema de información brinda un servicio tecnológico y optimiza el tiempo del usuario.

### **PALABRAS CLAVES:**

- **APLICACIÓN MÓVIL**
- **KML**
- **KMZ**
- **IONIC**

## **ABSTRACT**

In Ecuador the acquisition of a vehicle has been facilitated so that with the growth of the vehicle fleet there has been a considerable increase in vehicle congestion in the city the same one has had repercussions on the struggle to obtain a free parking lot especially during rush hours. This problem has become evident all days at the Armed Forces University ESPE situation that has essentially occurred during rush hours when the administrative and academic work start such us 7o'clock in the morning and 3o'clock in the afternoon. The information system developed in this work recommends an available parking lot near the place where people are working optimizing the parking time to the user. The tools used in the information system are IONIC, KML and KMZ files, REST services. The prototype presented positive results in terms of acceptance; the use of mobile application is focused on providing this type of service as well as the optimization of vehicle parking time. Finally, the use of the information system encourages users to use the mobile applications that provide a technological service in order to optimize the users time.

### **KEY WORDS**

- **MOBILE APPLICATION**
- **KML**
- **KMZ**
- **IONIC**



# CAPÍTULO I

## INTRODUCCIÓN

En este capítulo se expone los antecedentes, los objetivos, el alcance delimitado para el desarrollo del proyecto.

### 1.1 Antecedentes

La Universidad de las Fuerzas Armadas ESPE, actualmente cuenta con un servicio de control de parqueadero denominado UrbaPark. Este se define como un grupo de vigilancia vehicular, más no de desarrolladores de soluciones integradas de estacionamientos y movilidad urbana para mejorar la calidad de vida de sus clientes, ofreciendo consultoría, diseño, operación, administración y construcción de estacionamientos.

Este servicio es limitado ya que el servicio que ofrecen es proveer de seguridad a los vehículos que se encuentran en el interior de la institución. Tanto la Universidad de las Fuerzas Armadas ESPE, como el servicio externo contratado por la institución carecen de la funcionalidad de informar a la comunidad universitaria sobre la existencia de lugares disponibles para estacionamiento de vehículos en los diferentes sectores de parqueo que están disponibles en la institución.

Actualmente en la Universidad de las Fuerzas Armadas ESPE y el servicio externo contratado UrbaPark, no está disponible el servicio de direccionamiento a una zona libre de parqueo, tampoco existe la asignación dinámica de lugares de aparcamiento de acuerdo a la zona en donde desarrollan la mayor parte de la jornada laboral o académica.

## **1.2. Problemática**

En la Universidad de las Fuerzas Armadas ESPE los docentes, administrativos, estudiantes y personal de servicio de la institución requieren diariamente un sitio disponible para estacionar su vehículo, sin embargo, se encuentran con el problema de la poca o nula información sobre la disponibilidad de estos sitios de aparcamiento que sean cercanos a sus lugares de trabajo, en el caso de docentes, administrativos y personal de servicio, o estudios, en el caso de los estudiantes, en los cuales desarrollan la mayor cantidad de sus actividades cotidianas.

La deficiencia de este tipo de información relacionada a plazas disponibles de aparcamiento genera una serie de problemas en las instalaciones de la Universidad. Estos problemas se dan específicamente en los parqueaderos, en donde se produce congestión vehicular, peligro de accidentes de tránsito por la pugna elevada de los espacios de aparcamiento, pérdida de tiempo, impaciencia en los conductores, atraso a las horas de clase (especialmente en las horas pico, 07:00 y 15:00) que son los horarios en los que inicia la jornada laboral, entre otros.

Adicional a la problemática que se vive a diario en la Universidad, hay que sumarle la falta de existencia un servicio externo o un software informático que se encargue de emitir este tipo de información que es bastante relevante y necesaria para las personas que asisten en sus vehículos a la Universidad, es decir, no existe un aplicativo que mantenga informado a los usuarios que asisten a la Universidad en su vehículo propio sobre sitios disponibles de aparcamiento.

### **1.3. Justificación**

La Universidad de las Fuerzas Armadas ESPE en la actualidad no tiene disponible un software de alerta temprana en el cual se brinde el servicio de informar, ubicar y dirigir a sitios disponibles de aparcamiento, razón por la cual se vio oportuno el desarrollar un prototipo de sistema de información enfocado a dispositivos móviles. Este sistema tiene como objetivo principal el mantener informado al usuario en forma precisa y oportuna cada uno de los servicios previamente mencionados.

Con el desarrollo e implementación del sistema de información enfocado al ambiente móvil se solucionó los diferentes tipos de problemas, por ejemplo, la congestión vehicular, peligro de accidentes por pugna de los espacios de aparcamiento, pérdida de tiempo, impaciencia en los usuarios debido a que deben estar dando varias vueltas alrededor del parqueadero con el fin de encontrar un lugar disponible, atraso a las horas de clase, entre otros.

Ante esta realidad, en el presente trabajo se desarrolló un prototipo de sistema de información enfocado a dispositivos móviles, en el cual se muestra información sobre la disponibilidad de sitios de aparcamiento y la gráfica de la ruta hacia el sitio de aparcamiento asignado.

### **1.4. Objetivos**

#### **1.4.1. Objetivo General**

Desarrollar un sistema de información enfocado a dispositivos móviles, mediante la utilización de diferentes tecnologías de la información y la comunicación, con el propósito de facilitar a los integrantes de la comunidad universitaria ESPE la ubicación

de espacios de aparcamiento disponibles y cercanos a las instalaciones en donde desarrollan sus actividades cotidianas.

#### **1.4.2.Objetivo Específico**

- Analizar trabajos relacionados con la detección de plazas disponibles de aparcamiento vehicular publicados en revistas y sitios académicos, mediante una revisión sistemática de literatura, con el fin de tener una perspectiva inicial para el desarrollo del sistema de información.
- Digitalizar la zona de aparcamiento ubicado en la parte posterior al Departamento de Ciencias de la Computación (DCCO), a través de la utilización de las herramientas tecnológicas actuales a fin de ofrecer un servicio de calidad al usuario final.
- Integrar en una aplicación móvil los productos obtenidos de las herramientas tecnológicas a fin de publicar un sistema estable y que permita al usuario final un uso continuo y sin inconvenientes.
- Validar el aplicativo móvil mediante una encuesta de satisfacción a los usuarios del parqueadero de la Universidad ESPE para determinar la eficiencia del sistema construido para la búsqueda de una plaza disponible.

#### **1.5.Alcance**

El proyecto de desarrollo solucionó a la problemática que se vive diariamente en la Universidad, especialmente en las diferentes áreas de aparcamiento, ésta solución trata de un prototipo de sistema para dispositivos móviles, en la cual se presenta información en tiempo real sobre la disponibilidad de un espacio para aparcar un vehículo,

este espacio es el más cercano al lugar de trabajo o estudios del usuario que solicite la información y para mayor comodidad del usuario, la aplicación grafica la ruta que debe seguir el usuario para llegar al sitio de aparcamiento asignado, el sector que se utilizó para el desarrollo del aplicativo es el área de parqueo ubicado en la parte posterior del Departamento de Ciencias de la Computación (DCCO).

Para el desarrollo del sistema informático se incorporó el lenguaje de marcado **KML** que se basa en lenguaje **XML**, este lenguaje ayuda en la representación de los datos geográficos, tales como: ubicación exacta, distancia desde un punto de referencia, latitud, longitud, altura de cada lugar que tiene el área de aparcamiento ubicada en la parte posterior del DCCO.

### **1.6. Hipótesis**

El desarrollo del sistema de información enfocado al ambiente móvil facilitará la detección de espacios disponibles de aparcamiento en los parqueaderos de la Universidad de las Fuerzas Armadas ESPE.

### **1.7. Metodología de la investigación**

Para el presente trabajo de titulación se propone emplear una metodología Ad-Hoc la cual tiene consta de cinco fases, cada una de ellas con un entregable al finalizar cada fase.

Las etapas consideradas en la metodología se muestran en la figura 1, cada una de las fases tiene una serie de actividades y entregables al finalizar la fase:

### 1. Fase I. Análisis de la Situación Actual.

En esta fase se realizará una revisión de literatura en bases digitales sobre sistemas de información en tiempo real de plazas disponibles de aparcamiento en una zona de parqueo de vehículos.

### 2. Fase II. Evaluar las herramientas Hardware y Software para la solución planteada.

En la siguiente fase, el grupo de trabajo evaluará herramientas de Hardware y Software disponibles para la construcción del prototipo del sistema de información de plazas disponibles de aparcamiento, el objetivo de esta fase es seleccionar las mejores herramientas disponibles en el mercado tecnológico para construir una aplicación eficiente para los usuarios.

### 3. Fase III. Diseño del prototipo.

En la fase del diseño del prototipo netamente se desarrollará la aplicación móvil que ejecutará el sistema de información en tiempo real de plazas disponibles de aparcamiento, se utilizará las herramientas de hardware y software que fueron seleccionadas en la fase anterior, el objetivo de esta fase es presentar un posible prototipo funcional, mismo que permita ejecutar pruebas de validación en la comunidad universitaria ESPE.

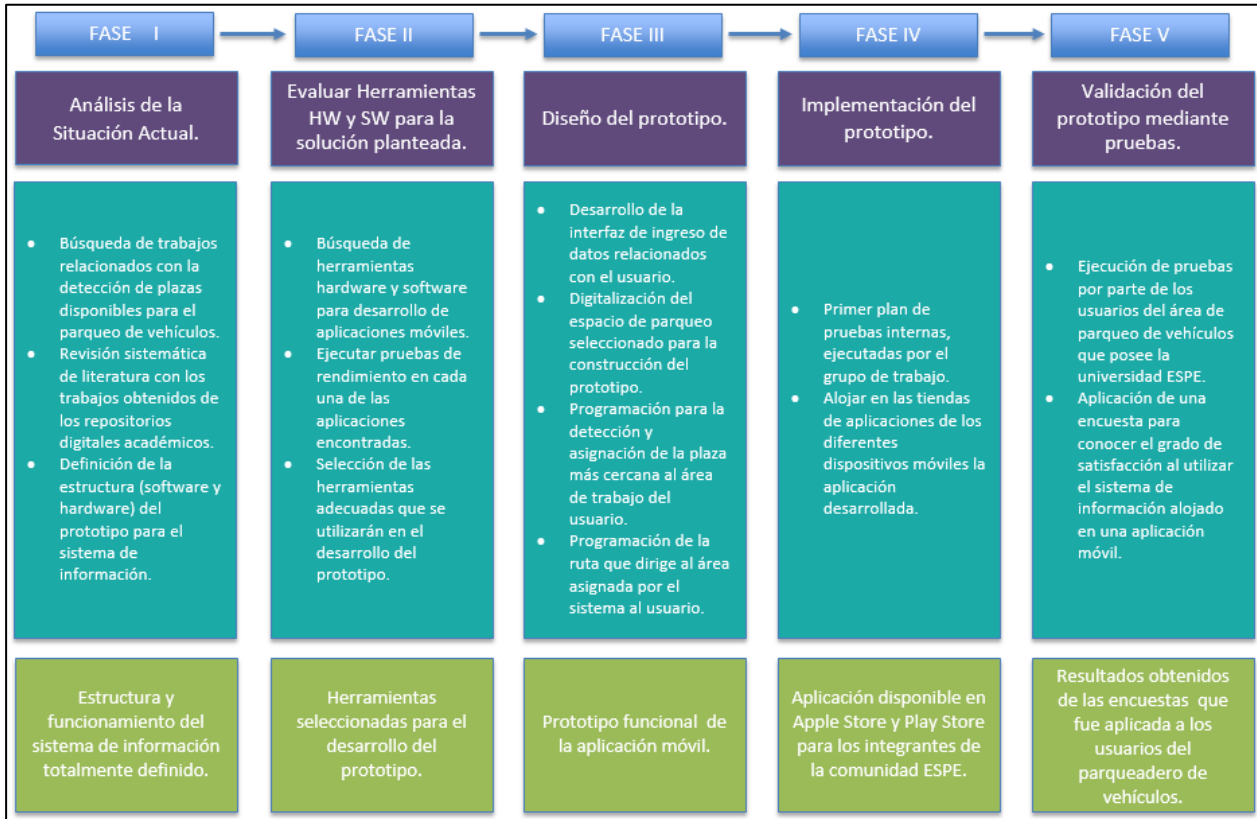
### 4. Fase IV. Implementación del prototipo.

Una vez que el diseño del prototipo este validado con un correcto y óptimo funcionamiento, el equipo de trabajo va a colocar la aplicación móvil en un ambiente de producción, es decir, alojará en las diferentes tiendas de aplicaciones disponibles para

cada Sistema Operativo móvil que actualmente existe, dando apertura a los usuarios para la validación del prototipo.

#### 5. Fase V. Validación del prototipo mediante pruebas.

Esta es la fase final, en la cual se validará el prototipo de información de plazas disponibles de aparcamiento para vehículos, esta validación se ejecutará tomando una pequeña muestra de estudiantes, docentes o integrantes de la comunidad ESPE que dispongan de un vehículo y sean usuarios de los parqueaderos disponibles en la ESPE. Una vez que hagan uso del aplicativo se ejecutará una encuesta en la cual se mida el grado de satisfacción después de haber sido localizada una plaza de aparcamiento disponible para su vehículo y dirigido hacia ella rápidamente.



**Figura 1.** Metodología Ad Hoc



## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1. Introducción**

El propósito de este capítulo es presentar los fundamentos teóricos sobre el que se sustenta el proyecto denominado sistema de información para informar, ubicar y dirigir hacia plazas disponible de aparcamiento vehicular en la Universidad de las Fuerzas Armadas ESPE, la tecnología empleada, la metodología aplicada en el desarrollo del proyecto.

#### **2.2. Dispositivos Móviles**

##### **2.2.1. Definición**

“Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento y conexión permanente o intermitente a una red, con memoria ilimitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales. De acuerdo con esta definición existen multitud de dispositivos móviles, desde los reproductores de audio portátiles hasta los navegadores GPS, pasando por los teléfonos móviles, los PDAs o los Tablet PCs” (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello EPSIG, 2011).

##### **2.2.2. Categorías de dispositivos móviles**

En el año 2005, el grupo T38 y DuPont Mobility Innovation Team definieron varios estándares que se debían cumplir para la clasificación de los diferentes dispositivos móviles que están disponibles en el mercado, esto se decidió hacer debido a la variedad

de funcionalidades que vienen incorporados en los dispositivos y los servicios que brinda cada uno de ellos. Los estándares para la clasificación son los siguientes:

- Dispositivos Móviles de datos limitados: en esta categoría están los teléfonos móviles clásicos. Su principal característica es tener una pantalla pequeña. Ofrecen servicios de datos generalmente limitados a SMS y a acceso WAP.
- Dispositivos Móviles de datos básicos: la principal característica de esta categoría es que tienen integrado una pantalla de mediano tamaño, menú o navegación basada en iconos, además ofrece acceso a emails, lista de direcciones, SMS y en la mayoría de los casos tienen incorporados un navegador web básico. Un Smartphone es el ejemplo de teléfono inteligente que representa esta categoría.
- Dispositivos Móviles de Datos Mejorados: esta categoría agrupa a los dispositivos que tienen pantallas de medianas y grandes dimensiones, esto comprende pantallas desde los 240 x 120 pixeles en adelante. Otra de las grandes ventajas que ofrece esta categoría es la incorporación de aplicaciones nativas como las que desarrolló Microsoft Office Mobile (Word, Excel, Power Point, OneDrive, etc) y aplicaciones de ámbito corporativo como SAP, Portales Intranet, etc.

### **2.2.3. Tipos de dispositivos móviles**

Existen diferentes tipos de dispositivos móviles, en la actualidad los más utilizados son los siguientes:

#### **2.2.3.1. Smartphone**

“El Smartphone es un teléfono inteligente que tiene más funcionalidades que un teléfono celular común. Posee características similares a las de un sistema operativo, por

lo cual son vulnerables a los virus o a los ataques al mismo sistema operativo tal como sucede como un computador normal” (Aponte Gomez Carlos Davila Ramirez Universidad Ean Facultad Ingenieria De Sistemas, 2011).

Las características más relevantes de un Smartphone son las siguientes:

- Tiene integrado la localización por medio de GPS.
- Soporta correo electrónico.
- Permite el ingreso de datos mediante una interfaz gráfica, ya sea por medio del teclado o por medio de la pantalla táctil.

#### **2.2.3.2. Feature Phones**

“Un Feature Phone, es cualquier teléfono móvil que no sea un Smartphone y que se destaca por alguna de sus características principales, como cámara de fotos de alta resolución y ajustes para mejorar las capturas, reproductor MP3 con controles dedicados para esta operación, teclado físico para mensajería y pantalla sensible al tacto entre otros” (Tomoyose, 2011).

Actualmente se está trabajando con el fin de tener una mejora continua, ya que las nuevas versiones de Java o Brew permiten que el software sea más poderoso y que se pueda integrar con algunas funciones del teléfono, aunque existe una gran diferencia en el lado de la interfaz gráfica de usuario, estos dispositivos están limitados a solo una sección especial de aplicaciones de la interfaz (Oracle, 2018).

Este tipo de dispositivo tecnológico tiene incorporado su propio sistema operativo y su propio firmware. Es posible que soporten una aplicación que sea desarrollado por

terceros, sin embargo, esto es posible mediante una plataforma limitada como Java o Brew que frente a plataformas móviles son mucho menos poderosas en el aspecto de que están estructuradas con menos funcionalidades que las de un teléfono móvil.

### **2.2.3.3. Tablets**

En este tipo de dispositivos se hace referencia a “dispositivos digitales con capacidades de procesamiento de información y navegación en Internet similares o ligeramente inferiores a la de un computador portátil del tipo notebook. Sus principales características son batería de larga duración (8 horas), pantalla táctil, bajo peso (500 gramos) y tamaño (hasta 10 pulgadas) lo que mejora la portabilidad. Poseen sistemas operativos específicos, más asimilables a las plataformas empleadas por los teléfonos inteligentes o smartphones (Blackberry, Iphone, Android). Las aplicaciones que le dan funcionalidad están íntimamente asociadas al perfil de usuario, y en la mayoría de los casos, permiten acceder a conectividad a través de wifi y 3G” (Marés, 2012).

## **2.3. Sistemas operativos para dispositivos móviles**

### **2.3.1. Definición**

Partiendo de la definición de sistema operativo: se puede definir como “el medio de comunicación entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo” (Baz Alonso et al., 2011).

### 2.3.2. Tipos de sistemas operativos

En la actualidad existen diferentes tipos de dispositivos móviles, entre los más reconocidos tenemos a los siguientes:

- **Android:** Sistema operativo de Google, actualmente es el más utilizado por varias industrias que se dedican a la fabricación de celulares.
- **iOS:** Sistema operativo más seguro del mercado, fue creado por la empresa Apple y es utilizado por iPhone e iPad.
- **Windows Phone:** Sistema operativo propio de la empresa Microsoft, actualmente solo está implementado en teléfonos de gama alta.

En la figura 2 se presenta los iconos de los tres sistemas operativos que actualmente lideran el mercado tecnológico móvil y están disponibles para dispositivos.



**Figura 2.** Sistemas operativos móviles – Iconos.

Fuente: (GCFGlobal, 2019)

## **2.4. Sistema Operativo - Android**

### **2.4.1. Antecedentes**

Android es una plataforma bastante popular de los teléfonos inteligentes debido a que es de código abierto. Este sistema operativo está basado en Linux, inicialmente fue desarrollado por Android, Inc., pocos años más tarde, en el año 2005, sería comprado por la compañía Google (López Michelone, 2013).

Las aplicaciones que existen para este sistema operativo móvil son creadas por una comunidad de desarrolladores que su única finalidad es extender la funcionalidad de estos dispositivos, estas aplicaciones son alojadas en la tienda virtual de Google que se denomina “Google Play” y actualmente ya son alrededor de un millón de aplicaciones que están disponibles para Android, de este millón de aplicaciones aproximadamente los dos tercios son gratuitos y el resto son pagadas.

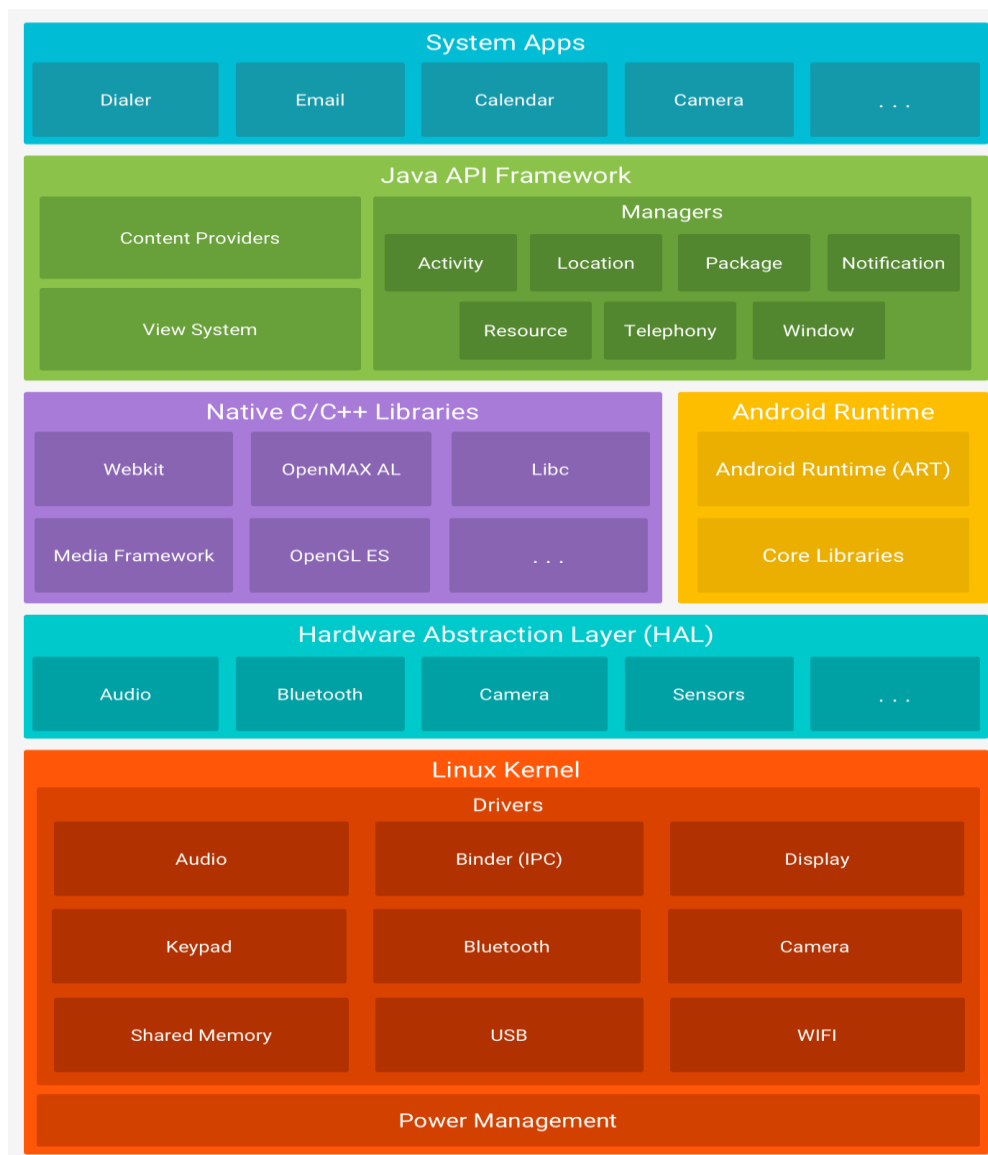
### **2.4.2. Definición**

“Android es un sistema operativo que inicialmente fue diseñado para dispositivos móviles, la base de este sistema es que está construido sobre Linux que es un núcleo de sistema operativo libre, gratuito y multiplataforma lo que ha permitido integrar dicho sistema operativo en muchos otros dispositivos en formato de licencias de software libre” (Gonzalez, 2011).

### **2.4.3. Componentes**

El sistema operativo Android está formado por varios componentes, los cuáles se agrupan en cuatro capas principales, cada elemento se relaciona con los elementos de las capas adyacentes para facilitar las funcionalidades y el trabajo que cada uno debe

cumplir, la arquitectura interna de Android proporciona todas las interfaces necesarias para el desarrollo de aplicaciones que puedan acceder a las funcionalidades del teléfono (GPS, llamadas, agenda, WiFi, Bluetooth, etc). En la figura 3, se observa el diagrama de arquitectura del sistema operativo Android.



**Figura 3.** Arquitectura interna de Android.

Fuente: (Developers, 2019a)

Sobre el Kernel se sitúan las bibliotecas de Android que están escritas en C o C++ y compiladas para la arquitectura específica de cada teléfono, su propósito es proporcionar funcionalidad a las aplicaciones, para tareas que se repiten con frecuencia garantizando que se ejecuten de la forma más eficiente. Algunas de las bibliotecas que se incluyen habitualmente son (Vico, 2011):

- **Gestor de superficies (Surface Manager):** compone y presenta las imágenes que se muestran en la pantalla en dimensiones 2D y 3D.
- **SGL (Scalable Graphics Library):** se encarga de la representación de los elementos en dos dimensiones, motor gráfico 2D de Android.
- **OpenGL (OpenGL for Embedded Systems):** es el motor gráfico 3D de Android, está basado en las APIs de OpenGL ES 1.0 y 1.1, por lo general utiliza un motor software altamente optimizado.
- **Bibliotecas multimedia:** está biblioteca permite la visualización, reproducción y grabación de los formatos de audio, video e imagen (JPG, GIF, PNG, MP3, AMR, etc)
- **WebKit:** es el motor web que utiliza el navegador que tiene incorporado el teléfono.
- **SSL:** está basado en criptografía y su funcionalidad es proveer de seguridad al momento de acceder a internet.
- **SQLite:** medio de almacenamiento de datos, es el motor de base de datos relacional, actualmente se encuentra disponible para todas las aplicaciones.



- **Biblioteca C de sistemas (libc):** aquí se basa las aplicaciones, es el motor en donde arrancan y se ejecutan las aplicaciones del dispositivo móvil. Se basa en la implementación de Berkeley Software Distribution (BSD).

#### 2.4.4. Características

Dentro de las principales características de Android tenemos a las más importantes que se resumen en la tabla 1:

**Tabla 1**  
*Características de Android*

Aplicación	Características
<b>Diseño de dispositivos</b>	La plataforma es adaptable a pantallas más grandes, VGA, biblioteca de gráficos 2D y 3D basadas en las especificaciones de OpenGL ES 2.0 y el diseño de teléfonos tradicionales.
<b>Almacenamiento</b>	SQLite, una base de datos liviana, es usada para propósitos de almacenamiento de datos.
<b>Conectividad</b>	Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EVDO, UMTS, Bluetooth, Wi-Fi, LTE y WiMax.
<b>Mensajería</b>	SMS y MMS son formas de mensajería, incluye mensajería de texto y actualmente Android Cloud to Device Messaging Framework (C2DM).
<b>Navegador Web</b>	El navegador web está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome.
<b>Soporte de Java</b>	Corre en la máquina virtual Dalvik que es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitado.

<b>Soporte Multimedia</b>	Android soporta los siguientes formatos multimedia: WebM, H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, MP3, MIDI, WAV, JPEG, PNG, GIF y BMP.
<b>Soporte para streaming</b>	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML5. Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player.
<b>Soporte para hardware adicional</b>	Android soporta cámaras de fotos, vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, termómetro, aceleración 2D y 3D.
<b>Entorno de desarrollo</b>	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado es Eclipse en su versión actual 3.6.
<b>Market</b>	El Android Market es un catálogo de aplicaciones gratuitas o de pago mediante el cual es posible descargar e instalar en dispositivos Android sin la necesidad de un PC.
<b>Multi-táctil</b>	Android tiene soporte nativo para pantallas multi-táctiles que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de Kernel. Google publicó una actualización para el NexusOne y el Motorola Droid que activa el soporte para pantallas multi-táctiles de forma nativa.
<b>Bluetooth</b>	El soporte para A2DF y AVRCP fue agregado en la versión 1.5, el envío de archivos (OPP) fue agregado en la versión 2.0, el marcador por voz junto con el envío de contactos entre teléfonos fue agregado en la versión 2.2.
<b>Video llamada</b>	Android soporta videollamada a través de Google Talk desde su versión HoneyComb.

<b>Multitarea</b>	Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.
<b>Características basadas en voz</b>	La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.
<b>Tethering</b>	Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2).

Fuente: (Juliana, Rivera, Sandoval, Santiago, & Franco, 2012)

### 2.4.5.Evolución

Android surge en los años 2008 lanzando por primera vez al mercado su versión oficial, desde el lanzamiento de su primera versión hasta la actualidad ha tenido varias versiones que han aportado significativamente en el desarrollo para llegar a convertirse en uno de los más grandes exponentes en el ámbito de los dispositivos móviles.

En la tabla 2 se presenta las características que ha incorporado el sistema operativo Android en cada versión lanzada al mercado.

**Tabla 2**  
*Versiones de Android y sus características*

No.	Nombre	Fecha Publicación	Kernel	Cámara	USB	Actualizaciones	Rendimiento	GPS	Bluetooth	Wi-Fi	Multicore
1.0	Angel Cake	Sep. 2008		X					X	X	
1.1	Battenberg	Feb. 2009		X					X	X	
1.5	Cupcake	Abr. 2009	2.6.27	X					X	X	
1.6	Donut	Sep. 2009	2.6.29	X			Opt.		X	X	
2.0/2.1	Éclair	Oct. 2009	2.6.29	X			Opt.		X	X	
2.2	Froyo	May. 2010	2.6.32	X	X		Opt.		X	X	X
2.3	Gingerbread	Dic. 2010	2.6.33	X	X				X	X	X

CONTINÚA 

<b>3.0</b>	Honeycomb	Feb. 2011	2.6.36	X	X		Opt.		X	X	X
<b>4.0</b>	Ice Cream Sandwich	Oct. 2011	3.0.1	X	X		Opt.	X	X	X	X
<b>4.1</b>	Jelly Bean	Jun. 2012	3.0.3	X	X	X	Opt.	X	X	X	X
<b>4.4</b>	KitKat	Oct. 2013	3.4.5	X	X	X	Opt.	X	X	X	X
<b>5.0/5.1</b>	Lollipop	Nov. 2014	3.14	X	X	Des.	Opt.	X	X	X	X

Fuente: (Ibarra, Esteban, Maldonado, & Mariel, 2015)

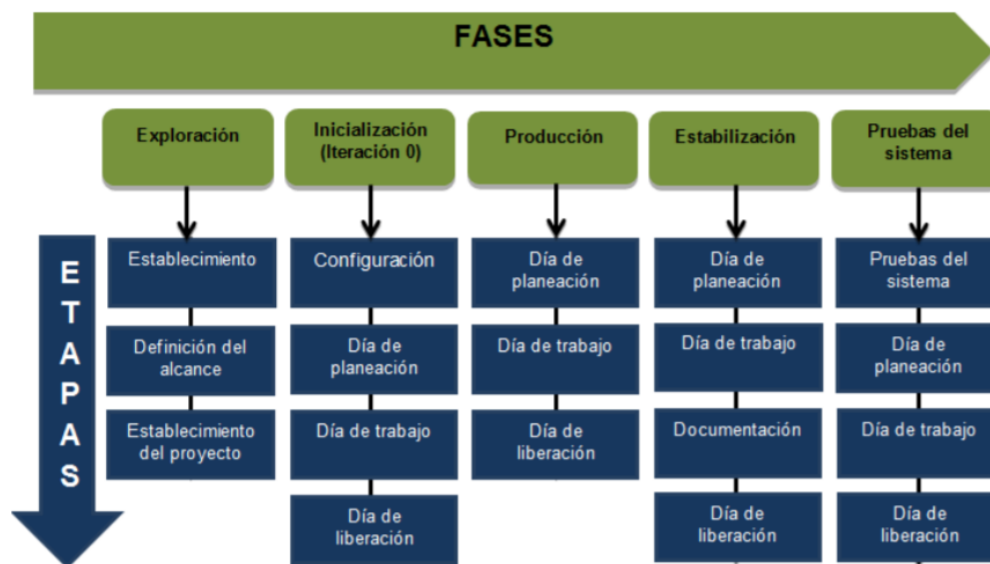
## 2.5. Metodologías de desarrollo móvil - Mobile D

### 2.5.1. Definición

“Abrahamsson muestra que el desarrollo ágil proporciona un buen ajuste para el entorno de desarrollo de aplicaciones móviles y propone un enfoque llamado Mobile D. Se basa en XP, metodologías Crystal y Rational Unified Process (RUP) y se recomienda que sea utilizado por equipos pequeños y ubicados conjuntamente. Trabajando en ciclos cortos de desarrollo. Mobile-D consta de cinco fases (explorar, inicializar, producir, estabilizar y prueba y arreglo del sistema), cada uno de que tiene varias etapas, tareas y prácticas asociadas.” (P, Hamsini, & Smitha, 2016)

### 2.5.2. Ciclo de vida de Mobile-D

Esta metodología está orientada a la necesidad de obtener disponible un ciclo de desarrollo muy rápido para equipos conformados por menos de diez personas. En la figura 4 se aprecia las etapas que comprende la metodología Mobile-D.



**Figura 4.** Ciclo de desarrollo Mobile-D

Fuente: (Bonilla Guillermo Santiago & Guzmán Alex Israel, n.d.)

A continuación, se describe cada una de las fases que comprende la metodología (Tumipampa Borja, 2016):

- **Exploración:** la fase de exploración se centra en la planificación y en los conceptos básicos del proyecto, en esta fase se define el alcance del proyecto y las funcionalidades que se quiere lograr.
- **Inicialización:** de esta fase depende el éxito de las siguientes etapas del proyecto ya que se debe preparar y verificar todos los aspectos críticos que se presentan en el desarrollo del proyecto, y el objetivo es corregir los inconvenientes en la fase de aplicación de los requisitos en el menor tiempo posible.
- **Producción:** aquí netamente se implementa la funcionalidad del producto, por lo general se suele emplear un ciclo de desarrollo iterativo e incremental.

- **Estabilización:** el objetivo de la fase de estabilización es asegurar la calidad de la ejecución del proyecto,
- **Pruebas:** es la fase final del producto, el objetivo es llegar a una versión estable del producto según los requisitos del cliente en las primeras fases, si existe algún tipo de error se debe corregir más no desarrollar nada nuevo.

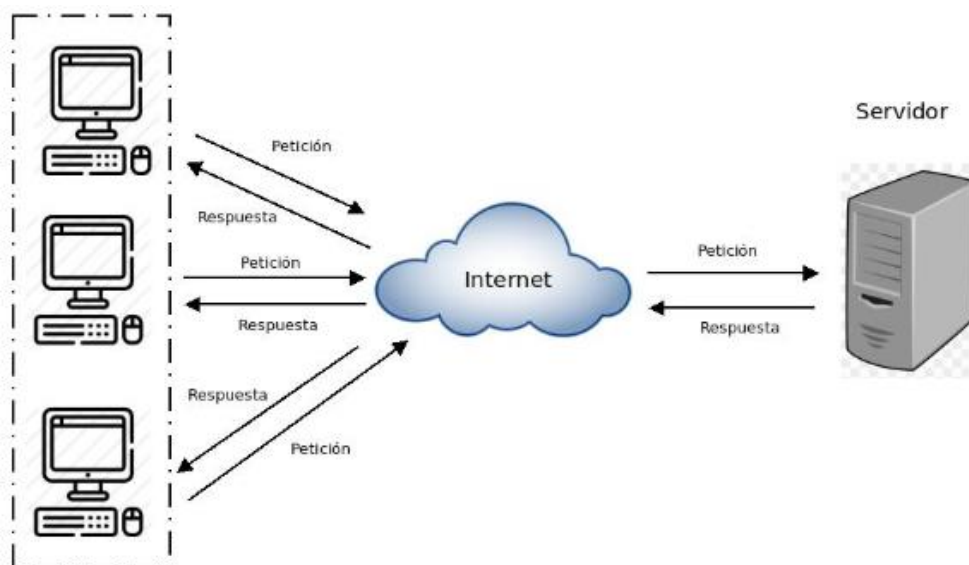
## **2.6. Patrones de arquitectura**

En el diseño de la arquitectura de software los elementos y relaciones están definidos a un alto nivel mediante los requisitos funcionales y no funcionales, estos requisitos definen la arquitectura ya que si se desea implementar un sitio web en el cual se solicite autenticación para acceder, su arquitectura estará definida por los requisitos de seguridad, confidencialidad e interfaz que a su vez definen las tecnologías a emplear (Calero, 2010) (Escalante, 2016).

A continuación, se cita a los patrones de arquitectura de software más comunes, haciendo referencia a sus principales características.

### **2.6.1. Modelo cliente - servidor**

Este modelo se caracteriza por tener dos partes: un cliente y un servidor. El modelo básicamente consiste en hacer peticiones desde el cliente hacia el servidor y recibir una respuesta de este. Las peticiones y respuestas son información en forma de texto HTML, imágenes u otros elementos (Escalante, 2016).



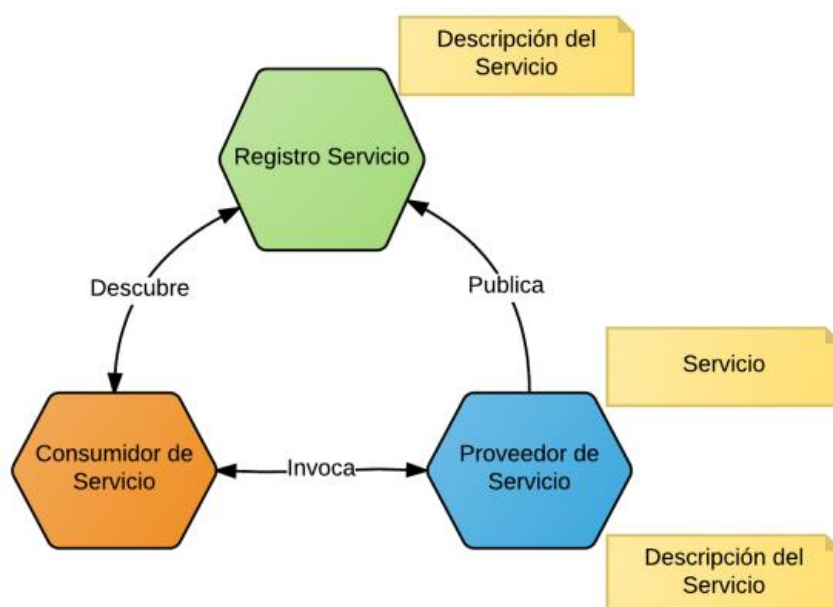
**Figura 5.** Diagrama Modelo Cliente-Servidor

Fuente: (Schiaffarino, 2019)

### 2.6.2.Arquitectura orientada a servicios SOA

La arquitectura orientada a servicios (Service Oriented Architecture) es un patrón que establece restricciones a tener en cuenta por el desarrollador de software: respuestas rápidas, adaptativas y posibles cambios en los requisitos. Este patrón orquesta el funcionamiento del software a partir de un servidor central, quien es el que reparte los servicios entre diferentes servidores, que a su vez lo llevan hacia sus respectivos clientes (Duarte, 2016).

En la figura 6 se representa de manera gráfica el paradigma descubrir, publicar e invocar que se utiliza en la arquitectura SOA.



**Figura 6.** Diagrama de arquitectura orientada a servicios

Fuente: (Duarte, 2016)

## 2.7. Geolocalización

### 2.7.1. Definición

“La geolocalización es la capacidad de asignar coordenadas geográficas a la información por medio de herramientas informáticas. La generalización de la tecnología GPS en dispositivos de uso personal como los teléfonos móviles y ordenadores personales han permitido que esta capacidad esté al alcance de cualquier ciudadano, y como consecuencia, el desarrollo de aplicaciones a distintos campos” (Rodríguez, 2010).

En el ámbito tecnológico, la geolocalización es la posibilidad de conocer la posición geográfica en la que está ubicado un objeto, ya sea Tablet, teléfono celular, ordenador, laptop o cualquier dispositivo que tenga incorporado la funcionalidad de ubicación. En la actualidad, la gran mayoría de estos dispositivos posee un sistema que proporciona esta



información para la localización y detección de las coordenadas en las que se encuentra un objeto.

### **2.7.2. Tipos de geolocalización**

Entre los tipos de geolocalización más importantes y los que más se utilizan a diario en las actividades cotidianas, citamos los siguientes:

#### **2.7.2.1. GPS**

GPS significa Global Positioning System, es decir, “Sistema de Posicionamiento Global. Se desarrolló a partir de los satélites de la constelación NAVSTAR (NAVigation Satellite Timing And Ranging), fue llevado a cabo por el departamento de defensa de los Estados Unidos, que lanzó el primer satélite el 22 de febrero de 1978. El sistema GPS comprende tres segmentos diferentes” (Vázquez, 2010):

- El segmento espacial que son los satélites que giran en órbitas alrededor de la Tierra.
- El segmento de control: formado por estaciones ubicadas cerca del ecuador terrestre para controlar a los satélites.
- El segmento de usuarios: este segmento va dirigido para cualquier persona que reciba y utilice las señales GPS.

#### **2.7.2.2. GSM**

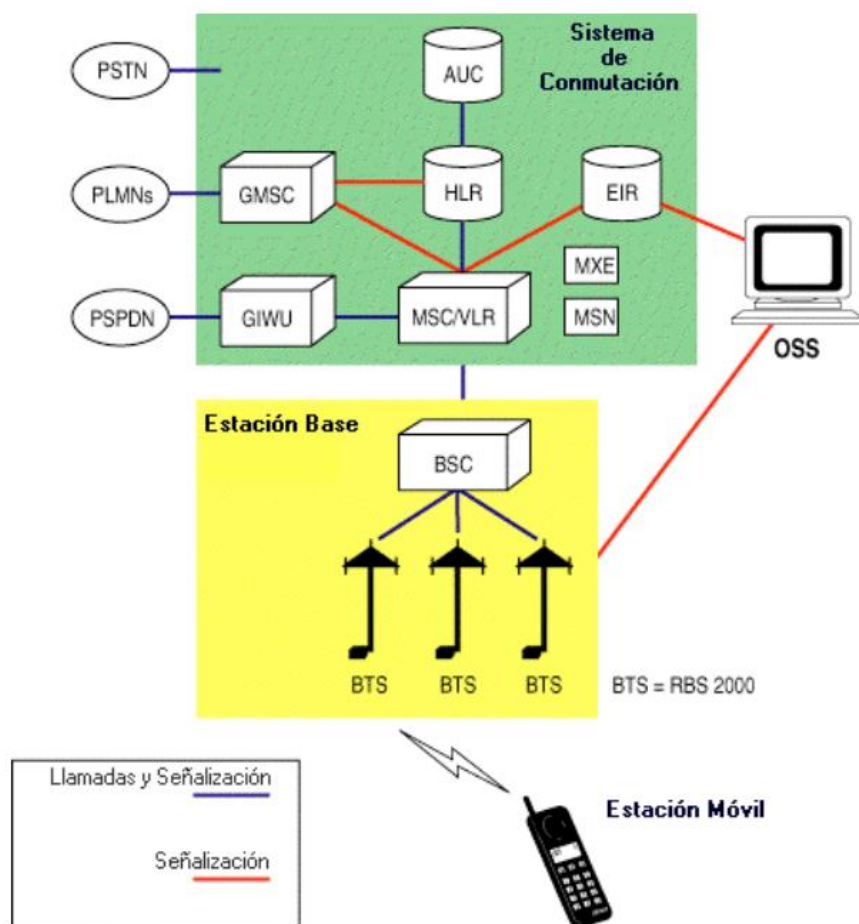
Con el paso del tiempo y la evolución acelerada que ha sufrido la tecnología, cada país desarrolló su propia tecnología, esto se debe a que la evolución de las comunicaciones celulares obligó a crear un medio para enviar y recibir información desde

diferentes partes del mundo, la creación de estos medios de comunicación provocó la aparición de incompatibilidades que aportaron de manera negativa al desarrollo ya que limitaron el uso de los equipos tecnológicos.

La CEPT (Conference of European Post and Telecommunications) formó la organización GSM (Group Special Mobile), que tiempo más tarde pasaría a conocerse como Global System for Mobile communications) cuyo principal objetivo sería desarrollar un sistema celular estandarizado, este sistema debería cumplir los siguientes requisitos (Lizón, 2013):

- Eficiencia espectral.
- Roaming internacional.
- Costes económicos de los móviles y las estaciones base.
- Calidad de voz.
- Compatibilidad con otros servicios como los ISDN (Integrated Services Digital Network).
- Capacidad de soportar nuevos servicios.

En la figura 7 se puede apreciar la estructura de una red GSM, la cual se divide en tres sistemas principales: Sistema de Conmutación (SS, Switching System), Sistema de Estación Base (BSS, Base Station System) y Sistema de Operación y Soporte (OSS, Operation and Support System).



**Figura 7.** Estructura de la red GSM

Fuente: (Lizón, 2013)

### 2.7.3. Archivos KMZ

Los archivos KML son archivos propios de Google Earth, con información geográfica de un lugar, ruta, polígono o conjunto de ellos. Se trata de archivos que ocupan muy poco espacio de disco duro y fácilmente distribuibles. Los archivos KMZ son una versión comprimida de los KML.

Basta con tener instalada la aplicación y ejecutar el archivo para que la aplicación se abra y nos lleve automáticamente al lugar deseado, pudiendo realizar todas las

operaciones sobre el “elemento” del que se trate en el ordenador de destino (Santos & Dpto, 2015).

El archivo KMZ tiene en su estructura un archivo KML que es el principal, adicional tiene más archivos complementarios empaquetados en formato .zip formando una unidad denominada fichero. El archivo KMZ tiene la posibilidad de almacenar y ser enviado por correo electrónico como una única entidad. Cuando el archivo KMZ se descomprime, el archivo .kml principal y los archivos complementarios se separan según sus formatos y estructura de directorios originales, con sus nombres de archivo y extensiones. El archivo KML de 10 KB se puede presentar como un archivo KMZ comprimido, teniendo un peso muy reducido equivalente al de 1 KB.

## **2.8. Lenguaje de marcado KML**

### **2.8.1. Antecedentes**

KML es la abreviatura de “Keyhole Markup Language” que originalmente fue creado por Keyhole Inc., En el año 2008 KML fue reconocido como un estándar en el “Open Geospatial Consortium” ya que inicialmente fue pensado para ser usado en el software Keyhole LT, este software fue modificado y utilizado para el uso de archivos KML conocido en la actualidad como Google Earth.

### **2.8.2. Definición**

“Keyhole Markup Language” o KML es un archivo que contiene coordenadas, direcciones, altura, entre otras variables que permiten representar en un mapa una ruta o punto de interés, en otras palabras, es el lenguaje de marcado basado en XML para representar datos geográficos en tres dimensiones (Du, Yu, & Liu, 2009).

### 2.8.3.Estructura

Un archivo KML especifica una característica (un lugar, una imagen, una posición, una ruta o un polígono) que contiene coordenadas (latitud, longitud y altura), que en muchas ocasiones puede venir comprimido en un archivo KMZ (Rohlf & McCledon, 2008).

Un archivo KML está formado de la siguiente estructura (Ibarra et al., 2015):

- **Cabecera del XML**

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

- **Namespace propio de KML**

```
<kml xmlns="http://www.opengis.net/kml/2.2"
xmlns:gx="http://www.google.com/kml/ext/2.2"
xmlns:kml="http://www.opengis.net/kml/2.2"
xmlns:atom="http://www.w3.org/2005/Atom">
```

- **El objeto Placemark que hace referencia a la posición, dividido en nombre, descripción y conjunto de coordenadas que conforman la ruta.**

```
<Placemark>
```

```
    <name>Ruta de Prueba</name>
```

```
    <styleUrl>#m_ylw-pushpin</styleUrl>
```

```
    <LineString>
```

```
        <tessellate>1</tessellate>
```

```
        <coordinates>
```

```
            -78.45876922394318,-0.2907473038582219,0-78.45962845232083
```

```
        </coordinates>
```

```
</LineString>
```

```
</Placemark>
```

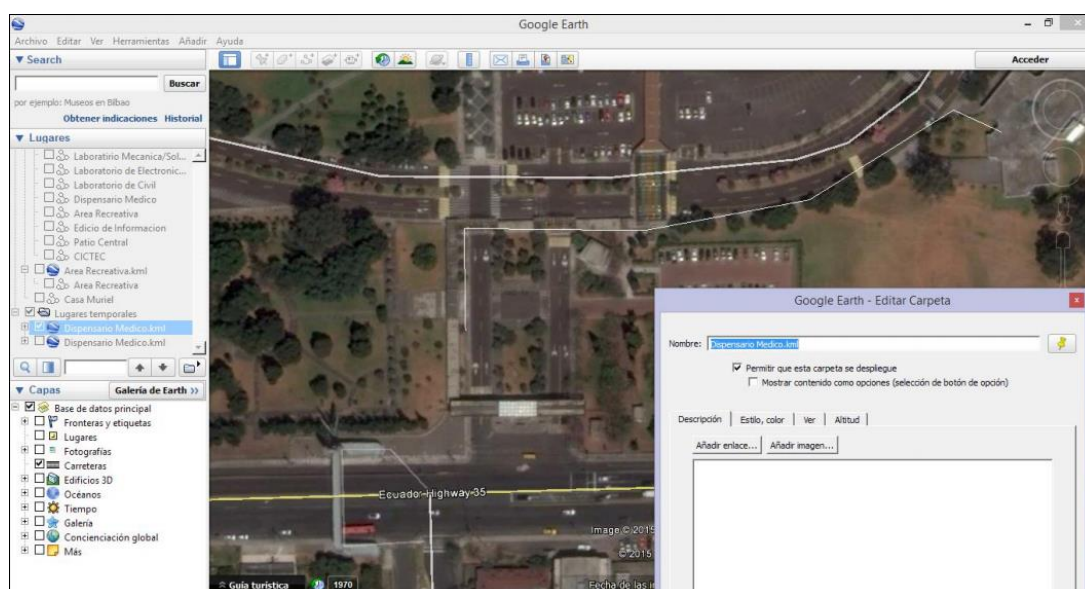
En los dispositivos móviles, un archivo KML permite los siguientes elementos para brindar un servicio más útil y óptimo al momento de utilizar el servicio de geolocalización:

- Iconos.
- Carpetas.
- Elementos HTML.
- Marcas de posición con elementos de nombre <name>.
- Cadenas de líneas y polígonos.

#### **2.8.4.Funcionalidad**

La principal funcionalidad de los archivos KML es crear y almacenar coordenadas geográficas tales como: puntos, rutas, imágenes, polígonos los cuales se pueden utilizar para referencia un lugar específico que otros usuarios deseen conocer.

Existen varias formas de crear un archivo KML, la más básica y que se utiliza con frecuencia es desde un archivo de texto plano, este editor debe ser un programa que sea especializado en coordenadas y que sea capaz de generar el archivo KML como en la figura 8.



**Figura 8.** Archivo KML creado desde Google Earth

Fuente: (Ibarra et al., 2015)

## 2.9. Base de datos No SQL

### 2.9.1. Definición

Las bases de datos NoSQL son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación. Mientras que las bases de datos relacionales basan su funcionamiento en tablas, joins y transacciones. Las bases de datos NoSQL no imponen una estructura de datos en forma de tablas y relaciones entre ellas, sino que proveen un esquema mucho más flexible (Martín, Chávez, Rodríguez, & Valenzuela, 2013).

Las bases NoSQL tienen como característica principal una escalabilidad realmente enorme, se basa en la utilización de modelos de consistencia los cuales no garantizan una consistencia de los datos que se almacenan, pero el principal objetivo es lograr un mayor performance y disponibilidad. Uno de los inconvenientes más grande es que tienen

disponible un lenguaje de consulta que tenga la característica de ser declarativo, esto obliga a que sea requerida una mayor programación para la manipulación de los datos.

En la tabla 3 se presenta la comparación de las diferentes bases de datos NoSQL, sus características y la aplicabilidad que tiene cada una de ellas.

**Tabla 3**

*Comparación de los diferentes tipos de base de datos NoSQL*

Base de Datos	Características	Aplicabilidad
<b>Mongo DB</b>	<ul style="list-style-type: none"> <li>• Escrito en C++. Algunas características SQL (Query,index).</li> <li>• Protocolo binario.</li> <li>• Replicación maestro-esclavo.</li> <li>• Permite ejecutar Javascript.</li> </ul>	<p>Querys dinámicos, no para Map/Reduce.</p> <p>Cuando necesitas CouchDB con muchos cambios</p> <p>Para muchos trabajos que ahora se realiza con MySQL</p>
<b>Riak</b>	<ul style="list-style-type: none"> <li>• Escrito en Erlang y CTolerancia a fallos.</li> <li>• Protocolo binario o HTTP/REST.</li> <li>• Map/Reduce en JS o Erlang.</li> <li>• Versión Opensource y Enterprise.</li> </ul>	<p>Estilo Cassandra pero sin su complejidad.</p> <p>Escalabilidad, disponibilidad y tolerancia a fallos.</p>
<b>Couch DB</b>	<ul style="list-style-type: none"> <li>• Escrito en Earlang Protocolo HTTP/Rest.</li> <li>• Incluye librería JQuery.</li> <li>• Replicación bidireccional con detección de conflictos.</li> </ul>	<p>Permite acumular datos que solo cambian ocasionalmente con queries predefinidos.</p> <p>Sistemas con replicación.</p>
<b>Redis</b>	<ul style="list-style-type: none"> <li>• Escrito en C++.</li> <li>• Base de datos en memoria con backup en disco.</li> </ul>	<p>Datos que requieren ser modificados constantemente en memoria.</p> <p>Base de datos en tiempo real.</p>

**CONTINÚA** 



	<ul style="list-style-type: none"> <li>• Replicación maestro – esclavo.</li> <li>• Clave – valor pero los valores pueden ser List, Sets, Hashes.</li> </ul>	Comunicación en tiempo real.
<b>HBase</b>	<ul style="list-style-type: none"> <li>• Escrito en Java, billones de filas por millones de columnas.</li> <li>• Protocolo HTTP/REST y Thrift.</li> <li>• Map/Reduce con Hadoop.</li> <li>• Optimizaciones para Querys en tiempo real.</li> <li>• Rendimiento random-access como MySQL.</li> </ul>	<p>Mejor opción para Map/Reduce.</p> <p>Almacenamiento y análisis ficheros de log.</p>
<b>Neo4J</b>	<ul style="list-style-type: none"> <li>• Escrito en Java, base de datos de grafos.</li> <li>• Protocolo HTTP/REST o Java.</li> <li>• Lenguaje de queries pattern-maching.</li> <li>• Interfaz web para la administración.</li> <li>• Backup online, monitorización y alta disponibilidad en versión comercial AGPL.</li> </ul>	<p>Para datos ricos interconectados estilo grafos.</p> <p>Empleado en redes sociales, topologías de red.</p>
<b>Cassandra</b>	<ul style="list-style-type: none"> <li>• Escrito en Java, lo mejor de BigTable y Dinamo.</li> <li>• Búsqueda por columnas o rango de claves.</li> <li>• Escrituras más rápidas que lecturas.</li> <li>• Map/Reduce con Hadoop.</li> </ul>	<p>Más escrituras que lectura (logging), cuando todos los componentes son Java.</p> <p>Análisis en tiempo real.</p>

<b>Membase</b>	<ul style="list-style-type: none"> <li>• Escrito en Erlang y CCompatible Memcached pero con persistencia y clustering.</li> </ul>	Aplicaciones con acceso de muy baja latencia.
	<ul style="list-style-type: none"> <li>• Acceso muy rápido por clave.</li> </ul>	Aplicaciones con alta concurrencia y alta disponibilidad.
	<ul style="list-style-type: none"> <li>• Persistencia en disco.</li> </ul>	Online gaming.
	<ul style="list-style-type: none"> <li>• GUI para gestión del clúster.</li> </ul>	
	<ul style="list-style-type: none"> <li>• Actualizaciones de software sin parar la base de datos.</li> </ul>	

Fuente: (Martín et al., 2013)

## 2.9.2. Bases NoSQL en comparación con SQL

En la tabla 4 muestra algunas de las diferencias que existen entre las bases de datos NoSQL y las bases de datos SQL.

**Tabla 4**

*Base de datos NoSQL en comparación con base de datos SQL.*

	<b>Bases de datos SQL</b>	<b>Bases de datos NoSQL</b>
<b>Cargas de trabajo óptimas</b>	Las bases de datos relacionales están diseñadas para aplicaciones de procesamiento de transacciones online (OLTP) altamente coherentes y transaccionales, y son buenas para el procesamiento analítico online (OLAP).	Las bases de datos clave-valor, documentos, gráficos y en memoria de NoSQL están diseñadas para OLTP para una serie de patrones de acceso a datos que incluyen aplicaciones de baja latencia. Las bases de datos de búsqueda NoSQL están diseñadas para hacer análisis sobre datos semiestructurados.

**CONTINÚA** 

**Modelo de datos**

El modelo relacional normaliza los datos en tablas conformadas por filas y columnas. Un esquema define estrictamente las tablas, las filas, las columnas, los índices, las relaciones entre las tablas y otros elementos de las bases de datos. La base de datos impone la integridad referencial en las relaciones entre tablas.

Las bases de datos NoSQL proporcionan una variedad de modelos de datos, que incluyen documentos, gráficos, clave-valor, en memoria y búsqueda.

**Propiedades ACID**

Las bases de datos relacionales ofrecen propiedades de atomicidad, coherencia, aislamiento y durabilidad (ACID):

- La atomicidad requiere que una transacción se ejecute por completo o no se ejecute en absoluto.
- La coherencia requiere que, una vez confirmada una transacción, los datos deban acoplarse al esquema de la base de datos.
- El aislamiento requiere que las transacciones simultáneas se ejecuten por separado.

Las bases de datos NoSQL a menudo hacen concesiones al flexibilizar algunas de las propiedades ACID de las bases de datos relacionales para un modelo de datos más flexible que puede escalar horizontalmente. Esto hace que las bases de datos NoSQL sean una excelente opción para casos de uso de baja latencia y alto rendimiento que necesitan escalar horizontalmente más allá de las limitaciones de una sola instancia.

**CONTINÚA** 

- La durabilidad requiere la capacidad de recuperarse de un error inesperado del sistema o de un corte de energía y volver al último estado conocido.

### Rendimiento

Normalmente, el rendimiento depende del subsistema de disco. Se necesita la optimización de consultas, índices y estructura de tabla para lograr el máximo rendimiento.

El rendimiento es, por lo general, depende del tamaño del clúster de hardware subyacente, la latencia de red y la aplicación que efectúa la llamada.

### Escalado

Las bases de datos relacionales generalmente escalan en forma ascendente las capacidades de computación del hardware o la ampliación mediante la adición de réplicas para cargas de trabajo de solo lectura.

Las bases de datos NoSQL normalmente se pueden particionar porque los patrones de acceso de valores clave son escalables mediante el uso de arquitectura distribuida para aumentar el rendimiento que proporciona un rendimiento constante a una escala casi ilimitada.

### API

Solicita almacenar y recuperar datos que están comunicados mediante consultas que se ajustan a un lenguaje de consulta estructurado (SQL). Estas consultas son analizadas y

Las API basadas en objetos permiten a los desarrolladores almacenar y recuperar fácilmente estructuras de datos en memoria. Las claves de partición permiten que las aplicaciones busquen pares de clave-

---

ejecutadas por la base de datos relacional. valor, conjuntos de columnas o documentos semiestructurados que contengan atributos y objetos de aplicación serializados.

---

Fuente: (AWS, 2019)

### 2.9.3. Clasificación

A los sistemas de base de datos NoSQL se puede clasificar en cuatro grandes categorías, las cuáles se describe a continuación:

- Framework Map-Reduce, este tipo de bases de datos tienden a ser utilizados por aplicaciones que ejecutan procesamientos analíticos online, este tipo de procesamiento se denomina OLTP. Un ejemplo de base de datos de este tipo es Hadoop.
- Almacenamiento Clave – Valor, en el grupo de las bases de datos que manejan un almacenamiento clave - valor podemos encontrar a Amazon, Dynamo, Cassandra, HBase, las cuales tienen como principal objetivo el procesar la información y las transacciones de manera online.
- Sistemas de bases de datos gráficas, el propósito de una base de datos gráfica es facilitar la creación y la ejecución de aplicaciones que funcionan con conjuntos de datos altamente conectados. Los casos de uso típicos para una base de datos gráficas incluyen redes sociales, motores de recomendaciones, detección de fraude y gráficos de conocimiento (AWS, 2019).
- Almacenamiento de Documentos, aquí se maneja el principio de que los datos se representan como un objeto o como un documento de tipo JSON, esto se debe

a que es un modelo eficiente para los desarrolladores, ya que les facilita el almacenamiento y la consulta de datos en una base mediante el formato de modelo de documento que fue empleado para la búsqueda mediante el código de aplicación. Entre las bases más comunes y utilizadas tenemos a CouchDB, Simple DB y MongoDB.

## **2.9.4.MongoDB**

### **2.9.4.1. Antecedentes**

MongoDB es una base de datos NoSQL que está orientada a documentos, se deriva de la palabra inglesa “humongous” que significa enorme, extraordinariamente largo. La primera versión de esta base de datos fue publicada en marzo del 2009, esta publicación fue en código abierto bajo licencia AGPL. Cinco meses más tarde en agosto del 2009, fue lanzada la versión estable 1.4 que fue considerada como apta para el ambiente de producción y distribución. La última versión 1.4, fue publicada el 29 de noviembre del 2016. Actualmente las únicas versiones que se mantienen actualizadas son desde la 2.0 en adelante.

### **2.9.4.2. Definición**

“MongoDB es una base de datos no relacional, que comparando con una base de datos relacional tiene colecciones como tablas y los documentos son archivos. La diferencia es que en una base de datos relacional cada archivo en una tabla tiene la misma cantidad de campos, mientras que una base de datos no relacional cada documento en una colección puede tener diferentes campos. Trabaja bajo un sistema de indexación que es similar a los de las bases de datos relacionales. Soporta el tipo de

replicación maestro - esclavo, en donde el maestro puede ejecutar comandos de lectura y escritura, mientras tanto que el esclavo puede copiar los datos del maestro y sólo se puede emplear para lectura y copias de seguridad” (Bazo, 2019).

MongoDB obligatoriamente se ejecuta en múltiples servidores a la vez, con el fin de equilibrar el duplicado de los datos para poder mantener el sistema funcionando en óptimas condiciones, ejecuta un procesamiento de datos por lotes mediante la función MapReduce. Actualmente tiene disponible, drivers oficiales para los siguientes lenguajes de programación: C, C++, C#, Erlang, Haskell, Java, Java Script, Lisp, node.JS, Perl, PHP, Python, Ruby, Scala.

#### 2.9.4.3. Características

MongoDB tiene varias características que facilitan la operación y el almacenamiento masivo de datos, las características son las siguientes (Javier Martin, 2019):

- **Almacenamiento orientado a documentos:** los documentos para el almacenamiento de información tienen un estilo JSON, con un esquema dinámico ofrecen simplicidad y poder.
- **Auto-Sharding:** escalabilidad horizontal completamente independiente de la operatividad, actualmente está limitada a 200 nodos.
- **Consultas:** las consultas se basan en documentos, son bastantes rápidas para la ejecución de queries de lectura y escritura básica.
- **Mapeo y reducción:** permite el procesamiento de datos y una agregación flexible de información hacia la base de datos.

- **Soporta consultas dinámicas:** permite ejecutar consultas sobre cualquier valor de los documentos en los cuales están almacenados, y no solamente en los indexados.

#### **2.9.4.4. Funcionamiento**

“MongoDB viene de serie con una consola desde la que podemos ejecutar los distintos comandos. Esta consola está construida sobre JavaScript, por lo que las consultas se realizan utilizando ese lenguaje. Además de las funciones de MongoDB, podemos utilizar muchas de las funciones propias de JavaScript. En la consola también podemos definir variables, funciones o utilizar bucles” (Ruben, 2014).

Con relación al almacenamiento de datos, utiliza documentos por medio de una serie binaria de JSON que se denomina BSON, está formado por una lista ordenada de elementos los cuales están limitados a un tamaño de 4 MB. La gran ventaja de este tipo de base de datos es la de almacenar documentos sin tener un esquema definido y aquí se puede guardar documentos con varios campos dentro de la misma colección.

### **2.10. Framework de desarrollo móvil**

#### **2.10.1. Definición**

“Los frameworks o marcos de trabajo, son esquemas para el desarrollo y/o implementación de una aplicación. Definen, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. Los frameworks incluyen programas de apoyo, compiladores, bibliotecas de código, interfaz de programación de aplicaciones (API) y un conjunto de herramientas que reúnen a todos



los diferentes componentes para permitir el desarrollo de un proyecto o solución.” (Ghatol & Patel, 2011)

El framework también se puede definir como un esquema básico y fundamental que sirve para el desarrollo de software, en el cual están automatizados uno o más procesos que son base para que el código escrito por el desarrollador se pueda ejecutar sin problema alguno.

## **2.10.2. Tipos**

### **2.10.2.1. IONIC**

IONIC Framework está formado por un conjunto de herramientas de interfaz de usuario la cuál es de código abierto, la funcionalidad esencial de este framework es utilizar tecnologías web como HTML, CSS y JavaScript, con el propósito de crear, diseñar aplicaciones móviles y aplicaciones de escritorio de muy buena calidad, y de alto rendimiento. Actualmente este framework está integrado con Angular y React, lo que permite mejorar la experiencia con el usuario ya que presenta una interfaz compuesta por controles, interacciones y animaciones que es bastante simple.

IONIC presenta varias metas para brindar un óptimo funcionamiento al momento de crear y ejecutar código, entras las metas más importantes se puede destacar a las siguientes:

- **Multiplataforma:** es multiplataforma porque permite crear y desarrollar aplicaciones que se puedan ejecutar en cualquier plataforma móvil, ya sea iOS o Android, o a su vez permite crear una aplicación web progresiva. Todo

esto es posible escribiendo el código una sola vez y se puede ejecutar en diferentes plataformas o ambientes.

- **Basado en estándares web:** la base en la que se apoya Ionic es fundamentada en tecnologías web estandarizadas como lo es HTML, CSS y JavaScript, en donde además incorporan API web modernas estables, las cuales no dependen de un único proveedor de plataformas.
- **Elegante diseño:** tiene incorporado varios componentes como tipografía, paradigmas interactivos que son prediseñados y están listo para ser utilizados en cualquier plataforma.
- **Sencillez:** Ionic está creado con el propósito de que sea fácil de aprender, agradable para los usuarios y lo más importante que esté disponible para cualquier desarrollador que tengas habilidades de programación.

“Ionic Framework es un proyecto gratuito y de código abierto, lanzado bajo la licencia MIT permitida. Esto significa que puede usar en proyectos personales o comerciales de forma gratuita. MIT es la misma licencia utilizada por proyectos tan populares como jQuery y Ruby on Rails.” (Ionic, 2019)

#### **2.10.2.2. Android Studio**

“Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android, basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ” (Developers, 2019b).

El IDE en sus nuevas versiones incorpora funcionalidades que potencian el desarrollo de aplicaciones para Android, estas funcionalidades son:

- Sistema de compilación muy flexible que se basa en Gradle.
- Emulador mucho más rápido en cuanto a la ejecución de las aplicaciones.
- Entorno de desarrollo unificado el cual permite el desarrollo de aplicaciones para todos los dispositivos Android.
- Compatibilidad con C++ y NDK.

Como ya se había mencionado, Gradle es el sistema de compilación que utiliza Android Studio como la base de compilación de su sistema, las funciones disponibles en el sistema de compilación son: personalizar el proceso de compilación, crear diferentes APK de la aplicación móvil esto dependiendo de la densidad de la pantalla hdpi, mdpi y el código se vuelva completamente reutilizable.

Una de las funcionalidades más importantes que tiene implementada la plataforma Android Studio es la reducción de recursos, ayuda a eliminar de forma automática los recursos que no son necesarios para el uso del paquete de la aplicación y las dependencias de las bibliotecas utilizadas en el funcionamiento.

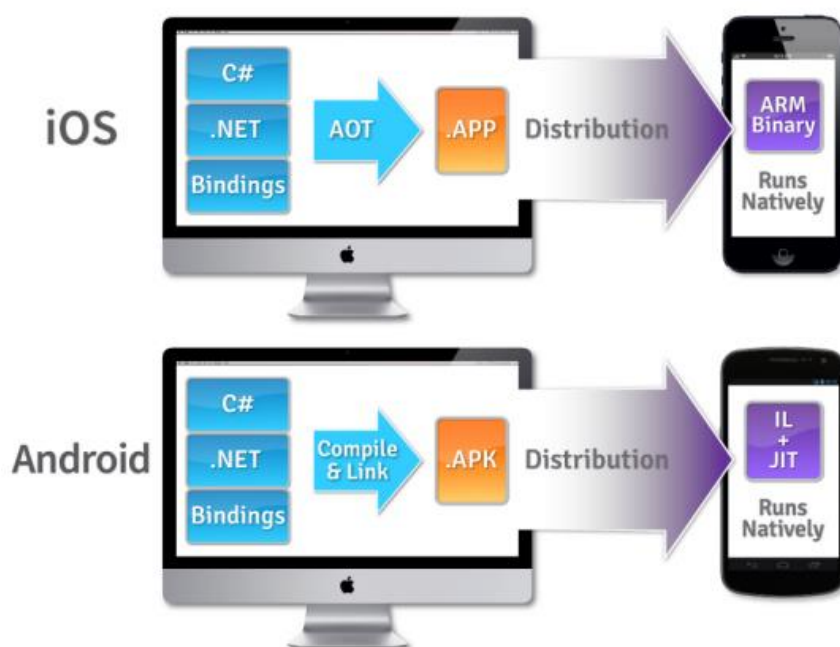
### **2.10.2.3. Xamarin**

“Xamarin es una solución de desarrollo de aplicaciones móviles multiplataforma que crea un entorno unificado para desarrolladores. En lugar de escribir código en ObjectiveC para iOS y Java para Android, permite a los desarrolladores escribir código en el lenguaje C#” (Dickson, 2013).

El desarrollo de una aplicación para cada plataforma móvil es independiente por que las interfaces de usuario son nativas, esto se lo realiza a través de un proyecto

específico que preserva los componentes nativos de cada plataforma. Cada aplicación en proceso de desarrollo tiene acceso directo a las API nativas, lo que permite poder trabajar con las funciones propias del hardware que está integrado en el dispositivo.

En la figura 9 se observa el mecanismo que emplea Xamarin para compilar de manera correcta las aplicaciones en cada plataforma, tanto iOS como Android.



**Figura 9.** Actuación de Xamarin en iOS y Android.

Fuente: (Dickson, 2013)

Xamarin compila el código C# para iOS en un binario ARM nativo que permite a las aplicaciones iOS estar al mismo nivel de sus homólogos que se basan en ObjectiveC y este proceso implica utilizar el motor LLVM que es usado por Xcode, el entorno de desarrollo integrado para crear aplicaciones híbridas. Mientras que la compilación de Android genera un código de lenguaje interpretado (IL) que se compila y ejecuta directamente en el kernel de Linux y no lo hace sobre una máquina virtual.

## 2.11. Google Maps

### 2.11.1. Definición

“La geolocalización es una de las herramientas utilizadas por los geógrafos para situar a las personas u objetos en el espacio mediante sus coordenadas y que ha cobrado una nueva dimensión a partir de la aparición de Internet y de los dispositivos móviles” (Beltrán López, 2015).

La geolocalización está compuesta por los siguientes elementos:

- **Dispositivo hardware:** este dispositivo actúa como una plataforma en la que se desarrollará el flujo que debe seguir el proceso de geolocalización, este dispositivo puede ser un ordenador, una laptop, un dispositivo móvil, etc.
- **Programa software:** será el encargado de iniciar y ejecutar el proceso de geolocalización de acuerdo con su implementación y función.
- **Conexión a internet:** la principal función de este elemento es obtener e intercambiar información entre los dispositivos tecnológicos que interactúen en el sistema, y en ocasiones actuará como sistema de almacenamiento y procesamiento de esta información.

### 2.11.2. Tipos de Geolocalización

- **Social:** se basa en la unión de dos elementos, los dispositivos móviles y la social media mediante los cuales se genera un canal de comunicación para comunicar y compartir el lugar exacto en el que se está visitando en cada momento.

- **Online:** es posible obtener información en tiempo real, por lo que se define como “el conjunto de aplicaciones que permiten ubicar una entidad en el espacio físico (localizar) con unos atributos (información) obtenidos a través de internet y que se visualizan sobre un mapa, mediante una técnica que se denomina georreferencia” (López, 2016).

### 2.11.3. Ventajas y Desventajas

En el proceso de Geolocalización participan una serie de satélites los cuales desempeñan un papel fundamental informando en que parte de la tierra está ubicado un elemento, las ventajas y desventajas de la geolocalización se presentan en la tabla 5.

**Tabla 5**  
*Ventajas y desventajas de la Geolocalización*

Ventajas	Desventajas
Localización de la ubicación exacta de todo tipo de objetos y lugares.	Exposición de nuestra ubicación actual al usar algunos apps o dispositivos que afecta a nuestra privacidad.
Planificación de nuevas rutas antes de un viaje y de gran interés comercial.	Conocimiento de nuestros gustos y preferencias de lugares cuando nos desplazamos, rutas habituales, etc.
Importante información sobre el tráfico, clima y otros a tiempo real.	
Se puede obtener la información que se requiere de manera inmediata.	

CONTINÚA 

Compartir experiencias y ubicaciones de lugares visitados con amigos y familiares a través de fotografías o mensajerías instantáneas.

Fuente: (Fajardo, Mercado, Morales, Sánchez, & Santillán, 2018)

#### 2.11.4. Generación de datos de la Geolocalización

La información presentada en la geolocalización se genera de dos formas en base a datos que se recopilan en un modo activo que se basa en el dispositivo tecnológico del usuario o a su vez, un modo pasivo que se conoce como búsqueda, en la tabla 6 se resume estos dos modos:

**Tabla 6**

*Modos de generación y recolección de datos de Geolocalización*

Modo	Modo de recolección	Tecnologías Involucradas
Activo: Basado en el dispositivo del usuario.	<ul style="list-style-type: none"> <li>Utiliza el firmware y el software de la computadora o dispositivo inalámbrico del usuario.</li> <li>La ubicación se determina a través del chip del GPS y triangulación mediante la información de las torres celulares.</li> <li>Modelo de petición-respuesta.</li> </ul>	<ul style="list-style-type: none"> <li>GPS.</li> <li>GPS asistido (A-GPS).</li> <li>Wi-Fi posicionamiento inalámbrico.</li> <li>3G/4G.</li> <li>Aplicaciones móviles: iPhone, dispositivos Android, BlackBerry.</li> </ul>
Pasivos: Búsqueda de datos basada en servidores.	<ul style="list-style-type: none"> <li>Involucra el uso de proveedores de servicios de</li> </ul>	<ul style="list-style-type: none"> <li>Ubicación IP: búsqueda Whois, DNS, LOC, nombres</li> </ul>

CONTINÚA 

- 
- |   |  |
|---|--|
| geolocalización de terceros, por ej. Quova, NatGeo, Bering Media. | geográficos en el usuario del nombre del dominio o información, aplicaciones, datos de horarios mediante la interferencia de ping basada en los datos de enrutamiento, por ej. Monitorización de Traceroute de las redes del proveedor del servicio de internet (ISP). |
|---|--|
- Basado en la dirección IP que no depende de una ubicación específica adquirida del dispositivo del usuario o de los identificadores del conjunto de servicios (SSID) para redes inalámbricas.
  - La correlación con las bases de datos de IP o SSID obtenidas de los registros de compras, la información suministrada por los usuarios, los análisis de red de las rutas de rastreo y los nombres almacenados del sistema de nombres de dominio (DNS).
- 3G/4G.
  - Wi-Fi – Posicionamiento inalámbrico.



## 2.12.Servicios REST (Web Services)

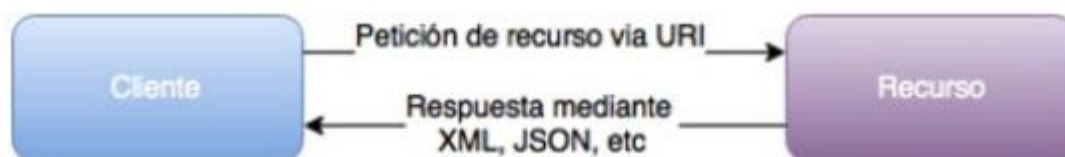
### 2.12.1. Definición

Representational State Transfer es una arquitectura de software que se encarga de representar la transferencia de datos. Dicha arquitectura se apoya en el protocolo HTTP para la transferencia de información entre ordenadores.

La principal funcionalidad de REST es tratar a los objetos como recursos que pueden ser creados y destruidos, teniendo para ello cuatro métodos básicos: put, get, delete y post, los cuales se encargan de crear, leer, eliminar y actualizar respectivamente (Melnichuk, Kornienko, & Boytsova, 2018).

REST se usa para la construcción de servicios web, los cuales, basados en REST reciben el nombre de “servicios RESTful” (Richardson & Ruby, 2007).

En la figura 10 se muestra un ejemplo en donde se puede observar la interacción de los componentes entre aplicaciones y servicios web.



**Figura 10.** Estructura básica de la arquitectura REST.

Fuente: (Richardson & Ruby, 2007)

### 2.12.2.Métodos HTTP

Los métodos HTTP definen las acciones a realizar sobre los recursos. Los más usados son POST, GET, PUT, DELETE que corresponden a las acciones Create, Read,

Update, Delete (CRUD) (Fredrich, 2012), a continuación, se describen cada una de estas acciones:

- GET, este método es usado para leer el contenido de un recurso, no es posible modificarlo o dañarlo. La característica esencial de este método es que para una respuesta exitosa emite el código 200 para OK y el código 404 o 400 para informar que el proceso tuvo error. La representación de la información es en formato XML o JSON.
- PUT, cuando se requiere realizar una actualización a un recurso se debe invocar a este método, el cuerpo del mensaje requiere contener los datos que actualizarán el recurso original. El cuerpo en la respuesta del PUT es opcional. PUT es idempotente, pero no es seguro porque modifica el estado del recurso (Rodríguez, 2016).
- POST, es el método usado para la creación de un nuevo recurso, al momento de la creación se le asigna una ID que lo asocia con el padre, al finalizar la creación retorna el código 201 para informar que el proceso finalizó con éxito.
- DELETE, este método es usado para eliminar un recurso identificado por su URI, el borrado exitoso retorna el código 200 con el cuerpo de la respuesta, o el código 204 si no se incluye contenido en el cuerpo de la respuesta (Rodríguez, 2016).

### **2.12.3. Características**

- Protocolo cliente/servidor sin estado, las peticiones HTTP contienen toda la información necesaria para ejecutarla, esto permite que tanto el cliente como el servidor no necesiten recordar ningún estado previo para satisfacerla.

- Las operaciones más importantes de los servicios REST que están relacionadas con los datos en cualquier sistema REST son las siguientes: POST, GET, PUT, DELETE.
- En la tecnología REST, los objetos siempre se manipulan a partir de la URI, misma que nos facilita acceder a la información para su modificación o eliminación, y también nos sirve para compartir su ubicación exacta con terceros.
- Para la transferencia de datos el sistema REST posee una interfaz uniforme que aplica acciones concretas como Post, Get, Put y Delete, sobre los recursos del servicio, esto lo ejecuta siempre y cuando estén identificados con una URI única.
- Los servicios web presentan una arquitectura jerárquica en su estructura de componentes, cada una de estas capas ejecuta una funcionalidad dentro del sistema REST.

#### **2.12.4.Ventajas**

- El protocolo REST permite la separación entre el cliente y el servidor, es decir, separa totalmente la interfaz de usuario del servidor del almacenamiento de datos permitiendo mejorar la portabilidad de la interfaz a distintas plataformas, aumenta la escalabilidad de los proyectos y permite que los componentes de cada desarrollo se puedan evolucionar de forma independiente.
- Convierte a las aplicaciones en productos más flexibles a la hora de trabajar ya que facilita la migración a otros servidores siempre y cuando los datos de cada una de las peticiones se envíen de forma correcta.

- Una de las grandes ventajas que brinda la tecnología REST es que es posible tener servidores corriendo en PHP, Java, Python, Node JS o servidores ejecutándose en cualquier otro sistema operativo diferente al que estamos trabajando es factible trabajar sin problema alguno, incluso probar nuevos entornos de desarrollo.
- Es totalmente independiente de la plataforma en la que está montado el servidor, razón por la cual se puede hacer uso de REST en Windows, Linux, Mac sin problema alguno.

## **2.13. Metodología para crear una encuesta**

### **2.13.1. Definición**

La encuesta es un método de la investigación que sirve para obtener y recolectar información específica de una muestra de la población mediante el uso de cuestionarios estructurados que se utilizan para obtener datos precisos de las personas encuestadas (Thompson, 2006).

Otra de las definiciones acertadas que tiene una encuesta es que “consiste en reunir datos entrevistando a las personas. Las encuestas se pueden hacer en persona, por teléfono, por correo o por Internet. La ventaja de la encuesta es que la información viene directamente de la persona que a usted le interesa. De hecho, puede ser la única forma de determinar las opiniones o planes de compra de un grupo” (Stanton, Etzel, & Walker, 2007).

### 2.13.2. Características

Este método tiene algunas características que lo hace ser bastante útil al momento de recolectar datos y analizar datos, ya que las respuestas obtenidas vienen directas de los entrevistados, algunas de estas características son:

- **Variedad y flexibilidad de aplicaciones**, es una de las características que la hace útil a este método de recolección de datos, ya que funciona en una variedad de campos y aplicaciones, por ejemplo, la comunicación, la salud, la política, la utilización de los sistemas, en las Universidades y los centros de investigación académica.
- **Comparabilidad**, los resultados que presenta una encuesta, permite comparar de manera directa y objetiva entre diferentes áreas, aplicaciones o ambiente en los cuales se esté aplicando dicha encuesta.
- **Oportunidad**, debido al desarrollo acelerado de las diferentes tecnologías que existen en la actualidad, los resultados de los estudios realizados mediante este método ahora se pueden presentar con una gran velocidad.
- **Comprensión**, en la actualidad este método de recolección de datos ha sido bastante difundido, razón por la cual ha permitido que cada uno de los usuarios que hacen uso la puedan comprender con mayor facilidad el funcionamiento del trabajo y los procesos que implica seguir para obtener los resultados esperados.
- **Precisión**, esta técnica tiene la naturaleza de permitir conocer el nivel de precisión de los resultados obtenidos, lo cual también permite conocer el

riesgo implícito que trae las decisiones que se tomarán en base a los resultados.

### 2.13.3. Tipos de encuestas

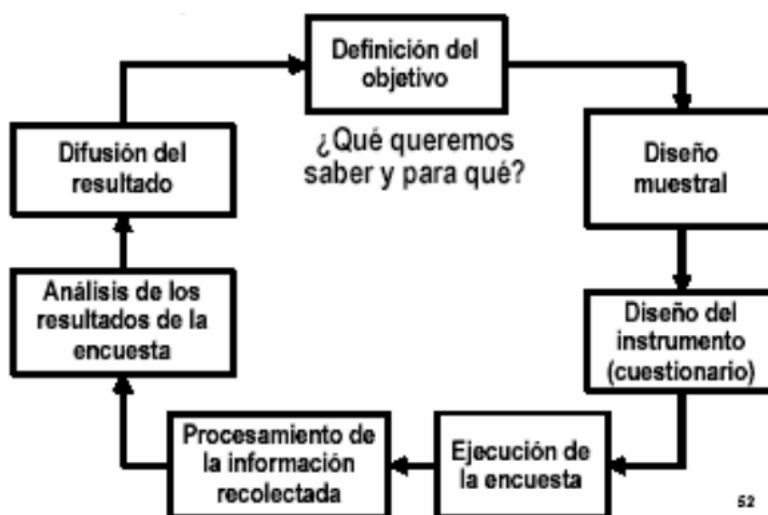
Actualmente existe cuatro tipos de encuestas que se utilizan según el medio en el cual se las requiera emplear, a continuación, se detallan los tipos de encuesta (Thompson, 2006):

- **Encuestas basadas en entrevistas cara a cara, o también denominadas, encuestas a profundidad**, consiste en realizar entrevistas directas o personales con cada uno de los encuestados, una de las grandes ventajas es que son guiadas y controladas por el encuestador y brinda mucha más información que los otros tipos de encuestas.
- **Encuestas telefónicas**, en este tipo de encuestas el principal dispositivo de comunicación es el teléfono, se realiza una entrevista vía telefónica con cada uno de los encuestados. Una gran ventaja que tiene realizar este tipo de encuesta es el abarcar un mayor número de personas en menos tiempo del que se emplea en la entrevista personal.
- **Encuestas postales**, es necesario tener preparado un cuestionario para que este sea remitido a los encuestados, los cuales tendrán que llenar y una vez completado lo reenvíen al encuestador. La principal ventaja de este tipo de encuesta es la sinceridad con la que suelen responder los encuestados, ya que no tiene la presión directa de un encuestador.

- **Encuestas por internet**, el objetivo de este tipo de encuestas es colocar una serie de preguntas, tipo cuestionario, en una página web para que cada encuestado acceda a responder las preguntas, uno de los beneficios más importantes que brinda es la amplia cobertura a la que se puede llegar pudiendo obtener miles de encuestas respondidas en cuestión de horas o minutos.

#### 2.13.4.Ciclo de una encuesta

La figura 11 muestra el ciclo de ejecución de una encuesta, en la cual se tiene en cuenta los siete pasos principales del proceso que implica la implementación de la encuesta:



**Figura 11.** Ciclo de una encuesta.

Fuente: (Diplomado de Gobierno Abierto y Participativo Institucional Mayo-Agosto 2005 Módulo 2

Metodología para llevar a cabo una encuesta, n.d.)

A continuación, se describe detalladamente cada una de las fases que comprende el ciclo de ejecución de una encuesta, el objetivo es facilitar el entendimiento de lo que se debe realizar en cada una las actividades del ciclo de la encuesta:

- **Definición del objetivo:** en esta fase se debe identificar a donde es que se quiere llegar con la encuesta, implica responder a las preguntas: ¿Qué se desea investigar y cuál es el propósito?, ¿Qué y Para qué?, es la fase más importante del ciclo ya que de esta fase depende los siguientes pasos a seguir.
- **Diseño muestral:** el desarrollo de esta fase consiste en la elaboración del diseño muestral que depende del universo que fue seleccionado para la aplicación de la encuesta. La muestra se define como la representación de una población que es objeto de estudio, es definida mediante el tamaño de la población.
- **Diseño del instrumento:** en esta fase se desarrolla el cuestionario que se va a utilizar para la encuesta, esto implica responder a la pregunta ¿Cómo se captará la información?, y aquí se define el tipo de pregunta que se van a utilizar, si son cerradas o abiertas.
- **Ejecución de la encuesta:** la persona encargada de ejecutar la encuesta debe tener un alto grado de credibilidad, habilidad para realizar las diferentes entrevistas.
- **Procesamiento de la información recolectada:** se basará en función a la metodología definida previamente, es importante que este procesamiento



se lo ejecute con un enfoque objetivo e imparcial con el fin de evitar errores de manipulación de información.

- **Análisis de los resultados de las encuestas:** en el análisis de los resultados que arroja la encuesta no debe intervenir la organización que solicitó ejecutar la encuesta, es importante garantizar transparencia, consistencia técnica durante todo el proceso.
- **Difusión del resultado:** la publicación de los resultados debe ser clara, amplia, precisa y de fácil acceso, ya que esto ayuda a generar credibilidad hacia los resultados.

## **CAPÍTULO III**

### **DESARROLLO**

#### **3.1. Introducción**

En el presente capítulo se describen las fases de la metodología empleada para el desarrollo del prototipo del sistema de información. El proyecto se desarrolló en IONIC versión 5.3.2, bajo la plataforma Windows como sistema operativo, se utilizó la herramienta Google Maps para la implementación de la ruta que dirige hacia el sitio disponible de aparcamiento. La metodología empleada es una metodología Ad Hoc propuesta por el grupo de trabajo, ya que permite implementar y presentar un prototipo construido en fases. Las fases desarrolladas fueron: Análisis de la situación actual, Evaluación de las herramientas Hardware y Software para la solución planteada, Diseño del prototipo, Implementación del prototipo, Validación del prototipo mediante pruebas.

Cada fase de la metodología Ad-Hoc propuesta por el grupo de trabajo se desarrolla a continuación:

#### **3.2. Fase I: Análisis de la situación actual**

En la Fase I se realizó una revisión de literatura en bases digitales sobre sistemas de información en tiempo real de plazas disponibles de aparcamiento en una zona de parqueo de vehículos.

### **3.2.1. Búsqueda de trabajos relacionados con la detección de plazas disponibles para el parqueo de vehículos**

Para la búsqueda de los trabajos relacionados con la detección de plazas disponible para el aparcamiento de vehículos, se creó una cadena de búsqueda que está compuesta por palabras claves que describen de manera más detallada lo que queremos buscar, la cadena de búsqueda tiene las siguientes versiones:

- **Cadena primera versión:** ALL ((“accessibility” OR “usability”) AND (“mobile applications” OR “apps”). Esta cadena de búsqueda inicial, reflejo un total de 54018 resultados, con lo cual se determina que la cadena no está desarrollada de forma correcta ya que se refleja una cantidad muy elevada de resultados.
- **Cadena segunda versión:** ALL ((“accessibility” OR “usability”) AND (“mobile applications” OR “apps”) AND (“KML” OR “KMZ”). Esta cadena de búsqueda obtiene 151 resultados, con lo que se determinó que la cadena arroja una cifra manejable.
- **Cadena tercera versión:** ALL ((“accessibility” OR “usability”) AND (“KML” OR “KMZ”) AND (“Responsive Web Designer” OR “Mobile Application”)), con esta cadena de búsqueda se obtuvo 40 resultados, con lo cual se puede determinar que la cadena arroja una cifra que es posible manejar y con la cual se va a realizar el desarrollo.

Es necesario recalcar que como base de datos digital en donde se va a buscar los artículos relacionados, se utilizará la plataforma “SpringerLink”.

### 3.2.2.Revisión sistemática de literatura con los trabajos obtenidos de los repositorios digitales académicos

Para la revisión sistemática de literatura se tomó 4 artículos principales, estos artículos fueron seleccionados en base a la relación directa que tienen con los objetivos específicos planteados para el desarrollo del trabajo. En la tabla 7 se presenta un resumen detallado de los 4 artículos que fueron seleccionados.

**Tabla 7**  
*Trabajos seleccionados de la plataforma SpringerLink*

Código	Título	Referencia
EP1	Evaluation of a Web-based and Mobile Ski Touring Application for GPS-enabled Smartphones.	(Haid, Kiechle, Göll, & Soutschek, 2008)
EP2	Geographic Information Systems.	(Stravros Kolios, 2017)
EP3	Mapping a City's Activity. A Project of Volunteered Geographic Information Using Mobile Mapping Collection.	(Borruso & Defend, 2016)
EP4	Geo-Web Services and New Exchange Formats to Develop Future Services Supporting Climate Change Adaptation Measures.	(Martin Scheinert, 2018)

**EP1 Evaluation of a Web-based and Mobile Ski Touring Application for GPS-enabled Smartphones**, en este proyecto se desarrolló y evaluó un sistema integrado de información móvil y web para excursiones de senderismo y esquí. En esta investigación se propone un desarrollo tecnológico en el área de turismo empleando herramientas tecnológicas como GPS, a fin de proporcionar información detallada sobre los diferentes recorridos existentes, así como también la funcionalidad de búsqueda, exploración y

descubrimiento de recorridos que se adapten a las necesidades individuales de los usuarios.

**EP2 Geographic Information Systems**, en la investigación, el autor argumenta que el concepto de sistema de información geográfica (SIG) se introdujo para cubrir todas las necesidades esenciales de la comunidad científica para proporcionar información espacial, analizar datos y crear mapas temáticos digitales a través de una computadora.

Esta investigación se enfoca en la determinación de cuatro principios básicos los cuales son: medición, mapeo, monitoreo y modelado, y se marcan un énfasis sobre elementos fundamentales para GIS como la superposición de diferentes tipos de información y datos. Se determina un aporte del profesor Ian L. McHarg quién fue uno de los primeros científicos que describieron el concepto de SIG y la utilidad de la superposición de mapas.

**EP3 Mapping a City's Activity. A Project of Volunteered Geographic Information Using Mobile Mapping Collection**, en este trabajo se aborda el concepto de información geográfica voluntaria basándose en el uso de una herramienta de recopilación de mapas móviles para la recuperación de datos geográficos de un trabajo de campo urbano.

La motivación principal para el desarrollo de este trabajo fue probar el potencial de crowdsourcing para recuperar datos utilizando un enfoque de abajo hacia arriba, confiando en un conjunto de "sensores urbanos" como recolectores de datos. Lo que permitió derivar datos geográficos de primera mano sobre un tema en particular y analizar

su distribución espacial por medio de sistemas de información geográfica y herramientas analíticas espaciales.

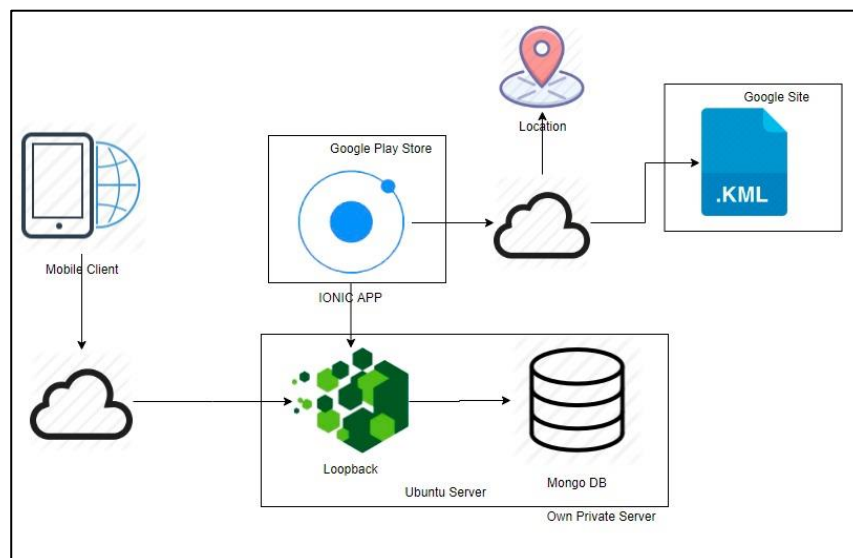
**EP4 Geo-Web Services and New Exchange Formats to Develop Future Services Supporting Climate Change Adaptation Measures**, en esta investigación se hace referencia y se describe el potencial que tienen los servicios Geo-Web actuales y los formatos innovadores de intercambio de datos para servicios climáticos futuros, la hipótesis es determinar si el uso de formatos de intercambio innovadores y servicios Geo-Web interoperables puede contribuir a un aumento de la calidad y un uso ampliado de los servicios web durante la definición de medidas regionales de adaptación al clima, para ello se centra en las medidas de adaptación al clima en las cuencas de pequeños ríos.

### **3.2.3. Definición de la estructura (software y hardware) del prototipo para el sistema de información**

El prototipo propuesto tiene una arquitectura orientada a servicios REST, en su estructura tenemos un servidor de base de datos ejecutándose en Windows 10, aquí se aloja la base de datos sobre MongoDB, para acceder a esta base de datos se consume los servicios web que el Loopback crea por defecto. En cuanto a la aplicación móvil, consume los archivos KML generados para cada ruta, estos archivos están alojados en la nube de Google, para la localización en tiempo real se consume desde la ubicación actual en la que se encuentra el teléfono celular.

Actualmente la aplicación móvil está alojada en la tienda de Google, denominada “Google Play Store”, a esta aplicación puede acceder cualquier usuario que tenga cuenta en dicha tienda ya que al tratarse de un prototipo, solo fue desarrollada y generada para

el sistema operativo Android.



**Figura 12.** Diagrama de arquitectura del prototipo propuesto

### 3.3. Fase II: Evaluar herramientas hardware y software para la solución planteada

En esta fase, el grupo de trabajo evaluó herramientas de hardware y software disponibles para la construcción del prototipo del sistema de información de plazas disponibles de aparcamiento, el objetivo de esta fase es seleccionar las mejores herramientas disponibles en el mercado tecnológico para construir una aplicación eficiente para los usuarios.

#### 3.3.1. Búsqueda de lenguajes de programación y modelado de entornos virtuales para desarrollo de aplicaciones móviles

En la actualidad existen una infinidad de herramientas, lenguajes y entornos de desarrollo que están disponibles para el desarrollo de aplicaciones móviles, y se pueden

elegir sin problema alguno ya que son Open Source y esto hace que esté disponible para cualquier desarrollador sin que esté tenga que pagar algún valor por adquirirlas.

En el ámbito del desarrollo móvil (celulares, tabletas, etc) están disponible varias opciones, es importante recalcar que el desarrollo nativo sin duda, es la mejor opción y para elegir una de ellas se debe tener en cuenta lo siguiente:

- Lenguajes y herramientas nativos de cada plataforma.

Cada plataforma de desarrollo móvil es un mundo diferente debido a los componentes, pluggings que requiere para tener un óptimo funcionamiento en cada sistema operativo móvil, razón por la cual cada una de estas plataformas utiliza un lenguaje de programación diferente. Esto obliga a tener que dominar gran variedad de lenguajes y herramientas diferentes influyendo considerablemente en el tiempo de desarrollo puesto que se multiplica el tiempo de desarrollo, ya que es necesario crear desde cero dos o tres versiones (dependiendo de la o las plataformas que se vaya a desarrollar) diferentes la misma aplicación.

Por ejemplo, si el desarrollador decide especializarse en aplicaciones para la plataforma popular Android debe saber programar con Android Studio y utilizar el entorno de Android para desarrollar sin problema alguno.

- Herramientas multiplataforma que compilan a código nativo.

En este tipo de herramientas se emplea un único lenguaje y se crean aplicaciones para todas las plataformas, con la condición principal que es



necesario adaptar algunos elementos para que funcione sin problema alguno en la que plataforma que se esté ejecutando. El aspecto positivo de este tipo de herramientas es que se debe conocer solo un lenguaje de programación, se puede reutilizar gran parte del código, excepto el uso de la interfaz, y también es posible generar aplicaciones nativas para cada entorno móvil.

En este tipo de herramientas la más conocido es Xamarin que está basada en el lenguaje C# que pertenece a Microsoft y se basa en la plataforma .NET, permite diseñar y crear aplicaciones móviles para todas las plataformas que están disponibles en el mercado, y la reutilización de gran parte de código fuente.

- Herramientas multiplataforma basadas en HTML.

Las herramientas que pertenecen a este grupo tienen una gran desventaja, y es que no tienen el mismo rendimiento que una aplicación móvil nativa esto implica que no tenga todos los accesos a las APIs nativas de cada plataforma en la que se está ejecutando.

### **3.3.2. Ejecutar pruebas de rendimiento en cada una de las herramientas encontradas**

Para el análisis del rendimiento en cada una de las herramientas utilizadas en el punto 3.3.1 se tomará como punto de partida y análisis los trabajos de titulación realizados en la Universidad Nacional de Chimborazo UNACH, cuyo objetivo principal es

el analizar el rendimiento de cada uno de los frameworks de desarrollo móvil (IONIC, Xamarin, Android Studio).

Para el análisis del framework IONIC, en (Ingeniería En Sistemas Y Computación, Luis Duchi Quishpe, & Lorena Paulina Molina Valdiviezo, 2019), utilizaron el sistema de notificación de actividades de la Universidad Nacional de Chimborazo (UNACH), para esto realizaron varias mediciones durante dos semanas utilizando el administrador de aplicaciones del sistema operativo Android obteniendo 400 datos de medición que analizados con el software estadístico BPM SPSS se obtuvieron los siguientes datos:

- Prueba de rendimiento en Memoria.

**Tabla 8**

*Prueba de normalidad en memoria IONIC vs Apache Cordova*

<b>Parámetros</b>	<b>Memoria IONIC</b>	<b>Memoria Apache Cordova</b>
<b>Media</b>	23,9490 Mb	29,7180 Mb
<b>Error Típico</b>	0,45855 Mb	0,48061 Mb
<b>p-valor</b>	0,000 Mb	0,000 Mb
<b>Varianza de la muestra</b>	21,027 Mb	23,099 Mb
<b>Mínimo</b>	11,50 Mb	22,10 Mb
<b>Máximo</b>	32,30 Mb	42,10 Mb
<b>Cuenta</b>	100 Mb	100 Mb

Fuente: (Ingeniería En Sistemas Y Computación et al., 2019)

- Prueba de rendimiento en CPU.

**Tabla 9***Prueba de normalidad en CPU IONIC vs Apache Cordova*

Parámetros	CPU IONIC	CPU Apache Cordova
<b>Media</b>	0,081700 Mb	0,150600 Mb
<b>Error Típico</b>	0,013707 Mb	0,022446 Mb
<b>p-valor</b>	0,000 Mb	0,000 Mb
<b>Varianza de la muestra</b>	0,018788 Mb	0,050381 Mb
<b>Mínimo</b>	0 Mb	0 Mb
<b>Máximo</b>	0,39 Mb	1,45 Mb
<b>Cuenta</b>	100 Mb	100 Mb

Fuente: (Ingeniería En Sistemas Y Computación et al., 2019)

Con respecto al análisis del framework Xamarin y Android Studio, en (Ingeniería, Actividades De La Unach, & Daniel Santos Galán Tutor Ing Geonatan Octavio Peñafiel Barros, 2018), emplearon el sistema de notificación de actividades de la Universidad Nacional de Chimborazo (UNACH), para esto realizaron varias mediciones durante dos semanas utilizando el administrador de aplicaciones del sistema operativo Android obteniendo un gran número de datos de medición que analizados con el software estadístico BPM SPSS se obtuvieron los siguientes datos:

- Prueba de rendimiento en Memoria.

**Tabla 10***Tabla de rendimiento de memoria Android Studio vs Xamarin*

Parámetros	Android Studio	Xamarin
<b>Media</b>	28,546 Mb	65,018 Mb
<b>Error típico</b>	0,928093974 Mb	0,9130536974 Mb
<b>Moda</b>	24,3 Mb	56 Mb

CONTINÚA 

<b>Varianza de la muestra</b>	86,13584242 Mb	147,6524 Mb
<b>Mínimo</b>	13,8 Mb	32 Mb
<b>Máximo</b>	68,9 Mb	98,3 Mb
<b>Cuenta</b>	100 Mb	100 Mb

Fuente: (Ingeniería et al., 2018)

- Prueba de rendimiento en CPU.

**Tabla 11**

*Tabla de rendimiento CPU Android Studio vs Xamarin*

<b>Parámetros</b>	<b>Android Studio</b>	<b>Xamarin</b>
<b>Media</b>	0,1497 Mb	0,1962 Mb
<b>Error típico</b>	0,031895807 Mb	0,02793202 Mb
<b>Moda</b>	0 Mb	0 Mb
<b>Varianza de la muestra</b>	0,101734253 Mb	0,07801976 Mb
<b>Mínimo</b>	0 Mb	0 Mb
<b>Máximo</b>	1,99 Mb	1,28 Mb
<b>Cuenta</b>	100 Mb	100 Mb

Fuente: (Ingeniería et al., 2018)

### **3.3.3. Selección de la herramienta de desarrollo móvil que sea adecuada para la construcción del prototipo**

Para la selección de la herramienta óptima en la que se basará el prototipo propuesto el grupo de trabajo analizó cada uno de los datos presentados en el punto 3.3.2, realizando una matriz de validaciones según los requerimientos y las necesidades que se deben solucionar con el desarrollo del prototipo, la tabla 12 resume los tiempos obtenidos con cada herramienta de desarrollo móvil, estos tiempos pertenecen tanto al rendimiento de memoria como al rendimiento del CPU:

**Tabla 12**  
*Consumo de recursos RAM y CPU*

Parámetros	IONIC	Android Studio	Xamarin
Consumo de RAM	23,9490 Mb	28,546 Mb	65,018 Mb
Consumo de CPU	0,081700 Mb	0,1497 Mb	0,1962 Mb

Adicional, existe un estudio en el cual se mencionan aspectos más relevantes que se debe tener en cuenta para elegir la herramienta más fácil y eficiente para el desarrollo de aplicaciones móviles, el estudio “Análisis comparativo entre frameworks de desarrollo para aplicaciones móviles híbridas” (De La, Cerón, Carlos, Fernando, & Moscol, 2017) y menciona algunas características importantes sobre el framework IONIC, en la tabla 13 se resumen las características tomadas en cuenta por el autor y las cuales favorecen a IONIC para ser la mejor opción para el desarrollo de aplicaciones móviles.

**Tabla 13**  
*Indicadores de evaluación para el framework IONIC*

Parámetros	Valor	
Horas de esfuerzo de aprendizaje	32 h	
Dificultad de aprendizaje (1-Fácil, 2-Intermedio, 3-Difícil)	1	
Número de plataformas compatibles	4	
	Crear (Create)	1372 ms.
Tiempo de respuesta para el	Leer (Read)	996 ms.
CRUD	Actualizar (Update)	1203 ms.
	Eliminar (Delete)	951 ms.

### **3.4. Fase III: Diseño del prototipo**

En la tercera fase, netamente se desarrolló las funcionalidades que abarca la aplicación móvil, estas son las que ejecutará en tiempo real el sistema de información de plazas disponibles para aparcamiento vehicular, se utilizó las herramientas de hardware y software que fueron seleccionadas en la fase anterior. El objetivo principal de esta fase es presentar un prototipo funcional que permita ejecutar pruebas de validación en la comunidad universitaria ESPE.

#### **3.4.1. Desarrollo de la interfaz de ingreso de los datos relacionados con el usuario**

Para el desarrollo de las interfaces de registro de datos de cada usuario del sistema se modelo cada una de las pantallas que fueron necesarias, este prototipo de pantallas fue creado en el software Adobe XD el cuál facilitó la generación de los bosquejos de pantallas iniciales.

Como todas las aplicaciones, la primera ventana que se presenta al usuario es la de inicio de sesión, está ventana contiene dos opciones principales, que son: Registro de Datos o Inicio de Sesión, en la figura 13 se puede apreciar el prototipo de la pantalla inicial de la aplicación denominada “EspePark”.



**Figura 13.** Prototipo pantalla inicial de la aplicación EspePark

Una de las vistas principales que el software presenta es la ventana de registro de datos de usuario, esta ventana contiene cuadros de texto para que el usuario registre datos tales como:

- Nombre Completo.
- Cédula de identidad.
- Número de celular.
- Correo electrónico.
- Tipo de Usuario.
- ID de estudiante ESPE.
- Nombre de usuario.

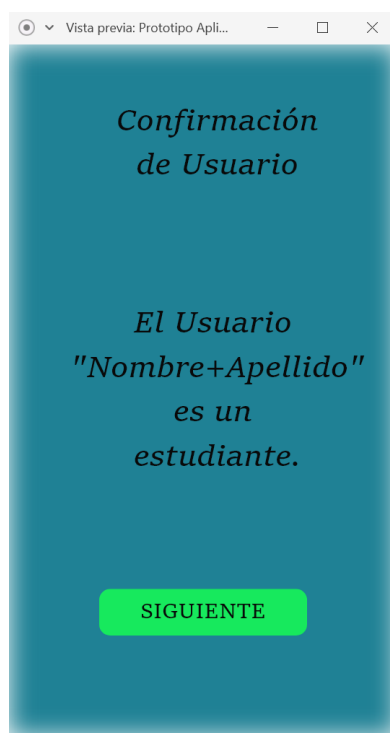
De igual manera se colocó una opción de guardar datos, está se ejecuta mediante un botón que está integrado en la pantalla de registro de datos.

The image shows a mobile application prototype window titled "Vista previa: Prototipo Apli...". The main content is a registration form with a teal background. At the top, the title "Datos Usuario" is displayed in a serif font. Below the title are six white input fields with rounded corners, each containing a label: "Nombre", "Apellido", "ID Espe", "Usuario", "Contraseña", and "Confirmar Contraseña". At the bottom of the form is a bright green button with the text "SIGUIENTE" in white capital letters.

**Figura 14.** Registro de datos del usuario

Una vez que el usuario ya registró los datos solicitados por la aplicación, la siguiente interfaz gráfica que se presenta es la de confirmación de datos, en esta ventana, la aplicación presenta el resumen de la información que el usuario ingresó como datos personales para un mejor servicio de ubicación y direccionamiento hacia una plaza disponible de aparcamiento vehicular. La figura 15 muestra la interfaz de confirmación de datos registrados por el usuario.





**Figura 15.** Prototipo de ventana de confirmación de datos de usuario

Luego de que el usuario confirma que están correctos los datos ingresados en la ventana anterior, el siguiente paso es el registro de datos del vehículo con el cuál ingresa a los diferentes estacionamientos que están disponibles en las instalaciones de la Universidad de las Fuerzas Armadas ESPE. Los datos del vehículo que son solicitados para la aplicación son los siguientes:

- Marca.
- Modelo.
- Número de Placa.
- Año.

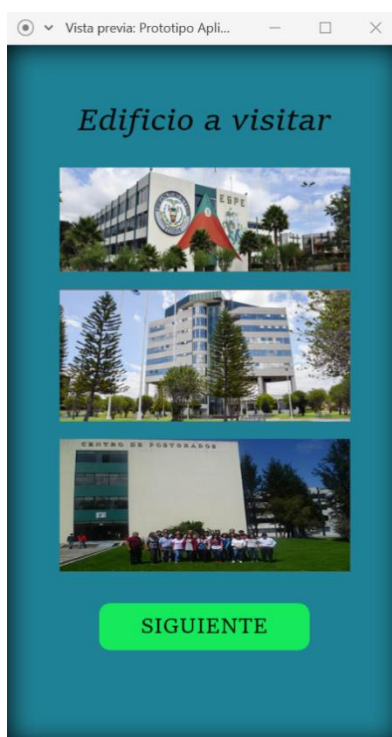
En la figura 16 se puede apreciar la ventana en la cual el usuario debe registrar los datos de su vehículo.

La imagen muestra un prototipo de una ventana de registro de datos del vehículo. La ventana tiene un título "Datos Vehículo" en un fondo azul oscuro. Hay tres campos de entrada de texto blancos con el texto "Placa", "Marca" y "Color" respectivamente. En la parte inferior hay un botón verde con el texto "SIGUIENTE". La ventana está en un navegador con el título "Vista previa: Prototipo Apli...".

**Figura 16.** Ventana para el registro de datos del vehículo

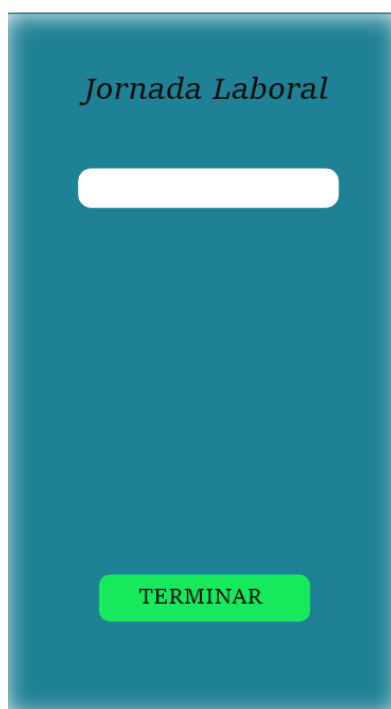
Al terminar el registro de los datos del vehículo, lo siguiente es seleccionar el área o edificio del campus, para este campo el usuario debe tener presente que seleccionará el área o edificio en el cuál desarrollará la mayor parte de su jornada laboral o académica, esto con el fin de almacenar la referencia para que la aplicación asigne un espacio de estacionamiento tan cercano como sea posible al lugar registrado.

En la figura 17 se muestra el prototipo de la ventana en la que la aplicación presentará al usuario los edificios o zonas en las que se desarrolla actividades laborales o académicas, esta información se presentará mediante gráficos.



**Figura 17.** Interfaz para el registro del lugar a visitar

Por último, la aplicación solicitará que se registre el tipo de jornada que el usuario cubrirá en las instalaciones de la Universidad, este dato servirá a la aplicación para validar la disponibilidad de cada uno de los espacios de aparcamiento al momento de los cambios de jornada. Al final del registro de la jornada laboral, el usuario debe confirmar la información y será el último paso que deberá completar para empezar a utilizar la asignación de plazas de aparcamiento para vehículos.



**Figura 18.** Interfaz para el registro de la jornada laboral

Luego de que el usuario finalice el registro de sus datos personales, datos de su vehículo y datos de información relacionados con el lugar que visitará y la jornada que cumplirá en la ESPE, lo siguiente es iniciar sesión con el usuario y la contraseña proporcionados en el registro de datos, para esta acción la aplicación presenta una interfaz de inicio de sesión, en la figura 19 se aprecia la ventana de inicio de sesión mediante la cual el usuario podrá acceder a los servicios de geolocalización de una plaza disponible de aparcamiento.



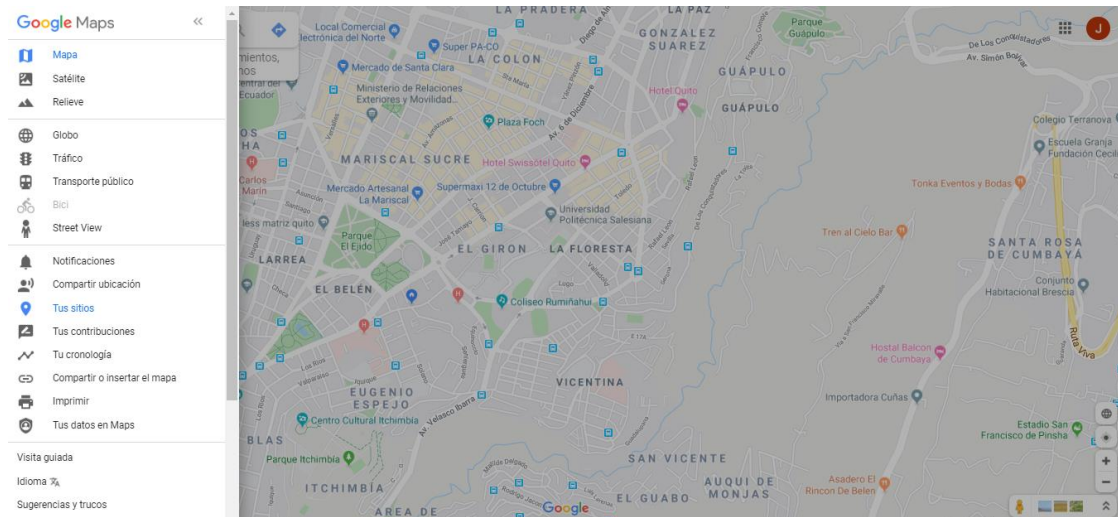
**Figura 19.** Interfaz de inicio de sesión de usuarios

Una vez culminado y diseñado el bosquejo general de las interfaces de usuario que tendrá la aplicación, se procedió a crear en el framework IONIC cada una de las interfaces de la aplicación, esto ya se lo creó en un ambiente más profesional y realista, con esto se dio inicio al desarrollo del prototipo propuesto para la ubicación y dirección de plazas disponibles de aparcamiento.

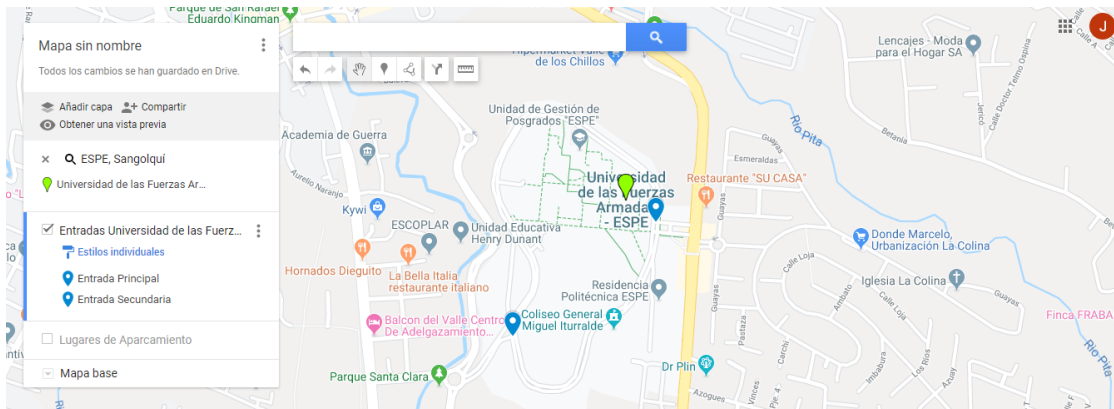
### **3.4.2. Digitalización del espacio de parqueo seleccionado para la construcción del prototipo**

Para el desarrollo de la digitalización del sector de aparcamiento seleccionado se empleó la herramienta que brinda "Google Maps" denominada "Sitios", la cual permite la creación y determinación de "Mapas", esta herramienta permite la determinación de

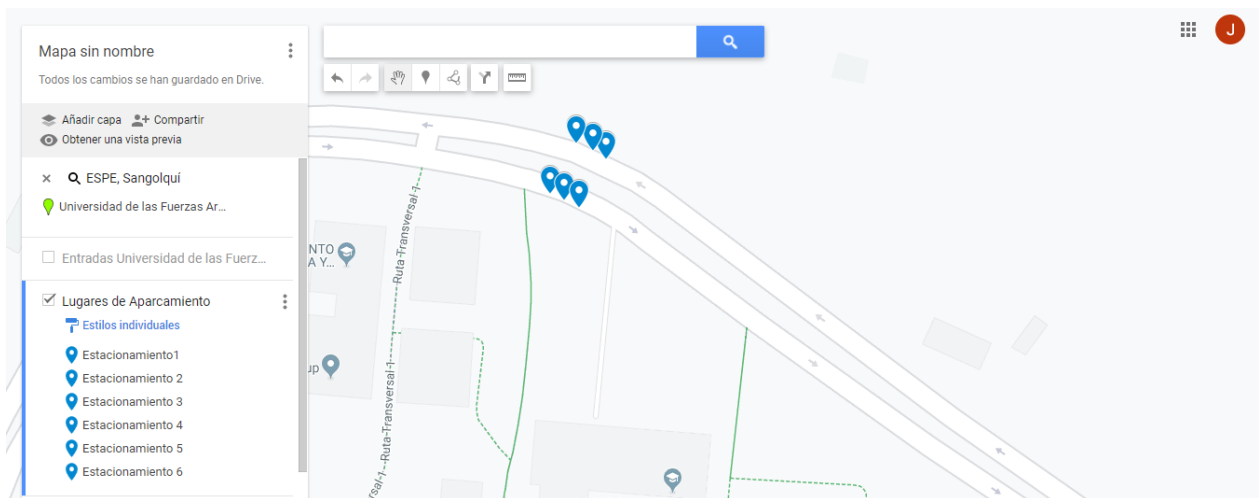
puntos de ubicación sobre las cuales se realiza el proceso de trazado de ruta más cercana entre dos puntos GPS, de igual forma permite la exportación tanto de la digitalización inicial como de las entradas de la Universidad de las Fuerzas Armadas ESPE, la digitalización de lugares de aparcamiento a utilizar así como de las rutas que se establecen en formato “.kml” ó “.kmz”, estos archivos son publicados en la plataforma denominada “Google Sites” para ser utilizados posteriormente en el desarrollo del aplicativo.



**Figura 20.** Herramienta Sitios de Google Maps



**Figura 21.** Digitalización de las entradas principales de la Universidad ESPE



**Figura 22.** Digitalización de los lugares de aparcamiento vehicular

### 3.4.3. Programación para la detección y asignación de la plaza más cercana al área de trabajo del usuario

Para el desarrollo de la funcionalidad de detección y asignación de plazas cercanas al área de trabajo del usuario, el usuario previamente debe haber culminado con:

- El registro inicial de la información básica de contacto.
- El registro de los datos de su vehículo.

- c) El registro del sector de visita, estudio o trabajo.
- d) El registro de la jornada de visita, estudio o trabajo.

Al finalizar con estos registros el usuario puede acceder al módulo de ESPEPARK en donde verifica:

- a) El registro completo de:
  - a. Datos del vehículo.
  - b. Selección de un lugar de visita, estudio o trabajo.
  - c. Registro de la jornada de visita, estudio o trabajo.
- b) La disponibilidad de lugares de aparcamiento asociados al registro de un lugar de visita, estudio o trabajo, esto lo realiza en base a:
  - a. La determinación de los estados de los lugares de aparcamiento que se encuentren almacenados en la Base de Datos, al momento se maneja 3 estados:
    - i. ASIGNADO. – El lugar de aparcamiento se almacena con este estado en el momento en el que el usuario solicita un sitio de aparcamiento.
    - ii. OCUPADO. – El lugar de aparcamiento se almacena con este estado en el momento que el usuario llega al sitio asignado a él, y selecciona o indica en el aplicativo que ha llegado a su destino.
    - iii. LIBRE. – El lugar de aparcamiento se almacena con este estado en el momento que:
      - 1. El usuario indique en el aplicativo su salida.



2. Al sobrepasar el tiempo registrado, en el apartado de *“Jornada laboral”*.
3. Al detectarse la ubicación del dispositivo fuera del perímetro de la Universidad de las Fuerzas Armadas ESPE.

#### **3.4.4. Programación de la ruta que dirige al área asignada para el usuario**

Una vez que se ha determinado el sector del parqueo de acuerdo con el lugar a visitar la aplicación valida la disponibilidad de plaza que se encuentren en estado libre, válida la cercanía a cualquiera de las entradas que dispone la Universidad, identifica la entrada más cercana y procede a graficar la ruta que el usuario debe seguir para llegar a la plaza recomendada.

#### **3.5. Fase IV: Implementación del prototipo**

Una vez que el diseño del prototipo fue validado con un correcto y óptimo funcionamiento, el grupo de trabajo alojó la aplicación móvil en un ambiente de producción, es decir, alojó en la tienda de aplicaciones que están disponibles para el Sistema Operativo Android que actualmente existe, dando apertura a los usuarios para la validación del prototipo.

##### **3.5.1. Primer plan de pruebas internas ejecutadas por el grupo de trabajo**

Este punto se relaciona con la fase de pruebas y resultados, por lo tanto, se va a desarrollar en el siguiente capítulo, que es el capítulo 4 denominado Pruebas y Resultados.

### **3.5.2. Alojamiento en las tiendas de aplicaciones disponibles para los diferentes dispositivos móviles**

Este punto está enfocado en las diferentes configuraciones que se debe realizar a fin de publicar el aplicativo móvil en la plataforma de “Google Play Store”, para lo cual debemos realizar el siguiente procedimiento:

1. Agregar la plataforma de exportación de la aplicación.

Previo a todas las configuraciones que se deben realizar, se debe agregar la plataforma con la que se generará la versión de lanzamiento, para ello desde la terminal por línea de comandos desde la raíz de nuestra aplicación, digitamos el siguiente comando:

*ionic cordova platform add android*

Con este comando obligará a la aplicación a generar tanto los recursos de la aplicación, así como la generación de versión de lanzamiento, enfocadas únicamente a la plataforma de Android.

2. Generar los recursos necesarios para la aplicación.

En este punto se generará los recursos necesarios de la aplicación que servirán para la especificación del icono identificativo de la aplicación, así como del splash o imagen de carga de la aplicación. Para generar estos recursos se debe tener en cuenta lo siguiente:

- a) Que el tamaño del icon y del splash deben ser inferiores a 2MB, debido a que cuando IONIC + Córdova, genera estos recursos las imágenes que son generadas para los diferentes dispositivos tienden a aumentar

su tamaño y esto puede generar errores al momento de generar la versión de lanzamiento.

- b) Las dimensiones que debe tener la imagen enfocada al icono de la aplicación son: 1024x1024 y debe ser almacenada en la carpeta de resources con el nombre de “icon”.
- c) Las dimensiones que debe tener la imagen enfocada al splash o imagen inicial de carga de la aplicación son: 2732x2732 y debe ser almacenada en la carpeta de resources con el nombre de “splash”.

Para generar los recursos lo realizamos desde la terminal por línea de comandos desde la raíz de nuestra aplicación, digitamos el siguiente comando:

*ionic cordova resources*

### 3. Deshabilitar el complemento de consola de depuración.

Antes de generar la versión de lanzamiento de la aplicación se debe considerar los complementos empleados durante el proceso de desarrollo que no son de utilidad en el modo de producción, como por ejemplo el complemento de consola de depuración, para ello se debe eliminarlo antes a través del siguiente comando:

*ionic cordova plugin rm cordova – plugin – console*

### 4. Generar una versión de lanzamiento de la aplicación.

Una vez el proceso anterior finalice, se procede a la generación de la versión de lanzamiento de la aplicación enfocada a la plataforma de Android a través del siguiente comando:

*ionic cordova build android --release --prod*

Este comando tomará como directrices de aplicación y depuración la información que se encuentra en el archivo “config.xml”. Al finalizar este proceso nos presentará la ubicación en donde se encuentra la apk que ha sido generada, pero sin una firma válida aún.

*...\platforms\android\app\build\outputs\apk\release\app -- release  
-- unsigned.apk*

#### 5. Firmar la versión de lanzamiento.

En este paso se debe generar una firma para la apk generada previamente y posterior a ello ejecutar una utilidad de alineación para optimizar y preparar la aplicación o apk para la tienda de aplicaciones de Google.

Para realizar esto inicialmente es generar un archivo correspondiente al almacenamiento de claves de la aplicación a generar, por lo que es necesario emplear el comando “keytool” que es una herramienta que incluye el JDK, para lo cual ejecutaremos lo siguiente:

*keytool --genkey --v --keystore espePark.keystore --alias espePark  
--keyalg RSA --keysize 2048 --validity 10000*

Esto nos generará un archivo “. keystore” el cuál nos permitirá realizar actualizaciones sobre la aplicación.

#### 6. Generar un archivo apk correspondiente a la aplicación y su respectiva firma.

En este paso emplearemos el comando “jarisgner”, que es una herramienta también incluida en el JDK:

```
jarsigner - verbose - sigalg SHA1withRSA - digestalg SHA1  
- keystore espePark.keystore app - release  
- unsigned.apk espePark
```

Este comando firma la aplicación, y finalmente debemos alinear y optimizar nuestra “.apk”, para ello emplearemos el comando “*zipalign*”, que es una herramienta también incluida en el JDK:

```
zipalign -v 4 app - release - unsigned.apk espePark.apk
```

Con esto realizado ya tenemos el archivo “.apk” final firmado con lo que se procedería con la publicación en “Google Play Store”.

#### 7. Subir el archivo “.apk” firmado a Google Play Store.

Este paso se enfoca en la carga y publicación de la aplicación en la tienda de aplicaciones para Android, para ello se debe configurar y adquirir una cuenta de desarrollador en la “*Consola de desarrollador de Google Play Store*”, una vez creada la cuenta se debe ubicar la opción de “*Publicar una aplicación de Android en Google Play*”, la consola de administración de Google Play Store nos solicitará el ingreso de información como:

- a. Nombre de la aplicación.
- b. Descripción de la aplicación.
- c. El archivo “.apk” para su publicación.

### **3.6. Fase V: Validación del prototipo mediante pruebas**

Es la fase final de la metodología propuesta, en esta fase se validó el prototipo del sistema de información de plazas disponibles para aparcamiento vehicular, esta validación se ejecutó tomando una pequeña muestra de estudiantes, docentes e integrantes de la comunidad ESPE los cuales dispongan de un vehículo y sean usuarios de los parqueaderos de la Universidad, una vez que iniciaron con el uso de la aplicación se ejecutó una encuesta con la cual logramos medir el grado de satisfacción de los usuarios después de haber sido recomendado y dirigido hacia una plaza de aparcamiento disponible para su vehículo.

#### **3.6.1. Ejecución de pruebas por parte de los usuarios del área de parqueo de vehículos que posee la Universidad ESPE**

Este punto se relaciona con la fase de pruebas y resultados, por lo tanto, se va a desarrollar en el siguiente capítulo, que es el capítulo 4 denominado Pruebas y Resultados.

#### **3.6.2. Aplicación de una encuesta para conocer el grado de satisfacción al utilizar el sistema de información alojado en una tienda móvil**

Para conocer y determinar el grado de satisfacción de los usuarios luego de hacer uso de la aplicación móvil, se empleó la siguiente encuesta:

##### **3.6.2.1. Objetivos:**

- Determinar el porcentaje de aceptación de la aplicación móvil en la comunidad universitaria mediante la ejecución de una encuesta para conocer el tiempo que disminuyó en la búsqueda de un área de aparcamiento libre.

- Conocer el nivel de aceptación que tiene la aplicación móvil en la comunidad universitaria.

### **3.6.2.2.Población o Universo seleccionado:**

La ejecución de la encuesta se lo hace en un universo seleccionado, este universo fue los estudiantes de la Universidad de las Fuerzas Armadas ESPE, estudiantes o docentes que pertenezcan específicamente al Departamento de Ciencias de la Computación (DCCO), y que dispongan de un automóvil, un teléfono celular que soporte la versión de la aplicación desarrollada y también que tenga acceso al área de aparcamiento ubicada en la parte posterior del DCCO.

### **3.6.2.3.Identificación de la información a recolectar**

La información a recolectar es la información necesaria y suficiente para realizar un análisis profundo y que permita cumplir con los objetivos de la encuesta que fueron planteados, entre los más aspectos más importantes para la recolección de la información tomamos en cuenta los siguientes:

- Tiempo empleado en la búsqueda de una plaza disponible de aparcamiento vehicular.
- Nivel de dificultad en la instalación y navegación por las diferentes ventanas de la aplicación móvil.
- Tiempo ahorrado en la búsqueda de una plaza disponible de aparcamiento vehicular.
- Nivel de aceptación de la nueva forma para detectar plazas disponibles de aparcamiento.

### 3.6.2.4. Diseño del cuestionario

El cuestionario de la encuesta debe tener las preguntas necesarias, de modo que permitan obtener la información suficiente y requerida para cumplir con los objetivos, las preguntas planteadas fueron las siguientes:

- ¿Cómo valoras esta iniciativa de crear una aplicación para la rápida detección de plazas disponibles de aparcamiento?
  - 1 (muy mala)
  - 2 (mala)
  - 3 (normal)
  - 4 (buena)
  - 5 (muy buena)
- ¿Estarías dispuesto a probar esta nueva aplicación?
  - 1 (nada dispuesto)
  - 2 (poco dispuesto)
  - 3 (dispuesto)
  - 4 (muy dispuesto)
  - 5 (totalmente dispuesto)
- Aproximadamente, ¿Cuál fue el tiempo que empleaste en la búsqueda de una plaza disponible de aparcamiento vehicular?
  - Menos de un minuto
  - De 1 a 3 minutos
  - Más de 3 minutos



- ¿Qué tan útil crees que fue el uso de nuestra aplicación?
  - 1 (nada útil)
  - 2 (poco útil)
  - 3 (normal)
  - 4 (útil)
  - 5 (muy útil)
  
- ¿Recomendarías esta aplicación a un amigo?
  - 1 (no recomendaría)
  - 2 (poco recomendada)
  - 3 (normal)
  - 4 (si la recomendase)
  - 5 (definitivamente la recomendaría)
  
- ¿Cuál fue el nivel de dificultad que tuviste para manejar nuestra aplicación?
  - 1 (muy difícil)
  - 2 (difícil)
  - 3 (normal)
  - 4 (fácil)
  - 5 (muy fácil)
  
- ¿Con que sistema operativo preferirías utilizarla?
  - Android
  - iOS
  - Windows Phone

### 3.6.2.5.Cálculo de la muestra

La muestra es el número de personas que representará a la población total que será encuestada, por ende, esto será el número de encuestas que se debe realizar para la validación del prototipo. Para obtener la muestra se va aplicar la siguiente fórmula de muestra que es una fórmula de estadística, su estructura es la siguiente (K, 2015):

$$n = \frac{Z^2 * p * q * N}{Ne^2 + Z^2 * p * q}$$

En donde:

- $n$ : muestra, número de personas representativo de la población a la que se va a encuestar.
- $N$ : población, es el conjunto de personas con características similares de las cuales se desea obtener información.
- $Z$ : nivel de confianza, este indicador ayuda a medir la confiabilidad de los resultados, es recomendable utilizar un nivel de confianza de 95% ya que los resultados tendrán mayor confiabilidad.
- $e$ : grado de error, mide el porcentaje de error que puede existir en los resultados obtenidos, lo recomendable es emplear un error de 5% o 10%, es importante tener en cuenta que mientras menor sea el margen de error los resultados tienen mayor validez.
- $p$ : probabilidad de ocurrencia, se trata de la probabilidad de que ocurra el evento, usualmente se utiliza una probabilidad de ocurrencia de 50%.

- $q$ : probabilidad de no ocurrencia, probabilidad de no ocurrencia de un evento, lo recomendable es utilizar una probabilidad de no ocurrencia de 50%.

Para determinar el tamaño de la muestra y el número de encuestas a realizar, se tomó en cuenta los siguientes datos:

- $n$ : es el número total de encuestas que se debe realizar para la validación y análisis de los resultados sobre el prototipo creado.
- $N$ : actualmente en el Departamento de Ciencias de la Computación de la Universidad de las Fuerzas Armadas ESPE son alrededor de 700 integrantes, docentes, estudiantes y personal administrativo. Para la toma de la población se toma en cuenta solo a las personas que disponen de un automóvil, estas personas son alrededor de 100.
- $Z$ : el nivel de confianza fue establecido en el 90%.
- $e$ : se eligió un grado de error de 5%.
- $p$ : la probabilidad de ocurrencia es la recomendada, 50%.
- $q$ : la probabilidad de no ocurrencia es la recomendada, 50%.

$$n = \frac{(0,96)^2 * (0,5) * (0,5) * (100)}{(100) * (0,05)^2 + (0,96)^2 * (0,5) * (0,5)}$$

$$n = \frac{11,9025}{0,4804}$$

$$n = 24,78$$

Una vez realizados los cálculos en la fórmula para determinar la muestra a la que se debe aplicar la encuesta, se obtiene que se debe aplicar la encuesta a 25 personas con las cuáles será suficiente determinar la utilidad de la aplicación móvil desarrollada.

## **CAPÍTULO IV**

### **PRUEBAS Y RESULTADOS**

#### **4.1. Introducción**

En este capítulo se presenta el protocolo de pruebas que fue planteado para la realización y ejecución de las pruebas, los resultados obtenidos durante la ejecución y utilización de la aplicación en la fase de pruebas.

#### **4.2. Protocolo de validación**

El protocolo de validación fue creado para aplicar en los parqueaderos de la Universidad de las Fuerzas Armadas ESPE, esta fase de pruebas tuvo dos fases: la fase de pruebas internas que fueron ejecutadas por el grupo de trabajo en donde se validó que todo esté funcionando correctamente y la otra fase que fueron ejecutadas con estudiantes que pertenecen al Departamento de Ciencias de la Computación (DCCO) mismos que se validó la disponibilidad de un vehículo.

#### **4.3. Primer plan de pruebas internas ejecutadas por el grupo de trabajo**

Para la ejecución del primer plan de pruebas realizadas por el grupo de trabajo se utilizó un automóvil de la marca Chevrolet Grand Vitara, este vehículo fue utilizado para realizar el primer plan de pruebas de la aplicación desarrollada.

El proceso de ejecución de las pruebas es el siguiente:

1. El usuario descarga la aplicación móvil alojada en la tienda de aplicaciones móviles denominada Play Store, para localizar la aplicación en la tienda de Android se deberá digitar el nombre "EspePark".

2. Una vez descargada la aplicación, el usuario debe buscar en la pantalla principal de su móvil, la aplicación denominada “EspePark” será la que ejecuta el sistema de información de plazas disponibles de aparcamiento vehicular.

Captura de la pantalla del teléfono con el ícono de la aplicación

3. Una vez que el usuario ejecute la aplicación, la primera ventana que se va a presentar es la de ingreso de datos para iniciar sesión o registrarse en el caso de que sea la primera vez que se accede a la aplicación, para este caso es la primera vez que se accede a la aplicación y el usuario deberá proporcionar todos sus datos tanto personales como del vehículo, lugar a visitar y jornada en la que estará en la Universidad.



**Figura 23.** Pantalla principal de la aplicación Espe Park

4. El usuario selecciona la opción Registrarse, ya que es la primera vez que ingresa al aplicativo, por lo tanto, tiene que registrar sus datos, primero deberá registrar sus datos personales. Una vez finalizado el registro de datos personales, el usuario selecciona la opción Registrarse.



The image shows a mobile application screen for user registration. The background is a dark teal overlay on a photo of a smiling woman with long blonde hair. The title 'INGRESO DE INFORMACIÓN' is centered at the top in white. Below the title are four input fields, each with a label and a value, separated by horizontal lines. The labels are in a smaller font, and the values are in a larger font. The last field has a small 'x' icon to its right.

**INGRESO DE INFORMACIÓN**

Nombres Completos \*  
ALEJANDRO


Apellidos Completos \*  
TORRES

C.I. \*  
1722340138

Número de celular \*  
0992831501

**Figura 24.** Registro de datos personales del usuario





Correo electrónico \*  
jatorres19@espe.edu.ec

Tipo de usuario \*      Estudiante ▾

ID ESPE \*  
S00350376

Usuario \*  
jatorres19

Contraseña \*  
.....

Confirmar contraseña \*  
.....

REGISTRARSE

CANCELAR

**Figura 25.** Registro de datos personales del usuario

Es necesario recalcar que los campos de cédula, correo electrónico y contraseña están debidamente validados, es decir, no se puede ingresar cualquier tipo de información.

5. La siguiente pantalla que la aplicación presenta es el registro de datos del vehículo, el usuario tiene que proporcionar la información solicitada del vehículo con el cuál accede a los parqueaderos de la Universidad, estos datos se recolectan con el fin de tener un registro del acceso y uso de las plazas disponibles de aparcamiento que han sido asignadas al usuario.

Una vez que el usuario termina el registro de los datos del vehículo, tiene que dar clic en “Confirmar Registro”.

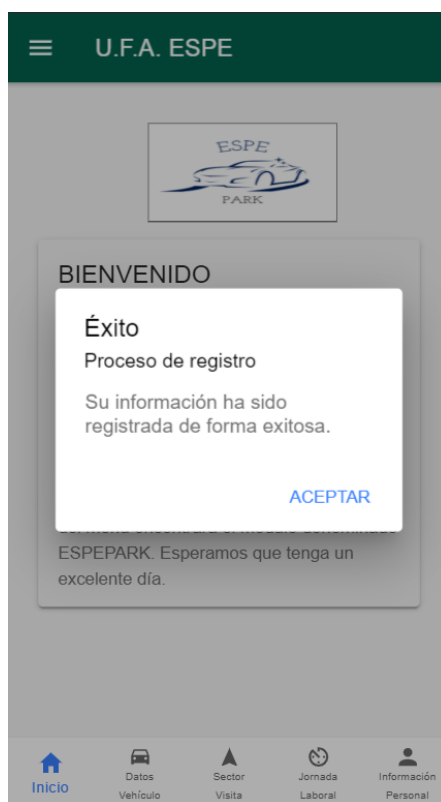
The image shows a mobile application screen titled "REGISTRO VEHICULAR". The screen contains a form with the following fields:

- Marca \*  
CHEVROLET
- Modelo \*  
GRAND-VITARA
- Número de placa \*  
PBT-1882
- Generación/Año \*  
2008

Below the form is a button labeled "CONFIRMAR REGISTRO". At the bottom of the screen is a navigation bar with five icons and labels: "Inicio", "Datos Vehículo", "Sector Visita", "Jornada Laboral", and "Información Personal".

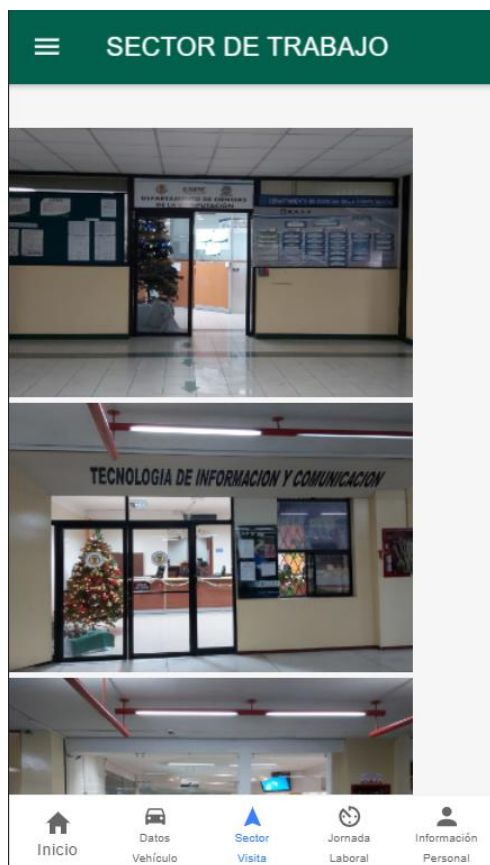
**Figura 26.** Registro de datos del vehículo

La aplicación informa al usuario que el registro de datos fue exitoso, de la misma manera si existe algún error la aplicación también informa al usuario.



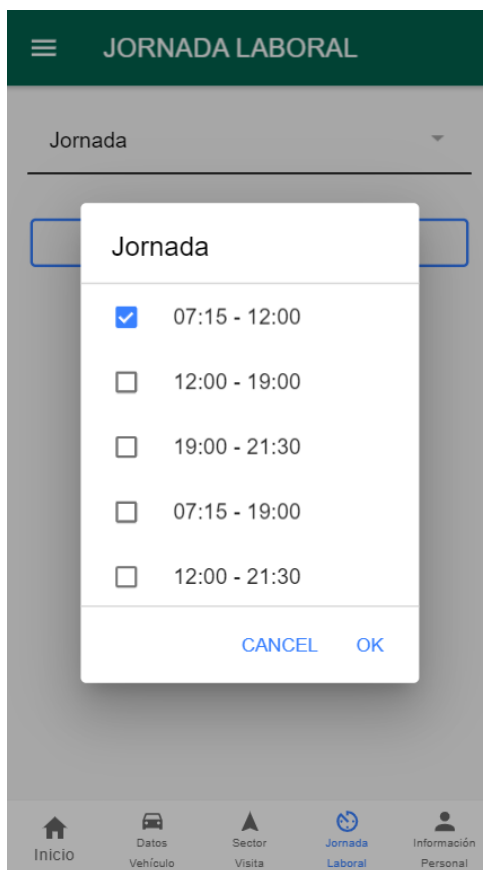
**Figura 27.** Confirmación de registro exitoso de los datos del vehículo

6. La siguiente ventana que presenta la aplicación es la de registro del área, edificio, departamento o bloque que el usuario va a visitar, este dato se solicita con el fin de recomendar una plaza de aparcamiento que sea cercana al lugar a visitar. De igual manera, una vez que el usuario termine el registro del lugar a visitar debe seleccionar la opción “Confirmar Registro”.



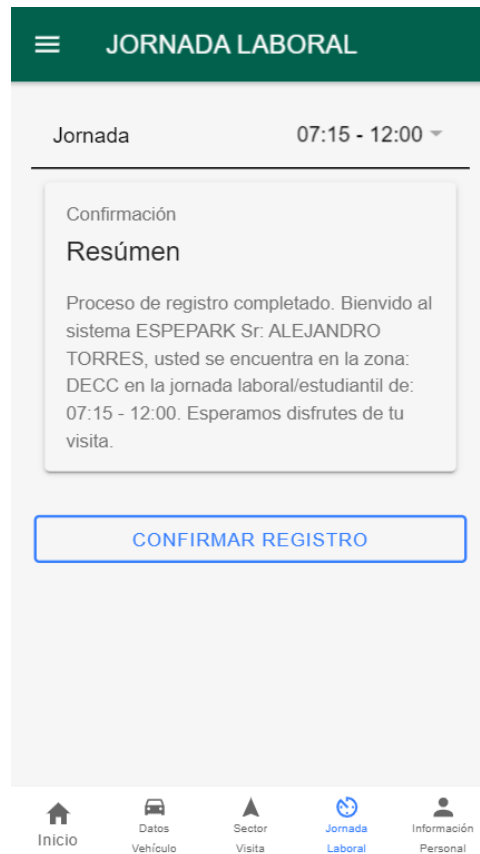
**Figura 28.** Registro del área, edificio, bloque o departamento a visitar

7. Como último paso de registro, el usuario debe proporcionar información sobre la jornada laboral que cumplirá en las instalaciones de la Universidad, este dato ayudará a la aplicación a liberar el espacio asignado cada que se cumpla la hora final de la jornada.



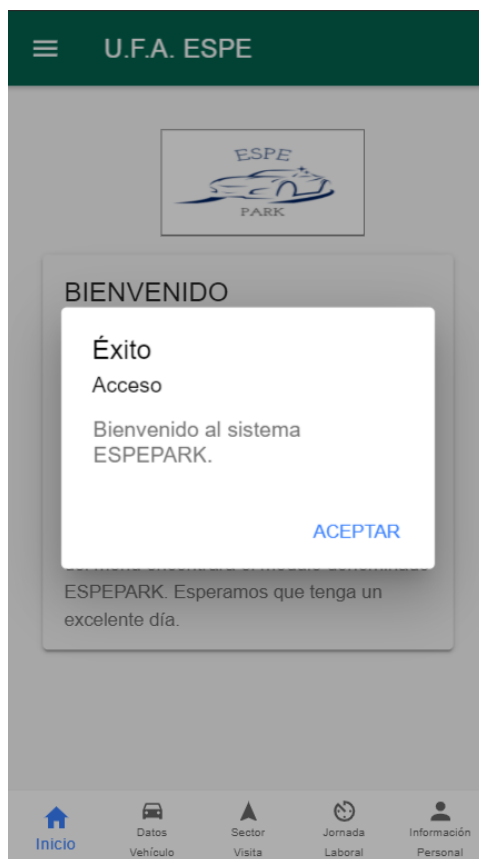
**Figura 29.** Registro de la jornada laboral del usuario

8. Al finalizar el registro de la jornada laboral, que es el último paso, la aplicación presenta un resumen con los datos proporcionados por el usuario, en esta ventana el usuario puede confirmar los datos proporcionados o corregir los mismos en caso de que exista algún error en la información.



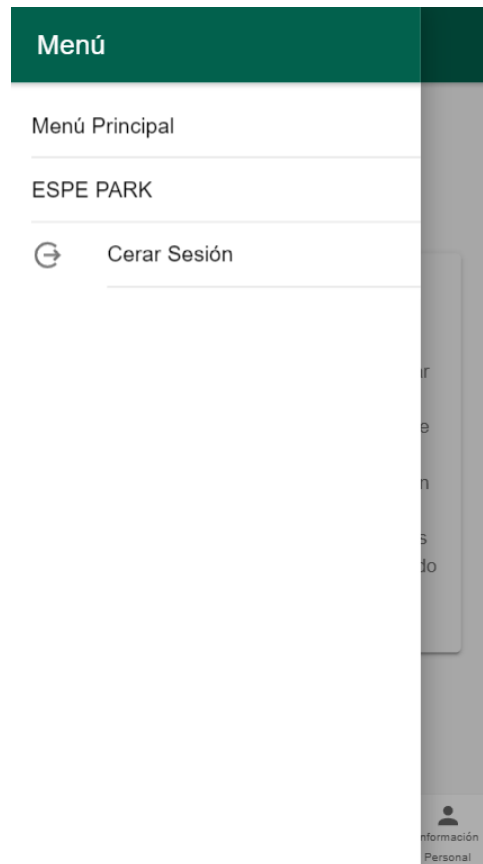
**Figura 30.** Confirmación de datos de usuario

9. Una vez que el usuario confirma los datos proporcionados en las diferentes ventas de registro, ya puede ser uso de la recomendación de una plaza disponible de aparcamiento, es decir, el usuario ya puede solicitar una plaza libre de aparcamiento. Para esto el usuario una vez que realizó el inicio de sesión, la aplicación presenta la pantalla de bienvenida.



**Figura 31.** Mensaje de bienvenido al usuario luego de iniciar sesión

10. El usuario debe solicitar una plaza disponible para aparcar su vehículo, para esto debe seleccionar la opción "ESPE PARK" la cual redirige al mapa interno de la Universidad e inmediatamente gráfica la ruta hacia el lugar libre para estacionamiento.



**Figura 32.** Solicitar una plaza disponible de aparcamiento

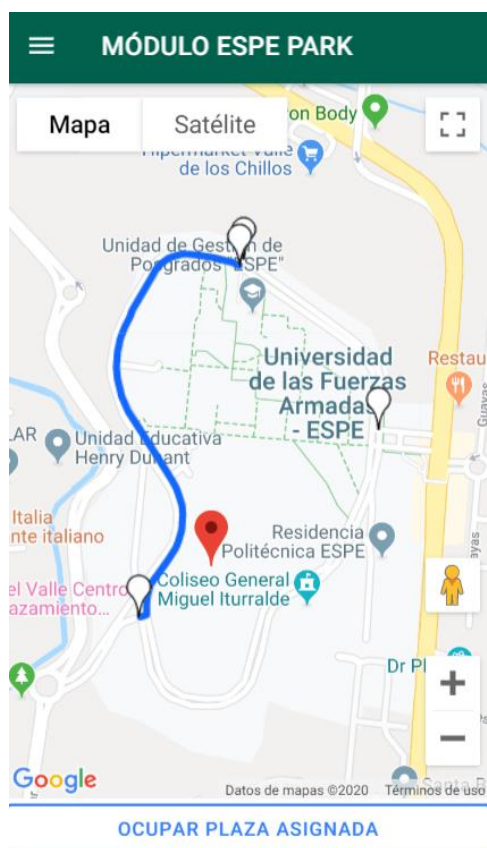
11. Para las pruebas internas realizadas por el grupo de trabajo, se ejecutó la aplicación sobre un teléfono Samsung J5 con versión de Android 8.0.0. Se solicitó una plaza disponible de aparcamiento, la ruta que la aplicación recomendó fue la siguiente:





**Figura 33.** Usuario solicita una plaza disponible de aparcamiento

Una vez que el usuario solicita una plaza disponible de aparcamiento vehicular, la aplicación grafica la ruta por la que el usuario debe seguir para llegar al objetivo. La siguiente imagen muestra como la aplicación grafica la ruta que el usuario debe seguir para llegar a la plaza disponible de aparcamiento que fue asignada.



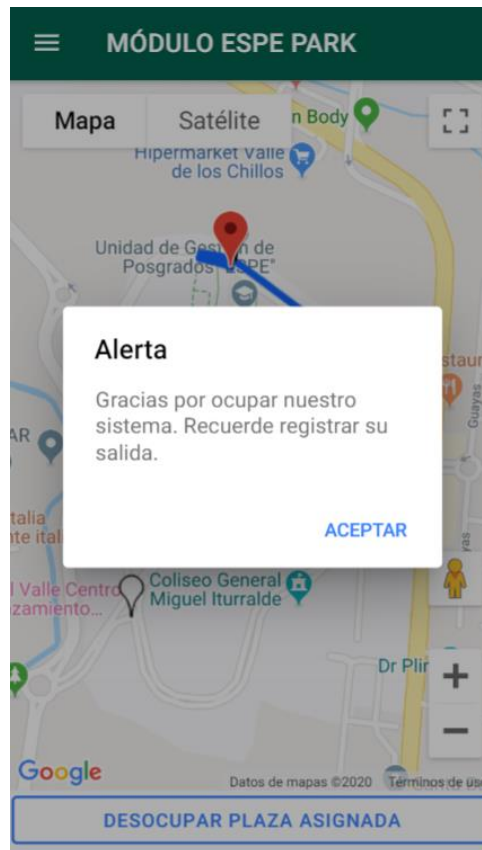
**Figura 34.** Ruta emitida por la aplicación para llegar a la plaza disponible de aparcamiento

12. Luego de que el usuario solicita una plaza disponible de aparcamiento, debe seguir la ruta graficada en la aplicación para llegar al objetivo que es la plaza disponible de aparcamiento.



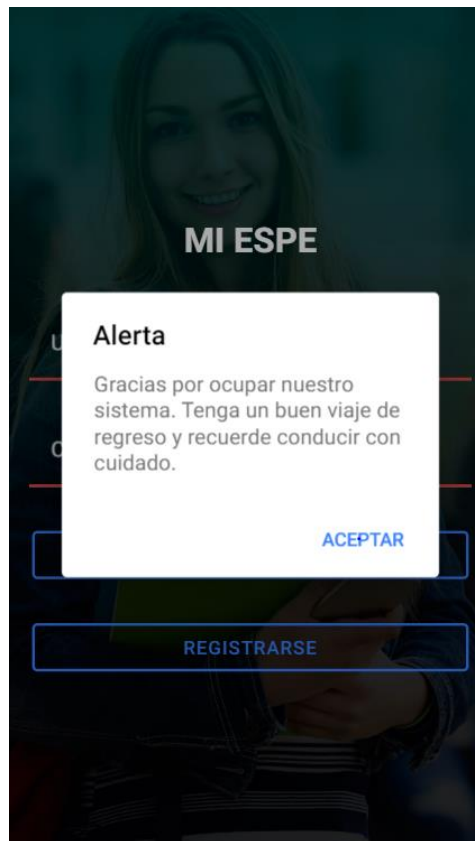
**Figura 35.** Usuario llega a la plaza de aparcamiento que fue asignada

13. Una vez que el usuario llegue a la plaza asignada debe dar clic en la opción “Ocupar Plaza Asignada” ya que así, dicha plaza se registrará como ocupada en el sistema y no podrá ser asignada a otra persona. Adicional, el sistema le recuerda que debe registrar la salida al momento de desocupar la plaza de aparcamiento, ya que si no registra la salida el sistema no le permitirá solicitar una plaza disponible de aparcamiento en otra ocasión.



**Figura 36.** El usuario confirma ocupa la plaza de aparcamiento asignada

14. Al momento de desocupar el área de aparcamiento asignada por la aplicación, el usuario debe registrar su salida, es decir debe seleccionar la opción “Desocupar Plaza Asignada”, para que dicha plaza se ponga en estado disponible para que otro usuario pueda ocuparla, al momento de abandonar la plaza, la aplicación móvil emite un mensaje como el que muestra la siguiente imagen.

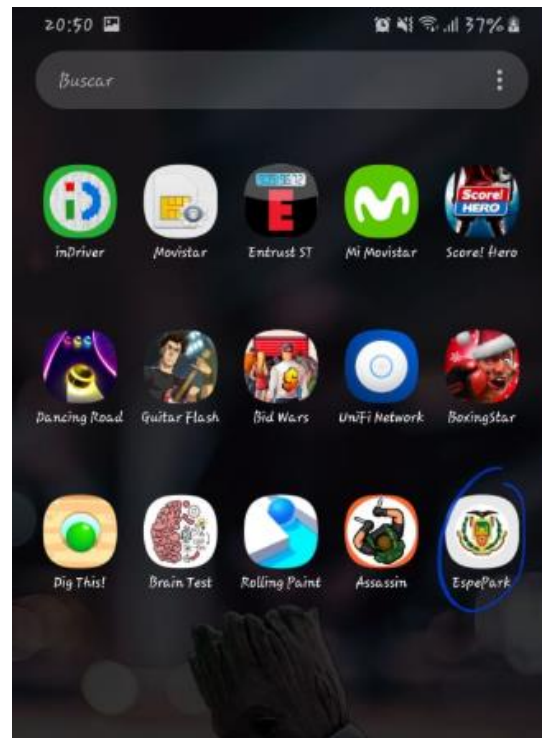


**Figura 37.** El usuario desocupa la plaza de aparcamiento asignada

#### **4.4. Ejecución de pruebas por parte de los usuarios del área de parqueo de vehículos que posee la Universidad ESPE**

Para la ejecución de pruebas de validación con los usuarios del área de parqueadero de la Universidad, el proceso de ejecución de pruebas fue similar al realizado por los integrantes del grupo. A continuación, se presenta varias imágenes de las pruebas ejecutadas con los usuarios del parqueadero.

- El usuario se descarga la aplicación móvil denominada EspePark, para empezar la sección de pruebas.



**Figura 38.** Icono de la aplicación móvil EspePark

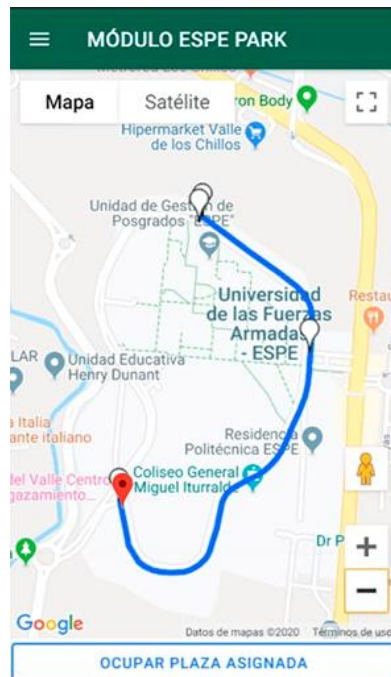
- Una vez realizado el proceso de registro de información, que es similar al proceso detallado en el punto anterior, el usuario solicita una plaza de parqueo para ocupar con su vehículo.





**Figura 39.** El usuario solicita una plaza de aparcamiento disponible

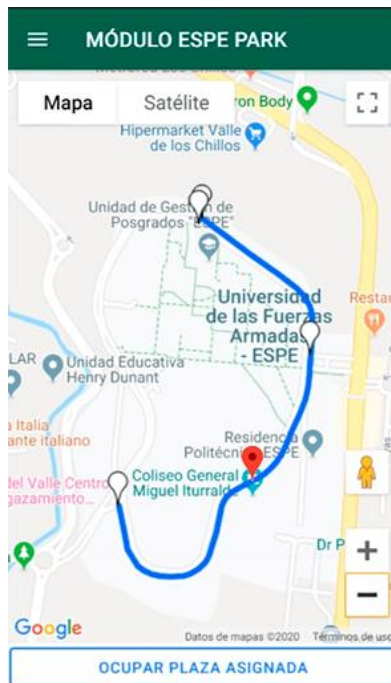
- Una vez que el usuario solicitó la plaza de aparcamiento, la aplicación asigna la plaza más cercana al lugar de trabajo, además grafica la ruta más corta que debe seguir para llegar al objetivo.



**Figura 40.** Gráfica de la ruta hacia la plaza asignada

- El usuario empieza a seguir la ruta en dirección a la plaza de aparcamiento asignada, para pruebas externas se realizó tres paradas estratégicas con el fin de recolectar capturas de cómo va avanzando en la ruta asignada. Una de las paradas realizadas, fue en el coliseo “Gral. Miguel Iturralde”, las siguientes capturas muestran la parada realizada y la ubicación exacta que la aplicación emite en tiempo real.



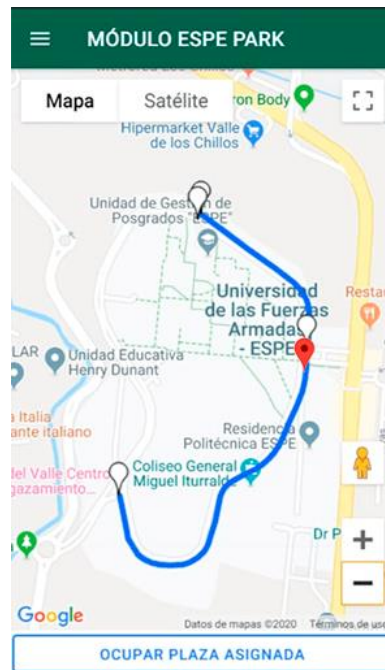


**Figura 41.** Ubicación de la aplicación en el coliseo "Gral. Iturralde"



**Figura 42.** Ubicación con el vehículo en el coliseo de la Universidad

- Otra de las paradas realizadas fue en la entrada principal de la Universidad, está para se realizó con el fin de tomar datos y capturas de la ubicación actual.

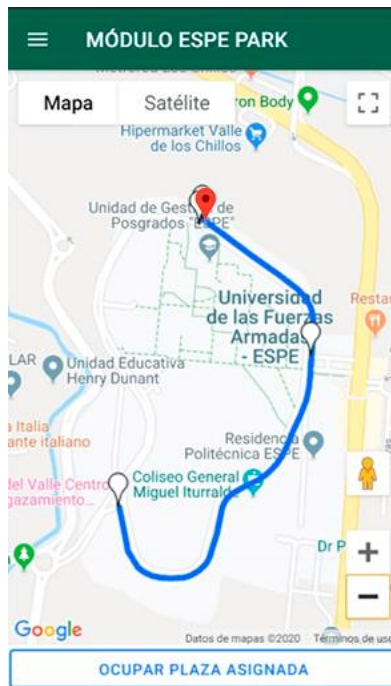


**Figura 43.** Ubicación de la aplicación en la entrada principal de la Universidad



**Figura 44.** Ubicación con el vehículo en la entrada principal de la Universidad

- Por último, el usuario llega a su punto destino, es decir, llega a la plaza de aparcamiento que fue asignada, una vez que llegó al lugar asignado, estacionó su automóvil.



**Figura 45.** Ubicación de la plaza de aparcamiento asignada



**Figura 46.** Usuario llega con el vehículo a la plaza de aparcamiento

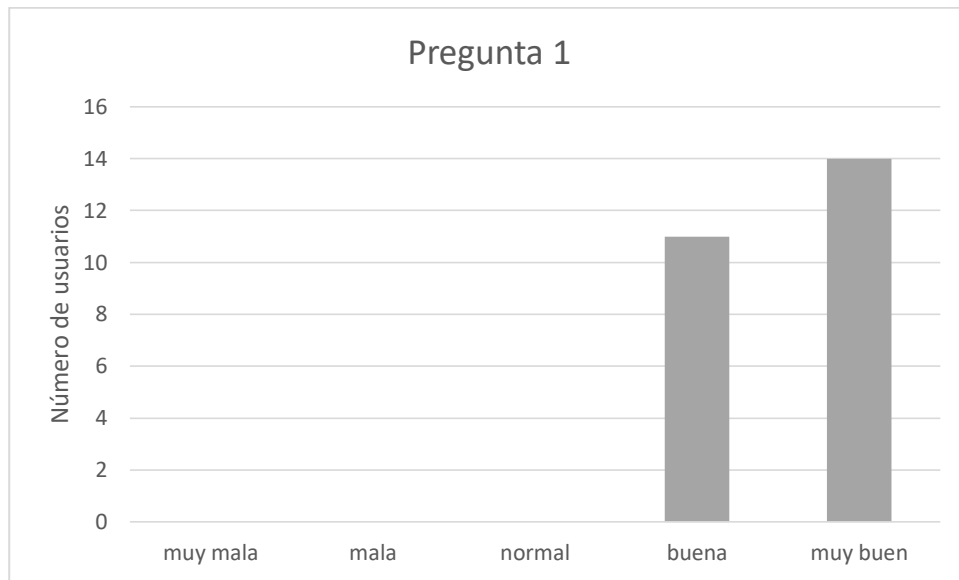
Una vez que el usuario llegó a la plaza de aparcamiento asignada, seleccionó la opción “Ocupar Plaza Asignada”, con esto la plaza de aparcamiento se mantiene ocupada hasta que el usuario la libere una vez que retire su automóvil de la plaza asignada.

#### **4.5. Resultados**

El total de usuarios que realizaron pruebas de la aplicación fue de 25 personas, estos usuarios fueron estudiantes del Departamento de Ciencias de la Computación (DCCO) que disponían un vehículo y acceso a los parqueaderos de la Universidad.

##### **4.5.1. Análisis de resultados sobre la satisfacción del usuario**

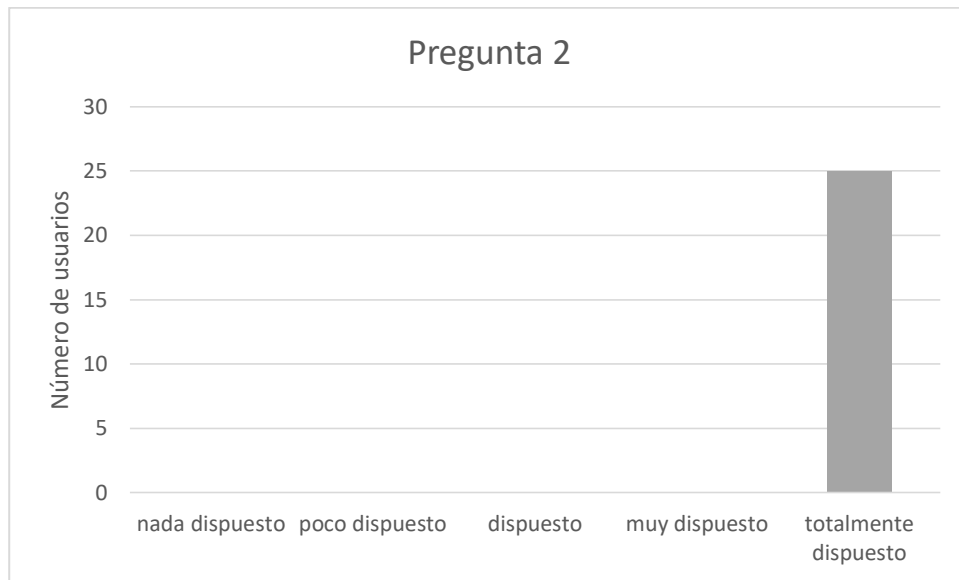
La iniciativa de crear una aplicación para la rápida detección de plazas libres de aparcamiento en la Universidad fue calificada como “muy buena” por la mayor parte de usuarios a los que se ejecutó el módulo de pruebas, la siguiente gráfica representa el número de personas que aceptaron esta iniciativa de informar en tiempo real sobre la disponibilidad de una plaza de aparcamiento.



**Figura 47.** Calificación de la iniciativa de crear la aplicación

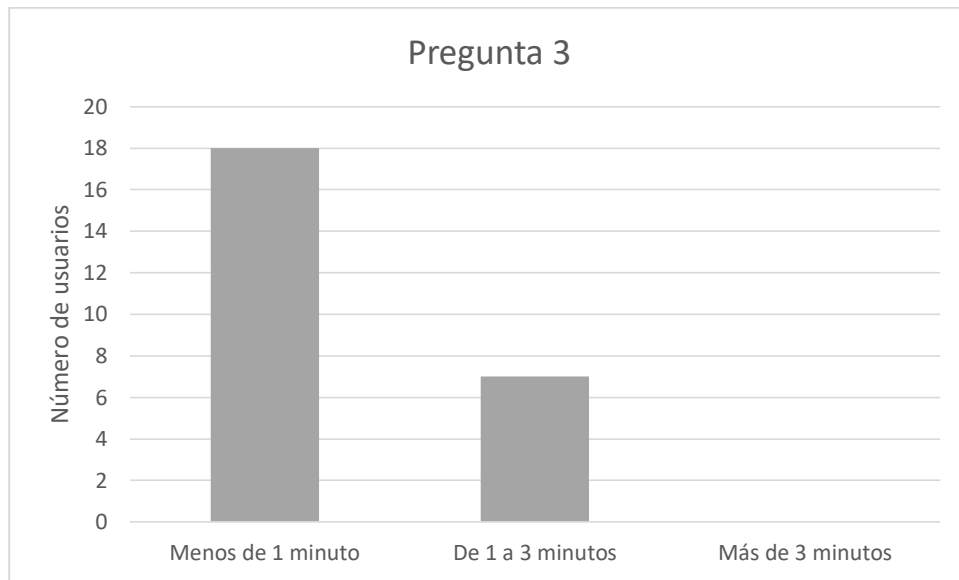
Uno de los comentarios más importantes que emitió un usuario en la fase de pruebas, fue que la aplicación ahorraría costos significativamente ya que al no emplear sensores para verificar la disponibilidad de una plaza de aparcamiento ya implica un ahorro considerable. Más de la mitad de los usuarios estuvo satisfecho con el funcionamiento y la eficacia del prototipo creado.

Los usuarios mostraron una predisposición para probar este nuevo prototipo creado, el objetivo era conocer sobre el producto y ver que facilidades y beneficios brinda en la búsqueda de una plaza libre para estacionar su vehículo, a continuación, se muestra en la gráfica el nivel de predisposición y colaboración que los usuarios mostraron para conocer esta aplicación.



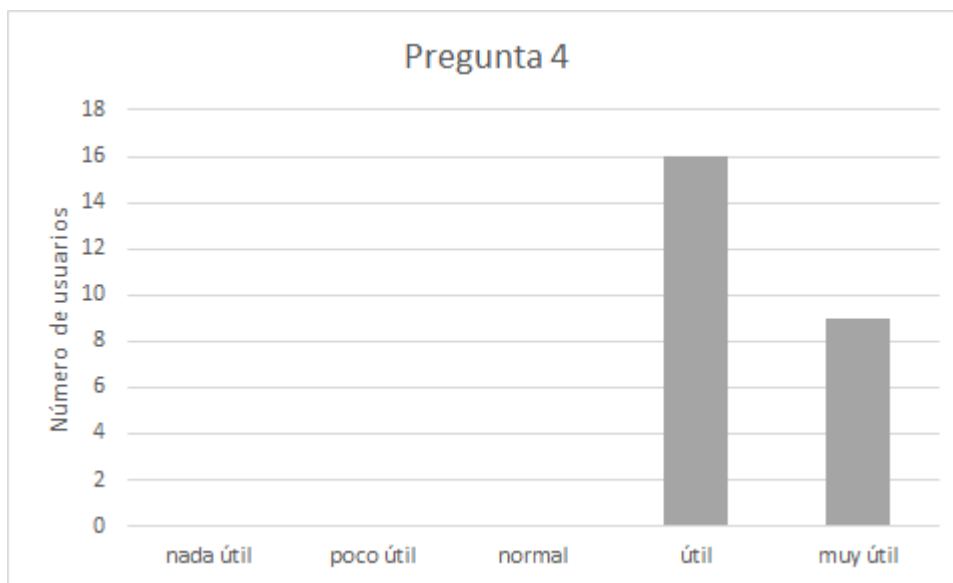
**Figura 48.** Número de usuarios dispuestos a probar la aplicación

En relación con el tiempo empleado para buscar un lugar libre de estacionamiento, los resultados de las encuestas realizadas arrojan que el tiempo empleado con el uso de la aplicación móvil es mucho menor al tiempo empleado normalmente. La mayoría de los usuarios empleó un tiempo menor a un minuto en buscar una plaza disponible, alrededor de siete personas emplearon más de un minuto para localizar una plaza disponible, esto se debe a factores como la conexión a la red de internet, tiempo de respuesta del teléfono.



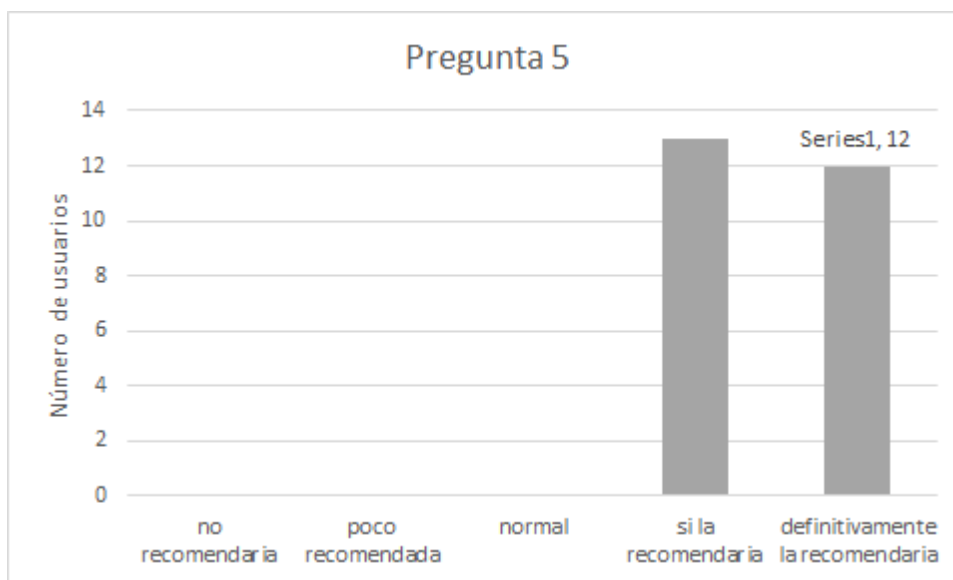
**Figura 49.** Tiempo empleado para la búsqueda de una plaza disponible por medio de la aplicación

Debido a las funcionalidades incorporadas en la aplicación desarrollada, los usuarios catalogaron como útil y muy útil a esta aplicación creada para la asignación de plazas libres para aparcar el automóvil, en la siguiente gráfica se puede observar los resultados sobre la utilidad de esta aplicación.



**Figura 50.** Nivel de utilidad de la aplicación móvil

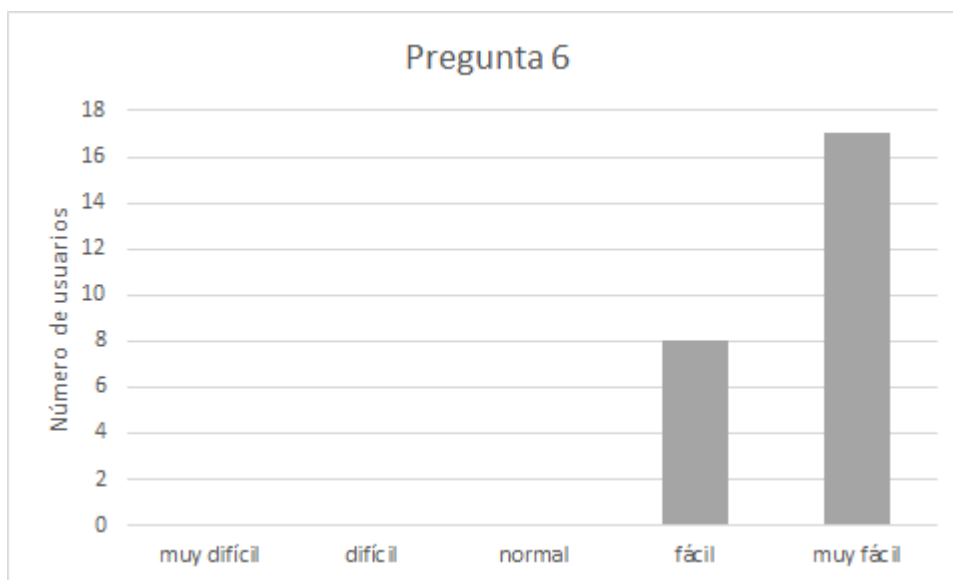
Una de las preguntas claves que fue formulada en la encuesta, habla sobre la recomendación de la aplicación a otras personas, es una pregunta clave ya que en base a esto se pudo definir el nivel de satisfacción de los usuarios al utilizar la aplicación.



**Figura 51.** Usuarios que recomendaría el uso de la aplicación

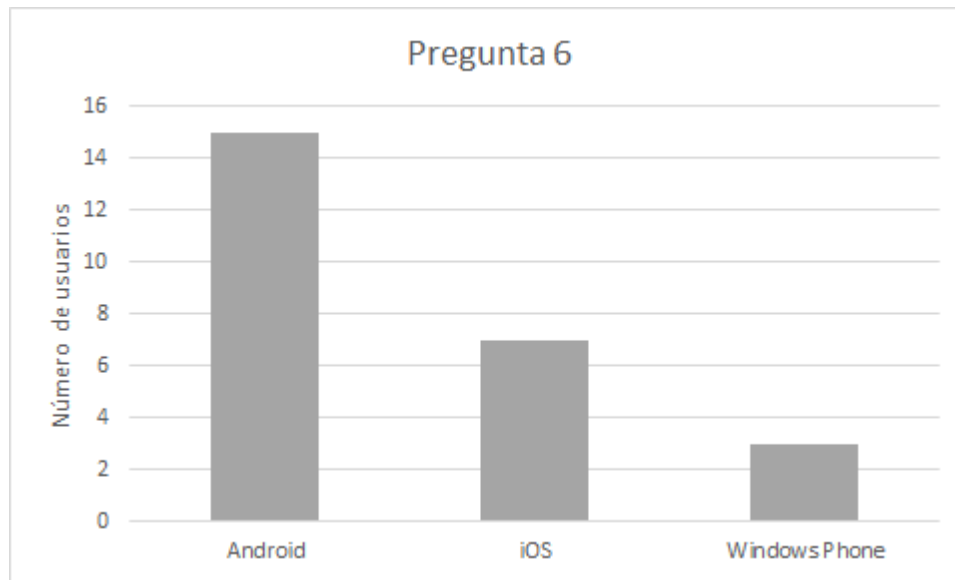


La aplicación fue construida con el objetivo de que no exista dificultad al manejar por cualquier usuario, razón por la cual se tomó en cuenta varios estándares de usabilidad, de manera que la navegación en la aplicación sea lo más sencilla posible, a continuación, se presenta los resultados de la dificultad que tiene la aplicación para navegar en ella, la mayoría de los usuarios respondió que fue muy fácil utilizar la aplicación.



**Figura 52.** Nivel de dificultad de la navegación en la aplicación móvil

Por último, para conocer la opinión de los usuarios y determinar cuál es el sistema operativo móvil más utilizado y el que presenta más beneficios, se planteó una pregunta que nos ayudará a determinar un posible trabajo a futuro, creando una aplicación multiplataforma e híbrida que funcione sin ningún problema en todos los dispositivos móviles.



**Figura 53.** Software móvil preferido por los usuarios de los parqueaderos de la Universidad

#### 4.5.2. Discusión de los resultados obtenidos en la fase de pruebas

En el numeral anterior se expusieron las pruebas realizadas a la aplicación desarrollada con los diferentes usuarios del parqueadero de la Universidad, estas pruebas se ejecutaron con diferentes usuarios de las áreas de aparcamiento. En el objetivo general se planteó el desarrollo de un sistema de información enfocado al ambiente móvil para la rápida detección de plazas disponibles de aparcamiento y al momento, la aplicación realizada recomienda un lugar disponible de aparcamiento, con lo cual el usuario ha empleado menos tiempo de lo que normalmente utilizaba para buscar una plaza disponible de aparcamiento vehicular.

Los resultados obtenidos en cada pregunta de la encuesta muestran un nivel alto de aceptación de la aplicación desarrollada, en el análisis de resultados también se pudo

determinar que el usuario actualmente prefiere ocupar el sistema operativo Android, ya que la mayoría de aplicaciones son gratis en dicho sistema operativo móvil y en los últimos años Android ha empezado a liderar el mercado tecnológico.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1. Introducción**

En el presente capítulo se presenta las conclusiones y recomendaciones del sistema de información para detectar plazas disponibles de aparcamiento vehicular, en este capítulo se expone varios aspectos acerca de la importancia y la gran ayuda que brindo la implementación de esta aplicación para la rápida detección de lugares disponibles de aparcamiento.

#### **5.2. Conclusiones**

En base a la revisión sistemática de literatura se determinó que, en la actualidad los sistemas de información enfocados a la determinación de plazas disponibles de aparcamiento vehicular integran herramientas hardware de tipo sensores, lo cual repercute en los costos elevados en el desarrollo de dichos sistemas.

La digitalización de las zonas de aparcamiento ubicadas en la parte posterior del DCCO se lo realizó mediante una herramienta de Google denominada "Sitios", la cual permite la creación y determinación de puntos específicos en el mapa guardando los datos de ubicación como latitud, longitud y elevación, datos que sirvieron para la comparación y verificación con los datos de geolocalización emitidos por el teléfono del usuario. Estos puntos representan los espacios de aparcamiento vehicular que fueron incluidos en la aplicación para el desarrollo del prototipo.

La integración de los componentes necesario para el funcionamiento adecuado del sistema de información fue realizada bajo el framework de desarrollo IONIC, empleando tecnologías orientadas a servicios web, las cuales facilitan el consumo de los recursos necesarios para el óptimo funcionamiento de la aplicación móvil.

En base a las pruebas ejecutadas se determina que el prototipo propuesto y desarrollado es totalmente funcional y válido para la rápida detección de plazas disponibles de aparcamiento vehicular, puesto que el tiempo empleado en la búsqueda mediante la aplicación es totalmente menor al que se emplea en una búsqueda normal.

Los resultados obtenidos de las encuestas aplicadas durante la fase de pruebas efectuada por los usuarios demuestran el cumplimiento de los objetivos propuestos obteniendo como resultado una aplicación móvil fácil de usar.

### **5.3. Recomendaciones**

Es recomendable trabajar en una sola versión de IONIC, esto facilitará el desarrollo de una aplicación estable, ya que las continuas actualizaciones que trae este framework incluyen nuevas librerías y actualizan la sintaxis de las ya existentes, generando errores e inconvenientes al momento de compilar el código.

Segmentar por zonas los espacios disponibles de aparcamiento en la Universidad, esto facilitará que trabajos futuros tengan identificado de mejor manera cada zona de parqueo, promoviendo de esta forma el uso de la aplicación móvil a todos los integrantes de la comunidad universitaria ESPE.

Trabajar con repositorios digitales con el fin de tener un control adecuado sobre las actualizaciones y configuraciones que se desarrollan en los diferentes módulos de la aplicación.

Al diseñar las interfaces de la aplicación se recomienda seguir normas de usabilidad, esto facilitará la navegación dentro de la aplicación, generando una reacción positiva en el usuario.

Es recomendable emplear gráficos en cada interfaz, así el usuario tendrá facilidad para ubicarse y navegar de mejor manera en la aplicación móvil.

#### **5.4. Trabajo Futuro**

Se propone para trabajos futuros, la digitalización de los diferentes sectores de la Universidad de las Fuerzas Armadas ESPE y de los lugares de aparcamiento que están asociados al sector digitalizado, así como del desarrollo de la lógica utilizada para el sector del Departamento de Ciencias de la Computación (DCCO), para lo cual se debe generar los archivos KML de las nuevas rutas que se va a emplear en los diferentes sectores, esto permitirá abarcar a más integrantes de la comunidad Universitaria ESPE, facilitando de mejor manera el aparcamiento de vehículos.

Otra línea para un trabajo futuro en base a los resultados obtenidos, se plantea la exportación de la aplicación a la plataforma iOS, lo cual permitirá incrementar el número de usuarios para la aplicación, adicionalmente se plantea la integración con el sistema de datos de la Universidad de las Fuerzas Armadas ESPE, lo que permitirá optimizar procesos dentro del sistema de información desarrollado en el presente trabajo de

titulación, brindando un servicio completo al usuario mejorando considerablemente la calidad de experiencia al reducir procesos manuales internos del sistema de información.

## BIBLIOGRAFÍA

- Calero, C. (2010). Calidad del producto y proceso software. Albacete.
- Duarte, G. (2016). Arquitectura Propuesta para un servicio web completo: Metodología de desarrollo e implementación. Bogotá: Universidad Distrital Francisco José de Caldas.
- Escalante, L. (2016). El patrón de arquitectura n-capas con orientación al dominio como solución en el diseño de aplicaciones empresariales.
- Fredrich, T. (2012). RESTful Service Best Practices. Recommendations for creating web services. Pearson eCollege.
- Lizón, R. (2013). Guía para el desarrollo de aplicaciones GSM basadas en el Sistema de Señalización N°7. Valencia: ETSIT.
- Melnichuk, M., Kornienko, Y., & Boytsova, O. (2018). Web-services. Restful architecture. Automation of Technological and Business Processes. Gran Bretaña: Journals.
- Richardson, L., & Ruby, S. (2007). Restful Web Services. United States of America: Safari.
- Rodriguez, E. (2010). La Geolocalización, coordenadas hacia el éxito. Salamanca.
- Rodríguez, M. (2016). Metodología para describir servicios Restful consumidos automáticamente por clientes máquina. Antioquia: unir.
- Rohlf, J., & McCledon, B. (1 de Abril de 2008). Patente US7353114 - Markup language for an interactive geographic information system: Google Patentes. Obtenido de <http://www.google.com/patents/US7353114>
- Schiaffarino, A. (12 de Marzo de 2019). infranetworking. Obtenido de <https://blog.infranetworking.com/modelo-cliente-servidor/>
- Vázquez, J. (2010). Fundamentos del sistema GPS. Formación Topografía.
- Aponte Gomez Carlos Davila Ramirez Universidad Ean Facultad Ingenieria De Sistemas, S. DE. (2011). Sistemas operativos móviles: funcionalidades, efectividad y aplicaciones utiles en colombia.
- AWS. (2019). Bases de datos no relacionales | Bases de datos de gráficos | AWS. Retrieved November 9, 2019, from [https://aws.amazon.com/es/nosql/?trk=ps\\_a131L0000057W46QAE&trkCampaign=Emerging\\_Markets\\_Search&sc\\_channel=ps&sc\\_campaign=emp\\_EC&sc\\_publisher=google&sc\\_category=Database&sc\\_country=EC&sc\\_geo=LATAM&sc\\_outcome=a cq&sc\\_detail=%2Bbase de %2Bdatos %2Bnosql&sc\\_co](https://aws.amazon.com/es/nosql/?trk=ps_a131L0000057W46QAE&trkCampaign=Emerging_Markets_Search&sc_channel=ps&sc_campaign=emp_EC&sc_publisher=google&sc_category=Database&sc_country=EC&sc_geo=LATAM&sc_outcome=a cq&sc_detail=%2Bbase de %2Bdatos %2Bnosql&sc_co)



- Baz Alonso, A., Ferreira Artime, I., Álvarez Rodríguez, M., & García Baniello EPSIG, R. (2011). Dispositivos móviles.
- Bazo, O. (2019). GitHub - uokesita/the-little-mongodb-book: The Little MongoDB Book. Retrieved November 13, 2019, from <https://github.com/uokesita/the-little-mongodb-book>
- Beltrán López, G. (2015). La geolocalización social. Polígonos. Revista de Geografía, 27(27), 97. <https://doi.org/10.18002/pol.v0i27.3290>
- Bonilla Guillermo Santiago, E., & Guzmán Alex Israel, C. (n.d.). . Introducción investigación y desarrollo de la Fuerza Aérea Ecuatoriana.
- Borruso, G., & Defend, V. (2016). Mapping a City's activity. A project of volunteered geographic information using mobile mapping collection. Lecture Notes in Geoinformation and Cartography, 391–411. [https://doi.org/10.1007/978-3-319-19602-2\\_24](https://doi.org/10.1007/978-3-319-19602-2_24)
- De La, A., Cerón, C., Carlos, R., Fernando, M., & Moscol, R. (2017). Facultad de Ingeniería, Arquitectura y Urbanismo.
- Developers, A. (2019a). Arquitectura de la plataforma. Retrieved January 4, 2020, from <https://developer.android.com/guide/platform?hl=es>
- Developers, A. (2019b). Introducción a Android Studio. Retrieved January 4, 2020, from <https://developer.android.com/studio/intro>
- Dickson, J. (2013). Xamarin Mobile Development. Retrieved from <http://scholarworks.gvsu.edu/cistechlibhttp://scholarworks.gvsu.edu/cistechlib/167>
- Diplomado de Gobierno Abierto y Participativo Institucional Mayo-Agosto 2005 Módulo 2 Metodología para llevar a cabo una encuesta. (n.d.). Retrieved from [http://www.worldbank.org/wbi/governance/pdf/guide\\_pdfs/05g-](http://www.worldbank.org/wbi/governance/pdf/guide_pdfs/05g-)
- Du, Y. J., Yu, C. C., & Liu, J. (2009). A Study of GIS development based on KML and Google Earth. NCM 2009 - 5th International Joint Conference on INC, IMS, and IDC, 1581–1585. <https://doi.org/10.1109/NCM.2009.17>
- Fajardo, R., Mercado, J., Morales, Z., Sánchez, O., & Santillán, C. (2018). Plan de negocio para un sistema de intermediación de servicios de belleza en la ciudad de Lima basado en la geolocalización.
- GCFGlobal. (2019). Informática Básica: Sistemas operativos para dispositivos móviles. Retrieved November 14, 2019, from <https://edu.gcfglobal.org/es/informatica-basica/sistemas-operativos-para-dispositivos-moviles/1/>
- Gonzalez, A. N. (2011). ¿Qué es Android? Xataka Android.
- Haid, E., Kiechle, G., Göll, N., & Soutschek, M. (2008). Evaluation of a Web-based and Mobile Ski Touring Application for GPS-enabled Smartphones. In Information and Communication Technologies in Tourism 2008 (pp. 313–323).

[https://doi.org/10.1007/978-3-211-77280-5\\_28](https://doi.org/10.1007/978-3-211-77280-5_28)

Ibarra, A., Esteban, J., Maldonado, M., & Mariel, T. (2015). Universidad de las Fuerzas Armadas ESPE ”.

Ingeniería, F. DE, Actividades De La Unach, N. DE, & Daniel Santos Galán Tutor Ing Geonatan Octavio Peñafiel Barros, A. (2018). Universidad Nacional de Chimborazo "Análisis comparativo de los IDE'S de desarrollo Xamarin Visual Studio vs Android Studio aplicado al sistema de.

Ingeniería En Sistemas Y Computación, C. DE, Luis Duchi Quishpe, J., & Lorena Paulina Molina Valdiviezo, I. (2019). Universidad Nacional de Chimborazo Facultad de Ingeniería.

Ionic. (2019). What is Ionic Framework? - Ionic Documentation. Retrieved November 13, 2019, from <https://ionicframework.com/docs/intro>

Javier Martin, J. (2019). MongoDB - EcuRed. Retrieved November 12, 2019, from <https://www.ecured.cu/MongoDB>

Juliana, Y., Rivera, M., Sandoval, J., Santiago, C., & Franco, A. T. (2012). Sistema operativo Android: Características y funcionalidad para dispositivos móviles Universidad Tecnológica de Pereira Facultad de Ingenierías: Eléctrica, Electrónica, Física y Ciencias de la Computación Programa: Ingeniería de Sistemas y Computación Pereira.

K, A. (2015). Cómo hacer una encuesta paso a paso | CreceNegocios. Retrieved December 22, 2019, from <https://www.crecenegocios.com/como-hacer-una-encuesta-paso-a-paso/>

López, B. (2016). Geolocalización online: La importancia del dónde - Beltrán López, Gersón - Google Libros. Retrieved November 20, 2019, from [https://books.google.com.ec/books?hl=es&lr=&id=5FLedQAAQBAJ&oi=fnd&pg=PT4&dq=geolocalización+conceptos&ots=AR0he5IIEN&sig=cCFKrEmvh31EUhxSluVx2zM0enk&redir\\_esc=y#v=onepage&q=capitulo III&f=false](https://books.google.com.ec/books?hl=es&lr=&id=5FLedQAAQBAJ&oi=fnd&pg=PT4&dq=geolocalización+conceptos&ots=AR0he5IIEN&sig=cCFKrEmvh31EUhxSluVx2zM0enk&redir_esc=y#v=onepage&q=capitulo III&f=false)

López Michelone, M. (2013). La historia de Android. Retrieved November 10, 2019, from <https://www.unocero.com/noticias/la-historia-de-android/>

Marés, L. (2012). Tablets en educación. Retrieved from <http://www.onlineuniversities.com/18-enlightening-ipad-experiments-in->

Martín, A., Chávez, S., Rodríguez, N., & Valenzuela, A. (2013). Bases de Datos NoSql en Cloud Computing.

Oracle. (2018). ¿Qué es J2ME o Java ME? Retrieved November 10, 2019, from [https://www.java.com/es/download/faq/whatis\\_j2me.xml](https://www.java.com/es/download/faq/whatis_j2me.xml)

P, G. P., Hamsini, R., & Smitha, G. R. (2016). Agile Development Methodology and Testing for Mobile Applications - A Survey. International Journal of New Technology

- and Research (IJNTR), 2(9), 98–101.
- Ruben, F. (2014). MongoDB. Qué es, cómo funciona y cuándo podemos usarlo (o no). Retrieved November 12, 2019, from <https://www.genbeta.com/desarrollo/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no>
- Santos, G., & Dpto, D. (2015). Proyecto de innovación y mejora docente : “ Prácticas de topografía con Google Earth ” ( ID2014 / 0074 ).
- Software, E. (2017). Qué es la geolocalización y cómo funciona - Evaluando Software. Retrieved November 20, 2019, from <https://www.evaluandosoftware.com/la-geolocalizacion-funciona/>
- Stanton, W., Etzel, M., & Walker, B. (2007). Fundamentos de Marketing.
- Thompson, I. (2006). Tipos de encuesta - Promonegocios.net. Retrieved December 8, 2019, from <https://www.promonegocios.net/mercadotecnia/encuestas-tipos.html>
- Tomoyose, G. (2011). Los celulares básicos resisten al embate de los smartphones - LA NACION. Retrieved November 10, 2019, from <https://www.lanacion.com.ar/tecnologia/la-vigencia-de-los-celulares-basicos-nid1361758>
- Tumipampa Borja, E. E. (2016). Desarrollo De Una Aplicación Móvil Que Permite A Los Docentes Y Estudiantes De La Universidad Central Del Ecuador Acceder A Las Bases De Datos Científicas. Universidad Central Del Ecuador, 63. <https://doi.org/https://doi.org/10.3929/ethz-b-000238666>
- Vico, Á. J. (2011). Arquitectura de Android. Retrieved November 10, 2019, from <https://columna80.wordpress.com/2011/02/17/arquitectura-de-android/>

# ANEXOS