Pontificia Universidad Javeriana Cali

**DETECCIÓN Y VISUALIZACIÓN DE ESPACIOS DE PARQUEO POR MEDIO DE UN APLICATIVO MÓVIL HACIENDO USO DE SENSORES**

**FABIO ALESSANDRO GATTI LERMA**

**IVÁN FERNANDO RAMÍREZ SANCLEMENTE**

**PONTIFICIA UNIVERSIDAD JAVERIANA CALI**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**SANTIAGO DE CALI - MAYO DE 2018**

DETECCIÓN Y VISUALIZACIÓN DE ESPACIOS DE PARQUEO POR MEDIO DE UN APLICATIVO MÓVIL HACIENDO USO DE SENSORES

FABIO ALESSANDRO GATTI LERMA

IVÁN FERNANDO RAMÍREZ SANCLEMENTE

TRABAJO DE GRADO

DIRECTOR

JUAN PABLO GARCÍA CIFUENTES

INGENIERO DE SISTEMAS

PONTIFICIA UNIVERSIDAD JAVERIANA CALI

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

SANTIAGO DE CALI - MAYO DE 2018

Santiago de Cali, mayo 10 de 2018.

Señores

**Pontificia Universidad Javeriana Cali.**

Dr. Andrés Adolfo Navarro Newball

Director Carrera Ingeniería de Sistemas y Computación.

Cali.

Cordial saludo.

Por medio de la presente hago constar que en calidad de director he revisado el proyecto de grado titulado “Detección y visualización de espacios de parqueo haciendo uso de aplicaciones móviles, sensores de proximidad y almacenamiento en la nube”, de los estudiantes Fabio Alessandro Gatti Lerma (Cod: 71009) e Iván Fernando Ramírez Sanclemente (Cod: 70882); el cual cumple con los requisitos para ser sometido a consideración del jurado.

Atentamente,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Juan Pablo García Cifuentes

Santiago de Cali, mayo 10 de 2018.

Señores

**Pontificia Universidad Javeriana Cali.**

Andrés Adolfo Navarro Newball

Director Carrera Ingeniería de Sistemas y Computación.

Cali.

Cordial saludo.

Permitimos a su consideración el anteproyecto de grado titulado “Detección y visualización de espacios de parqueo haciendo uso de aplicaciones móviles, sensores de proximidad y almacenamiento en la nube” con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el proyecto de grado y posteriormente optar al título de Ingenieros de Sistemas y Computación.

Al firmar aquí, damos fe que entendemos y conocemos las directrices para la presentación de trabajos de grado de la Facultad de Ingeniería aprobadas el 26 de noviembre de 2009, donde se establecen los plazos y normas para el desarrollo del anteproyecto y del trabajo de grado.

Atentamente,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fabio Alessandro Gatti Lerma Iván Fernando Ramírez Sanclemente

Código: 71009 Código: 70882

**AGRADECIMIENTOS**

Queremos hacer un reconocimiento a nuestras familias, a nuestra institución y a nuestros docentes, los cuales nos han acompañado durante todo nuestro proceso formativo y educativo.

Finalmente, elevamos un sincero agradecimiento a nuestro director de tesis, cuya   
colaboración y presencia oportunas permitieron desarrollar y concluir esta investigación.

A todos ustedes, mil gracias.

CONTENIDO

**Pág.**

[RESUMEN 12](#_Toc519005007)

[INTRODUCCIÓN 13](#_Toc519005008)

[1. DEFINICIÓN DEL PROBLEMA 15](#_Toc519005009)

[1.1. Planteamiento del problema 15](#_Toc519005010)

[1.2. Benchmarking 15](#_Toc519005011)

[1.2.1. Soluciones nacionales 16](#_Toc519005012)

[1.2.1.1. Eco Parking 16](#_Toc519005013)

[1.2.1.2. ParkEyes 17](#_Toc519005014)

[1.2.1.3. Centro comercial Chipichape 18](#_Toc519005015)

[1.2.2. Soluciones internacionales 19](#_Toc519005016)

[1.2.2.1. Parkifi 19](#_Toc519005017)

[1.2.2.2. Rover Parking 20](#_Toc519005018)

[1.2.2.3. Park Smart 21](#_Toc519005019)

[1.2.3. Comparación y conclusiones 22](#_Toc519005020)

[2. OBJETIVOS 25](#_Toc519005021)

[2.1. Objetivo general 25](#_Toc519005022)

[2.2. Objetivos específicos 25](#_Toc519005023)

[3. MARCO TEÓRICO 26](#_Toc519005024)

[3.1. An embedded sensing approach to monitoring parking lot occupancy (*Un enfoque de detección embebido para monitorear la ocupación del estacionamiento)* 26](#_Toc519005025)

[3.2. Building sustainable parking lots with the Web of Things *(Construyendo estacionamientos sostenibles usando el Internet de las Cosas)* 26](#_Toc519005026)

[3.3. Applying mobile and internet of things technologies in managing parking spaces for people with disabilities *(Aplicando tecnologías móviles e internet de las cosas al manejo de espacios de parqueo para personas con discapacidades)* 27](#_Toc519005027)

[3.4. Wireless sensor networks and service-oriented architecture, as suitable approaches to be applied into ITS *(Redes de sensores inalambricos y arquitectura orientada a servicios como enfoques adecuados para aplicarse en los SIT)* 27](#_Toc519005028)

[3.5. Low Cost Internet of Things based Vehicle Parking Information System *(Sistema de información de parqueo basado en Internet de las Cosas de bajo costo)* 28](#_Toc519005029)

[3.6. Car parking management system using AMR-sensor technology *(Sistema de administración de parqueadero usando sensores magnéticos)* 28](#_Toc519005030)

[3.7. Real time car parking system using image processing *(Sistema de parqueadero en tiempo real usando procesamiento de imágenes)* 29](#_Toc519005031)

[3.8. iParking: An intelligent parking system for large parking lots *(Iparking: Un Sistema de paqueo inteligente para grandes establecimientos)* 29](#_Toc519005032)

[3.9. Performance Analysis of Proximity and Light Sensors for Smart Parking *(Análisis de desempeño para sensores de proximidad y luz en parqueaderos inteligentes)* 30](#_Toc519005033)

[3.10. Automatic Smart Parking System using Internet of Things (IOT) *(Sistema de parqueo automático inteligente usando el Internet de las Cosas)* 30](#_Toc519005034)

[4. DESARROLLO 31](#_Toc519005035)

[4.1. Metodologías 31](#_Toc519005036)

[4.1.1. Design Thinking 31](#_Toc519005037)

[4.1.1.1. Empatizar 33](#_Toc519005038)

[4.1.1.2. Definir 33](#_Toc519005039)

[4.1.1.3. Prototipar 34](#_Toc519005040)

[4.1.1.4. Testear 34](#_Toc519005041)

[4.1.2. SCRUM 35](#_Toc519005042)

[4.1.2.1. Roles en Scrum: ¿Quién? y ¿Qué? 35](#_Toc519005043)

[4.1.2.2. El Sprint: ¿Dónde? ¿Cuándo? 36](#_Toc519005044)

[4.2. Levantamiento de requerimientos 38](#_Toc519005045)

[4.2.1. Entrevistas a los usuarios 38](#_Toc519005046)

[4.2.2. Necesidades de los usuarios 39](#_Toc519005047)

[4.2.3. Historias de usuario 40](#_Toc519005048)

[4.2.4. Diagramas de secuencia 45](#_Toc519005049)

[4.3. Prototipo inicial 51](#_Toc519005050)

[4.4. Prototipo funcional / Prototipo final 52](#_Toc519005051)

[4.4.1. Tecnologías 52](#_Toc519005052)

[4.4.1.1. Back-end 54](#_Toc519005053)

[4.4.1.2. Front-end 57](#_Toc519005054)

[4.4.1.3. Hardware 58](#_Toc519005055)

[4.5. Modelo de negocio 59](#_Toc519005056)

[4.6. Marketing 59](#_Toc519005057)

[5. COSTOS Y PROYECCIÓN ECONÓMICA 62](#_Toc519005058)

[6. RESULTADOS OBTENIDOS 64](#_Toc519005059)

[6.1. Pruebas funcionales 64](#_Toc519005060)

[6.2. Pruebas de usuario 82](#_Toc519005061)

[6.3. Pruebas del dispositivo de detección 84](#_Toc519005062)

[7. CONCLUSIONES 85](#_Toc519005063)

[8. TRABAJOS FUTUROS 86](#_Toc519005064)

[9. CRONOGRAMA DE ACTIVIDADES 87](#_Toc519005065)

[10. BIBLIOGRAFÍA Y REFERENCIAS 89](#_Toc519005066)

[11. ANEXOS 91](#_Toc519005067)

[11.1. Preguntas de las entrevistas 91](#_Toc519005068)

[11.2. Grabaciones de las entrevistas 92](#_Toc519005069)

[11.3. Grabaciones de pruebas de usuario 93](#_Toc519005070)

[11.4. Base de datos JSON 93](#_Toc519005071)

LISTA DE TABLAS

**Pág.**

[Tabla 1. Comparación entre las soluciones 24](#_Toc519004719)

[Tabla 2. Costos para el desarrollo del proyecto 62](#_Toc519004720)

[Tabla 3. Costos por diseño de sistema de detección 63](#_Toc519004721)

[Tabla 4. Consumo de energía del dispositivo 84](#_Toc519004722)

**LISTA DE FIGURAS**

**Pág.**

[Figura 1. Eco Parking y su solución 16](#_Toc519004641)

[Figura 2. ParkEyes y su solución 17](#_Toc519004642)

[Figura 3. Chipichape y su solución 18](#_Toc519004643)

[Figura 4. Parkifi y su solución 19](#_Toc519004644)

[Figura 5. Rove Parking y su solución 20](#_Toc519004645)

[Figura 6. Park Smart y su solución 21](#_Toc519004646)

[Figura 7. Etapas de Design Thinking 32](#_Toc519004647)

[Figura 8. Refinamiento de una solución 32](#_Toc519004648)

[Figura 9. Diagrama de inicio de sesión 45](#_Toc519004649)

[Figura 10. Diagrama de cierre de sesión 46](#_Toc519004650)

[Figura 11. Diagrama de registro de usuario 47](#_Toc519004651)

[Figura 12. Diagrama de consulta de establecimiento 48](#_Toc519004652)

[Figura 13. Diagrama de navegación de menú 49](#_Toc519004653)

[Figura 14. Diagrama de administración de perfil 50](#_Toc519004654)

[Figura 15. Diagrama de administración de configuración 50](#_Toc519004655)

[Figura 16. Prototipo inicial 51](#_Toc519004656)

[Figura 17. Arquitectura del proyecto 53](#_Toc519004657)

[Figura 18. Primer diseño de la base de datos 55](#_Toc519004658)

[Figura 19. Diseño final de la base de datos 56](#_Toc519004659)

[Figura 20. Volante promocional 60](#_Toc519004660)

[Figura 21. Poster promocional 61](#_Toc519004661)

[Figura 22. Registro de usuario en el aplicativo 65](#_Toc519004662)

[Figura 23. Registro de usuario ya creado 66](#_Toc519004663)

[Figura 24. Inicio de sesión 67](#_Toc519004664)

[Figura 25. Inicio de sesión con un usuario no registrado 68](#_Toc519004665)

[Figura 26. Iniciar sesión con contraseña inválida 69](#_Toc519004666)

[Figura 27. Restablecer contraseña 71](#_Toc519004667)

[Figura 28. Consultar establecimiento por nombre 72](#_Toc519004668)

[Figura 29. Consultar establecimiento seleccionándolo 73](#_Toc519004669)

[Figura 30. Funcionalidad de micrófono 74](#_Toc519004670)

[Figura 31. Ruta desde la posición del usuario 75](#_Toc519004671)

[Figura 32. Cambio de estado del parqueadero a ocupado 76](#_Toc519004672)

[Figura 33. Cambio de estado del parqueadero a desocupado 77](#_Toc519004673)

[Figura 34. Modificar información 78](#_Toc519004674)

[Figura 35. Cambiar el tema del aplicativo 79](#_Toc519004675)

[Figura 36. Cambiar el idioma del aplicativo 80](#_Toc519004676)

[Figura 37. Cambiar el tipo de mapa 81](#_Toc519004677)

[Figura 38. Pruebas de usuario 83](#_Toc519004678)

**LISTA DE ANEXOS**

**Pág.**

[Anexo 1. Preguntas de las entrevistas 91](file:///D:\GitHub\ParkMe_API\Proyecto%20de%20Grado.docx#_Toc519004621)

[Anexo 2. Grabaciones de las entrevistas 92](file:///D:\GitHub\ParkMe_API\Proyecto%20de%20Grado.docx#_Toc519004622)

[Anexo 3. Grabaciones de pruebas de usuario 93](file:///D:\GitHub\ParkMe_API\Proyecto%20de%20Grado.docx#_Toc519004623)

[Anexo 4. Base de datos JSON 93](file:///D:\GitHub\ParkMe_API\Proyecto%20de%20Grado.docx#_Toc519004624)

# 

# RESUMEN

Varios establecimientos comerciales y educativos han tratado de dar solución a las problemáticas que se generan dentro de sus parqueaderos, ya que estos no dan abasto para el creciente número de vehículos que ingresan al mismo, especialmente en horarios de alto tráfico vehicular para el establecimiento (temprano en la mañana para las universidades o los fines de semana para los centros comerciales).

En el siguiente documento se presenta una nueva propuesta como posible solución a una de las problemáticas mencionadas anteriormente, basándose en los resultados obtenidos por ideas que han sido aplicadas en diferentes establecimientos en Colombia y algunos países del mundo. Este documento nació con el fin de resolver la siguiente pregunta fundamental: ¿Cuál es la forma más eficiente de encontrar un parqueadero disponible dentro de un establecimiento?

Gracias a los avances tecnológicos de la actualidad, una gran mayoría de la población cuenta con un teléfono inteligente, razón por la cual se optó por llevar a cabo el desarrollo de un aplicativo móvil, con la opción de desarrollarse como aplicativo web, que dé a conocer a los usuarios el estado de los espacios de un parqueadero en tiempo real.

**Palabras Clave:** Aplicativo móvil, Internet de las cosas, Innovación, Base de datos en la nube, Cloud Computing.

# INTRODUCCIÓN

Conforme pasan los años, el número de vehículos dentro de la ciudad aumentan y, con esto, aumentan también los problemas de movilidad. Esto no solo se ve reflejado en las ya congestionadas vías de las ciudades, sino que además crean problemáticas dentro de los establecimientos, los cuales muchas veces no cuentan con la infraestructura o el capital suficiente para ampliar sus instalaciones.

Centrando este estudio en la Pontificia Universidad Javeriana Cali, es evidente, para las personas que asisten a este establecimiento, que los lugares de estacionamiento no son suficientes para la gran demanda que hay por parte de la comunidad estudiantil, el profesorado y demás usuarios.

Esta problemática por sí sola genera otras que ponen en riesgo tanto a los usuarios como al establecimiento, dado que algunas personas optan por estacionar sus vehículos en lugares no aptos, obstruyendo así las vías por las que se puede transitar o dañando, de manera usual, otros vehículos o propiedad de la universidad.

No obstante, aquellas personas que eligen evadir estas alternativas se enfrentan al hecho de que pueden tardar más tiempo buscando un espacio de estacionamiento libre sin ningún éxito garantizado, lo cual genera que, en algunos casos, se vean obligados a aparcar en establecimientos ajenos a la universidad. Todo esto y más generan un gasto innecesario de dinero debido al consumo de gasolina y de tiempo por parte de los usuarios, además del estrés que le pueda provocar a los mismos y el impacto ambiental que puede tener el uso del vehículo durante la búsqueda de un estacionamiento disponible.

Debido a lo anterior, se ha planteado una solución tecnológica que permita a los usuarios conocer el estado de los parqueaderos en tiempo real desde cualquier dispositivo móvil conectado a internet, ayudando a aliviar algunas de las problemáticas y el caos que se presentan en este tipo de establecimientos a ciertas horas del día.

Para la fase de investigación y levantamiento de información, se hizo uso de la metodología Design Thinking, que proporcionó las herramientas necesarias para recopilar información de interés, por medio de entrevistas realizadas a 7 usuarios finales y poder dar respuesta a la problemática de forma innovadora. En cuanto a las fases de diseño y desarrollo, se hizo uso de la metodología de desarrollo de software SCRUM, ya que, además de ser una metodología ágil conocida por centrarse en sus iteraciones y relación con el usuario, permite ir generando entregables funcionales con cada iteración, lo cual va en perfecta concordancia con la metodología de investigación. En la última fase, se realizaron pruebas funcionales de campo, integrando el aplicativo con los sensores y la base de datos, para así poder disminuir la probabilidad de fallos y obtener datos más reales y relevantes.

A continuación, se detalla la descripción del problema, la necesidad de este tipo de plataformas tecnológicas en las empresas, además de las fases de diseño y desarrollo de la solución que se decidió implementar.

# 1. DEFINICIÓN DEL PROBLEMA

## 1.1. Planteamiento del problema

Con el constante aumento en la comunidad estudiantil de la Pontificia Universidad Javeriana Cali, se siente la urgente necesidad de tener un sistema que permita ayudar a mantener el control sobre los parqueaderos, de tal forma que ayude a las personas y a sus vehículos a encontrar lugares disponibles de manera eficiente y rápida. Puesto que el caos que se genera conlleva a un consumo de tiempo y gasolina, que a largo plazo puede causar problemas de salud derivados del estrés y la polución.

## 1.2. Benchmarking

Antes de iniciar las fases de diseño y desarrollo del proyecto, es necesario realizar una fase de investigación en la cual se revisen las diferentes aproximaciones y propuestas que existen para poder resolver la problemática descrita anteriormente. De esta manera, se analizaron y obtuvieron 6 soluciones innovadoras, tres nacionales y tres internacionales, de las cuales se recopiló información significativa para encaminar el proyecto hacia la solución deseada.

A continuación, se encuentra un breve resumen de las propuestas escogidas y las conclusiones que se obtuvieron de las mismas.

### 1.2.1. Soluciones nacionales

En el territorio nacional se han llevado a cabo diferentes implementaciones para atacar la problemática que conllevan la falta de parqueaderos y la administración de los mismos.

#### 1.2.1.1. Eco Parking

Es la solución de parqueo implementado por la empresa Laboratorio Clínico Hematológico de Medellín, la cual consiste de un parqueadero inteligente que consta de un software encargado de identificar y mostrar una celda o espacio disponible frente al usuario, sin necesidad de que este se vea obligado a rondar por el establecimiento. El costo de desarrollo de cada celda es aproximadamente 30 millones de pesos. Esto podría deberse al hecho de que es totalmente automatizado y requiere de un alto costo de construcción y un amplio espacio para ser implementado eficientemente. (Para más información favor visitar el sitio web http://ecoparking.co/)

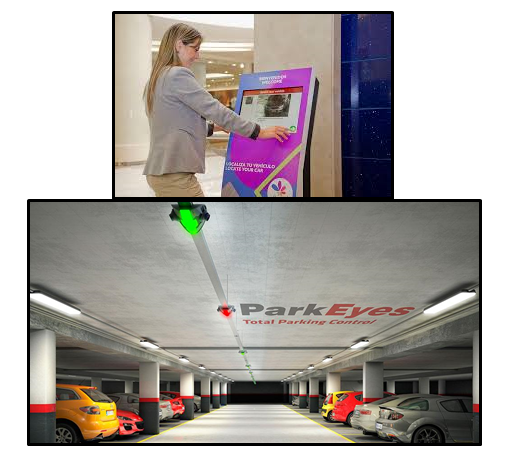
Figura 1. Eco Parking y su solución

**

#### 1.2.1.2. ParkEyes

El sistema ParkEyes, proyecto implementado por el centro comercial Santafé de Medellín, tuvo un costo de inversión de 2300 millones de pesos. Este trata de un parqueadero inteligente al cual se puede acceder por medio de pantallas táctiles ubicadas en el centro comercial y sólo necesita como parámetro la placa del vehículo para que el sistema le muestre al usuario su vehículo sin importar en qué espacio se haya estacionado. (Para más información, favor visitar la página http://parkeyes.com/?lang=es)

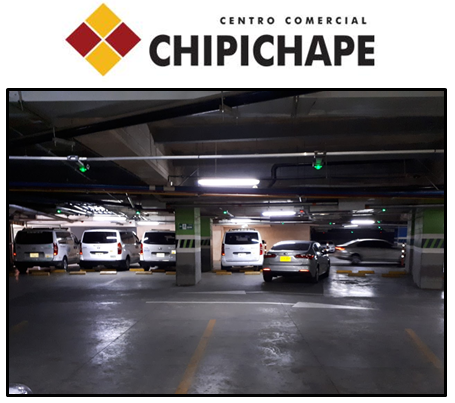
Figura 2. ParkEyes y su solución



#### 1.2.1.3. Centro comercial Chipichape

Esta solución fue implementada en el parqueadero del Centro Comercial Chipichape que se encuentra en la ciudad de Cali. Al igual que las otras soluciones, trata de resolver el problema de encontrar un espacio disponible, haciendo uso de sensores conectados a bombillos leds que se iluminan de color rojo o verde, indicando si el espacio de parqueo se encuentra ocupado o está libre, respectivamente. Estos componentes de leds y sensores están ubicados en la parte superior de los espacios de parqueo, lo cual ayuda a aumentar su visibilidad.

Figura 3. Chipichape y su solución



### 1.2.2. Soluciones internacionales

Para ampliar el margen de estudio, se optó por revisar soluciones implementadas en otros países como lo son Canadá, Estados Unidos e Italia, esto con el fin de disminuir el riesgo de sesgar la investigación.

#### 1.2.2.1. Parkifi

Es una aplicativo desarrollado en la ciudad de Denver, Colorado, USA, el cual hace uso de un tipo específico de sensores (magnetómetros) los cuales, a petición del usuario, son instalados en espacios disponibles y en los que él mismo tenga propiedad. Estos sensores envían información acerca del precio y la disponibilidad de un espacio de parqueo y lo muestran al usuario por medio de la aplicación móvil. El costo por sensor es aproximadamente de 60 USD (dólares estadounidenses), esto sin contar el precio por instalación. (Para más información, favor visitar el sitio web https://www.parkifi.com/)

Figura 4. Parkifi y su solución

**

#### 1.2.2.2. Rover Parking

El sistema Rover Parking, proyecto realizado en Toronto, Ontario, Canadá, pretende dar solución a la problemática descrita mediante el uso de un aplicativo móvil, el cual funciona y se maneja exclusivamente por un medio de comunicación persona a persona (chat peer to peer). En el mapa de la aplicación se describe la ubicación del espacio que uno de los usuarios desea prestar como servicio a otros usuarios. Al momento en el que otro usuario requiere un espacio disponible, simplemente selecciona en el mapa el espacio que desea utilizar, lo reserva añadiendo el tiempo de duración de su estadía y después se le realiza un cobro por el tiempo utilizado. (Para más información, favor visitar el sitio web https://roverparking.com/)

Figura 5. Rove Parking y su solución



#### 1.2.2.3. Park Smart

Park Smart es un aplicativo móvil desarrollado en Catania, Sicilia, Italia, que hace uso de múltiples cámaras con la cuales se recopila información y se analiza el estado del estacionamiento con un hardware propio que se encarga de comunicarle al usuario el estado del parqueadero a través de un aplicativo móvil. Uno de los puntos interesantes es que este proyecto permite ser integrado con diferentes sistemas, pues tiene una API que puede ser accedida por un usuario, siempre que este contrate el servicio. (Para más información, favor visitar el sitio web https://www.parksmart.it/)

Figura 6. Park Smart y su solución

**

### 1.2.3. Comparación y conclusiones

Teniendo en cuenta las soluciones nacionales descritas se puede decir que los sistemas analizados son bastante avanzados en cuanto a tecnología, pero también se puede concluir que no solucionan en su totalidad el problema. En el primer caso (Eco Parking), no todos los establecimientos tienen disponible un espacio tan grande para su construcción, ni pueden llegar a ver como necesario la implementación de un sistema tan complejo y costoso. El segundo caso (ParkEyes), no es una solución directa al problema, ya que se ofrece al usuario saber la posición del vehículo una vez se encuentra estacionado, además, solamente genera eficiencia en la salida de vehículos y en el sentimiento de confianza y seguridad por la vigilancia automática provista. En estos dos casos se puede identificar un problema grave, lo cual es su elevado costo de adecuación, implementación y mantenimiento, lo que implica que no son soluciones aplicables a todos los establecimientos, pues no todos tienen la capacidad de equipar su parqueadero con este tipo de tecnologías. Para el tercer caso (Centro comercial Chipichape), se da una respuesta parcial al problema, permitiendo conocer el estado y la ubicación del vehículo, pero para ello es necesario haber ingresado al establecimiento y, usualmente, deambular un cierto tiempo para encontrar un espacio libre, además, al analizar con detalle esta solución se evidencia lo limitada de la misma, puesto que es necesario estar en la vía del parqueadero disponible para poder detectarlo. Es posible que el usuario demore una cantidad de tiempo considerable antes de llegar a un espacio libre, dado que no se puede saber con antelación la cantidad aproximada de espacios libres ni la ubicación exacta de cada uno de los mismos en tiempo real. Además de todo esto, la sección del parqueadero que provee esta solución es solo una pequeña fracción del espacio total con el que cuenta el establecimiento (debido a la infraestructura requerida y problemas ambientales). Con todo esto se puede decir que, a nivel nacional, no se encontró ningún aplicativo ni sistema que dé solución a este problema de manera completa.

En contraste, los sistemas internacionales que fueron analizados ofrecen soluciones que, en general, se puede decir son más completas y no tan costosas como las nacionales. Empezando por la primera opción analizada (Parkifi), que es la más completa y cercana a lo que se desea realizar con diferencias como el cobro por la instalación de los sensores y el uso de su plataforma a los conductores, lo cual en el proyecto no se pretende dar ya que existirá un establecimiento de por medio (La universidad) que asumirá estos costos. La segunda opción revisada (Rover Parking) es también una solución bastante completa al problema, aunque tiene dos distintivos importantes; el primero sería el hecho de que no utiliza ningún sistema de detección para informar acerca de la disponibilidad del espacio, sino que asume que la información provista por ambos usuarios es correcta, posiblemente creando incertidumbre y problemas debido a la falta de exactitud en la información, dado que esta misma es creada por usuarios y es más susceptible a fallos y malentendidos; y la segunda es que no existiría una entidad de por medio, lo cual causaría que, al igual que la anterior solución, los conductores asuman el costo. Por último, la tercera opción (Park Smart) es la solución que abarca en gran medida la problemática principal, ya que permite que los establecimientos hagan uso de su propia infraestructura para conectarse con el aplicativo, pero, así mismo, este requiere de un hardware que sirva de intermediario, encargado de transformar la información para que pueda ser consumida por los usuarios.

A continuación, se puede observar una tabla comparativa entre los anteriores casos analizados y nuestra propuesta.

Tabla 1. Comparación entre las soluciones

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Eco Parking | ParkEyes | Chipichape | Parkifi | Rover Parking | Park Smart | Nuestra propuesta |
| El costo por espacio de parqueo es menor a $120.000 COP |  |  |  | ✓ | ✓ | ✓ | ✓ |
| Se conoce el estado del parqueadero | ✓ |  | ✓ | ✓ |  | ✓ | ✓ |
| La información provista es actualizada en tiempo real | ✓ | ✓ | ✓ | ✓ |  | ✓ | ✓ |
| Da a conocer espacios disponibles | ✓ |  | ✓ | ✓ |  | ✓ | ✓ |
| No está ligado a hardware propio |  |  |  | ✓ | ✓ |  | ✓ |
| No requiere de espacios o modificaciones grandes para su aplicación |  |  |  | ✓ |  | ✓ | ✓ |
| Recolecta datos para el establecimiento |  |  |  | ✓ |  |  | ✓ |
| Requiere conexión a internet |  |  |  | ✓ |  | ✓ | ✓ |
| Muestra la información en un dispositivo personal del usuario |  |  |  | ✓ | ✓ | ✓ | ✓ |
| No supone ningún costo para los conductores | ✓ | ✓ | ✓ |  |  | ✓ | ✓ |
| Manejo de la aplicación por voz |  |  |  |  |  |  | ✓ |

En la parte izquierda de la tabla se evidencia las características principales de cada uno de los proyectos analizados, y en la parte superior se muestran las soluciones.

# 2. OBJETIVOS

## 2.1. Objetivo general

Diseñar y desarrollar una herramienta tecnológica que facilite la visualización y localización de espacios disponibles para parquear desde un teléfono móvil inteligente.

## 2.2. Objetivos específicos

* Realizar una búsqueda y análisis de necesidades de los usuarios alrededor del problema.
* Diseñar y desarrollar un prototipo no funcional con el cual se pueda obtener información de las necesidades de los usuarios, haciendo enfoque en la parte visual del software del cliente.
* Diseñar y desarrollar la herramienta tecnológica que permita identificar cuando un parqueadero está libre.
* Diseñar y desarrollar un prototipo funcional que permita gestionar la información obtenida del sistema de detección y visualizarla en un dispositivo móvil, adquiriendo, además, retroalimentación de su funcionamiento con los usuarios finales.
* Diseñar y ejecutar pruebas funcionales y pruebas de usuario de la solución.
* Diseñar un modelo de negocio que permita validar la posible comercialización de la plataforma.

# 3. MARCO TEÓRICO

Además de las soluciones descritas anteriormente, se estudiaron las siguientes investigaciones, con el objetivo de esclarecer aún más el cómo se llevaría a cabo el diseño y desarrollo del proyecto.

## 3.1. An embedded sensing approach to monitoring parking lot occupancy (*Un enfoque de detección embebido para monitorear la ocupación del estacionamiento)*

Dentro de esta investigación se propone una idea rentable y de alta fidelidad la cual consiste en una solución no convencional para detectar los espacios disponibles, por medio de una red de sensores de sonido y un algoritmo de eliminación de ruido. Esta solución puede implementarse virtualmente en cualquier tipo de estacionamiento sin realizar grandes cambios a la infraestructura del mismo. Además, es una solución que puede ser considerada relativamente viable, costando aproximadamente $350 USD por el sistema ubicado en la entrada/salida del parqueadero.

## 3.2. Building sustainable parking lots with the Web of Things *(Construyendo estacionamientos sostenibles usando el Internet de las Cosas)*

En esta investigación se propone el uso del Internet de las Cosas (IoT, Internet of Things) para generar una infraestructura escalable con la cual se puede tener una mejor administración de los parqueaderos. El sistema utiliza la arquitectura orientada al servicio, que permite a los conductores realizar reservas a los espacios de parqueadero dentro de los establecimientos y consultar el estado del parqueadero sin tener que estar dentro del mismo, por medio del internet. Además, una evaluación del mismo estudio demostró una reducción de máximo un 40% en el tiempo para encontrar un espacio libre, junto con un 30% de reducción en la polución generada hacia el medio ambiente por los mismos vehículos.

## 3.3. Applying mobile and internet of things technologies in managing parking spaces for people with disabilities *(Aplicando tecnologías móviles e internet de las cosas al manejo de espacios de parqueo para personas con discapacidades)*

Este proyecto fue desarrollado con la intención de ayudar a aquel porcentaje de la población que sufre de alguna discapacidad y que además poseen automóviles en los cuales se desplazan. Debido a que los ciudadanos con discapacidad tienen un número limitado de espacios asignados para su uso exclusivo, a menudo no son suficientes para satisfacer la demanda. Por tal motivo, se diseñó un sistema inteligente que utiliza sensores para recolectar información de estos mismos espacios y determinar si se encuentran o no ocupados, de igual manera, les permite a los usuarios registrados consultar y reservar los espacios de estacionamiento en tiempo real.

## 3.4. Wireless sensor networks and service-oriented architecture, as suitable approaches to be applied into ITS *(Redes de sensores inalambricos y arquitectura orientada a servicios como enfoques adecuados para aplicarse en los SIT)*

En este trabajo se analizan los enfoques de Redes de Sensores Inalámbricos (WSN, Wireless Sensor Networks) y Arquitectura Orientada a Servicios (SOA, Service Oriented Architecture), que representan un papel clave y emergente del Sistema Inteligente de Transporte (ITS, Intelligent Transportations Systems), con el cual se puede dar solución a múltiples problemas orientados a viajes y tráfico de vehículos por medio de sensores, redes inteligentes y demás instrumentos de última tecnología. Esta solución se enfoca, además, en la generación de una solución eficiente para el problema de encontrar espacios libres para que los usuarios puedan aparcar fácilmente dentro de la Universidad de Alicante.

## 3.5. Low Cost Internet of Things based Vehicle Parking Information System *(Sistema de información de parqueo basado en Internet de las Cosas de bajo costo)*

Este documento describe una solución de bajo costo con la cual se alivia el problema de encontrar espacio dentro de un estacionamiento. Este sistema hace uso de la tecnología de comunicación inalámbrica para transmitir y recibir datos desde sensores ubicados en los espacios de parqueo, los cuales se almacenan en la nube, asegurando alta confiabilidad y disponibilidad del sistema. Además, permite al usuario darse cuenta de la disponibilidad de un espacio vacío a través de una aplicación móvil en tiempo real. El sistema se desacopla y se presenta de tal manera que muchas zonas de estacionamiento se pueden agregar y quitar sin afectar la funcionalidad básica del sistema.

## 3.6. Car parking management system using AMR-sensor technology *(Sistema de administración de parqueadero usando sensores magnéticos)*

En este paper se propone un sistema de visualización de parqueadero por medio de un panel LED en la entrada el estacionamiento y una aplicación móvil para que el usuario mantenga informado acerca de los cambios en los espacios disponibles. Para detectar la presencia de los espacios libres se hace uso de sensores magneto resistivos, los cuales permiten identificar cambios considerables en todas las direcciones. Además, se dan algunas pautas para el manejo debido del sistema y el costo aproximado que este podría tener.

## 3.7. Real time car parking system using image processing *(Sistema de parqueadero en tiempo real usando procesamiento de imágenes)*

En esta investigación se hace uso de una cámara situada en la entrada del parqueadero para poder detectar los vehículos y realizar un conteo acerca del número de espacios disponibles dentro del establecimiento. Para la detección de los automóviles se optó únicamente por el uso de la cámara sin recurrir a sensores ni demás elementos, la cual identifica a los vehículos por medio de detección de bordes y el operador prewitt, aplicando una serie de algoritmos y operaciones matemáticas complejas para realizar un procesamiento de la imagen adecuado y objetivo, lo que permite identificar los vehículos con una mayor precisión.

## 3.8. iParking: An intelligent parking system for large parking lots *(Iparking: Un Sistema de paqueo inteligente para grandes establecimientos)*

En esta investigación se utilizaron una serie de sensores conectados a unos routers que envían la información pertinente a un servidor web, que más tarde era consumida por medio de un servidor web y aplicativos móviles. Para esto se utilizaron sensores magnéticos, los cuales reducen el ruido y aprovechan la cantidad de materiales ferromagnéticos presentes en los vehículos. Se optó por realizar un protocolo de comunicación por demanda y utilizar estados de hibernación en los nodos, todo esto con el fin de reducir la cantidad de energía que gasta el sistema.

## 3.9. Performance Analysis of Proximity and Light Sensors for Smart Parking *(Análisis de desempeño para sensores de proximidad y luz en parqueaderos inteligentes)*

Para esta investigación se realizó una comparación entre un par de sensores para poder encontrar el que pudiera detectar de una manera más objetiva y precisa un vehículo. La comparación se realizó entre sensores de resistencia dependientes de la luz (fotorresistencias) y sensores infrarrojos. El estudio determinó no solamente que los sensores infrarrojos eran mejores para la detección de los vehículos, ya que detectaban con una mayor claridad la presencia de un automóvil, sino que además demostró la dependencia al medio ambiente que poseen las fotorresistencias para mostrar resultados objetivos, ya que dependen de factores tan simples como la cantidad de luz solar, el clima, etc. A pesar de esto se advierte que optar por sensores infrarrojos no es una opción muy eficiente en cuanto a costo se refiere.

## 3.10. Automatic Smart Parking System using Internet of Things (IOT) *(Sistema de parqueo automático inteligente usando el Internet de las Cosas)*

Este proyecto fue pensado para ser aplicado y adaptado a los distintos parqueaderos independiente del terreno o ubicación. Para su desarrollo se utilizaron herramientas como el raspberry pi, la pi-camera, un servidor centralizado y uno en la nube para el intercambio de datos. La cámara de la raspberry pi, la cual se puede ubicar fácilmente en el techo de un parqueadero cerrado o sobre un poste de luz, sirve para recopilar la información que, mediante el procesamiento de imágenes, determina constantemente el estado de los parqueaderos para proveer a los usuarios con información exacta en tiempo real.

# 4. DESARROLLO

## 4.1. Metodologías

A continuación, se describen las metodologías utilizadas para el diseño y desarrollo del proyecto, con el fin de obtener software de buena calidad durante un lapso corto, tratando de cumplir de manera correcta las necesidades de los usuarios.

### 4.1.1. Design Thinking

La metodología de diseño escogida para este proyecto fue *Design Thinking,* esto debido a que es una metodología que se centra fuertemente en el uso de iteraciones y conexión con las necesidades de usuario para lograr su objetivo.

Todo proceso de Design Thinking se compone de cinco etapas no lineales, es decir, no necesariamente consecutivas, por lo cual se puede saltar entre las etapas del proyecto en cualquier momento si este así lo requiere. El proceso comienza con la recolección de una gran cantidad de información, generando así una suma de contenido útil, que crecerá o disminuirá dependiendo de la cantidad de “ciclos” que se vayan ejecutando. En la siguiente imagen se visualiza el proceso no lineal con sus respectivos ciclos:

Figura 7. Etapas de Design Thinking



A lo largo del proceso se irá refinando tanto el contenido útil de información como el desarrollo mismo de la solución, hasta que se pueda obtener una que cumpla con los objetivos de manera clara y satisfactoria.

Figura 8. Refinamiento de una solución



A continuación, se describen los 5 procesos o etapas diferenciadas que la metodología de Design Thinking ofrece:

#### 4.1.1.1. Empatizar

El proceso de Design Thinking comienza con una profunda comprensión de las necesidades de los usuarios implicados en la solución que se esté desarrollando, así como también del entorno en el cual se desarrolla. Se debe ser capaz de “ponerse en la piel” de dichas personas para generar soluciones consecuentes con sus realidades, ya que es una forma bastante acertada de dar un resultado preciso y objetivo con el contexto y entorno dados.

Algunas técnicas utilizadas en esta fase son:

* Mapa de actores
* Mapa de empatía
* Usuarios extremos
* Mapa del presente, mapa del futuro

#### 4.1.1.2. Definir

La etapa de Ideación tiene como objetivo la generación de un ‘sinfín’ de opciones, idea análoga a la “lluvia de ideas”. No se debe apegar a la primera opción que se obtenga, ya que esta fase busca la generación de la mayor cantidad posible de ideas. En esta, las actividades favorecen el pensamiento expansivo, lo cual evita generar juicios sobre las ideas obtenidas, ya que esto empezaría debates sobre el verdadero valor o importancia de estas sin un análisis previo. Esta etapa se basa en la idea de que ciertas veces las ideas poco comunes son las que generan soluciones visionarias. Algunas técnicas usadas en esta fase son:

* Lluvia de ideas
* Storytelling

#### 4.1.1.3. Prototipar

En la fase de Prototipado es donde las ideas se vuelven realidad, ya que se construyen prototipos que generan las ideas palpables y se ayuda a visualizar las posibles soluciones, poniendo de manifiesto elementos que se deben mejorar o refinar antes de llegar al resultado final. Algunas técnicas usadas en esta fase son:

* Prototipado en imágenes
* Prototipado en bruto

#### 4.1.1.4. Testear

Durante la fase de *Testeo*, se prueban los prototipos con los usuarios implicados en la solución que se está desarrollando. Esta fase es crucial y ayuda a identificar mejoras significativas, fallos a resolver, posibles carencias y demás caminos o defectos de las soluciones encontradas. Durante esta fase evoluciona la idea seleccionada hasta convertirla en la solución que se estaba buscando. Algunas técnicas usadas en esta fase son:

* Pruebas de usabilidad
* Casos de uso
* Storyboard

### 4.1.2. SCRUM

*Scrum* es una metodología de desarrollo ágil, destinada a la producción exitosa de un proyecto de software mediante un equipo competente y dedicado, mediante el uso de una serie de iteraciones que tienen como propósito ir creando y mejorando prototipos funcionales de acuerdo a las necesidades, requerimientos y satisfacción del cliente. Los componentes en los que se divide y define Scrum son los siguientes:

#### 4.1.2.1. Roles en Scrum: ¿Quién? y ¿Qué?

En esta metodología existen tres roles, que nos ayudan a guiar el proyecto y permiten una división clara del trabajo pertinente para cada uno de los miembros del equipo:

* **El Product Owner / Dueño del producto:** Es la persona encargada de tener una mayor interacción con el cliente, de escuchar sus sugerencias y transmitir la información al resto del equipo, razón por la cual es considerado como la “voz del cliente” y el responsable de desarrollar, mantener y priorizar las tareas en el backlog.
* **El Scrum Master:** Es la persona responsable de asegurarse que el trabajo del equipo vaya de acuerdo con las directrices de la metodología Scrum, además de no dejar que se pierda el objetivo del proyecto. También se encarga de remover cualquier obstáculo con el cual se pueda topar el equipo de desarrollo y se encarga de que las metas se cumplan en un tiempo establecido o pertinente. Por esto mismo se puede decir que es quien vela por la seguridad y el buen desarrollo tanto del equipo como del proyecto.
* **El Development Team / Equipo de desarrollo:** Son las personas encargadas de escribir y probar el código, siguiendo las tareas y guías dictadas por el Scrum Master.

#### 4.1.2.2. El Sprint: ¿Dónde? ¿Cuándo?

El Sprint es la unidad básica por medio de la cual el equipo de trabajo mantiene una visión objetiva durante todo el proyecto. Esta es la característica principal que marca la diferencia entre Scrum y otros modelos para el desarrollo ágil, que puede traducirse en una simple iteración llevada a cabo por los miembros del equipo durante toda la vida útil del proyecto y que, al final de cada iteración, se obtiene un prototipo funcional mejor y más completo que el anterior. Un equipo puede completar varios sprints durante el desarrollo del proyecto.

Un Sprint requiere, para su inicio, un equipo que se compromete a realizar el trabajo y, para su finalización, requiere la demostración de un entregable en el que se refleja el trabajo desarrollado. El tiempo mínimo promedio para un Sprint suele ser de una semana y el máximo promedio tiende a ser 4 semanas, tiempos que pueden variar de acuerdo con el tamaño, duración y complejidad del proyecto.

Dentro del desarrollo de un Sprint se llevan a cabo ciertos eventos, denominados, Scrum Events o Eventos Scrum. Estos son:

* **Planeamiento del Sprint / Sprint Planning:**

Todos los miembros del equipo se reúnen para planificar el Sprint. Durante este evento se decide qué requerimientos o tareas se le asignan a cada uno de los miembros del mismo. Cada integrante deberá asignar el tiempo que crea prudente para llevar a cabo sus requerimientos, ya que de esta manera se define el tiempo de duración del Sprint y los elementos que se van a desarrollar en el mismo.

* **Reunión de Equipo de Scrum / Scrum team meeting:**

Estas reuniones se deben realizar diariamente y no deben durar más de 15 minutos. En ellas, cada miembro del equipo deberá responder tres simples preguntas:

* + ¿Qué hiciste ayer?
  + ¿Qué tienes planeado hacer hoy?
  + ¿Qué obstáculos encontraste en el camino?

Estas reuniones sirven para que todos los miembros del equipo se apoyen entre ellos, ya que si alguno tiene algún inconveniente o problema en el trabajo que se vea reflejado negativamente en el tiempo asignado, este debe tratarse más a fondo en una reunión enfocada a buscar la mejor solución para ello, así como asignar más personal y recursos a la resolución del mismo.

* **Refinamiento del Backlog / Backlog Refinement:**

El Product Owner revisa cada uno de los elementos dentro del Product Backlog con el fin de esclarecer cualquier duda que pueda surgir por parte del equipo de desarrolladores. También sirve para volver a estimar el tiempo y esfuerzo dedicado a cada uno de los requerimientos.

* **Revisión del Sprint / Sprint Review:**

Los miembros del equipo y los clientes se reúnen para mostrar el trabajo de desarrollo de software que se ha completado. Además, se hace una demostración de todos los requerimientos finalizados dentro del Sprint. En este punto no es necesario que todos los miembros del equipo formen parte de la conversación, los cuales pueden o no estar presentes, ya que la presentación está a cargo del Scrum Master y el Product Owner.

* **Retrospectiva del Sprint / Retrospective:**

En este evento el Product Owner se reúne con todo su equipo de trabajo y su Scrum Master para hablar sobre lo ocurrido durante el Sprint. Los puntos principales por tratar en esta reunión son:

* + ¿Qué se hizo mal durante el Sprint para poder mejorar el próximo?
  + ¿Qué se hizo bien para seguir en la misma senda del éxito?
  + ¿Qué inconvenientes se encontraron y no permitieron poder avanzar como se tenía planificado?

## 4.2. Levantamiento de requerimientos

Para poder entender las necesidades de los usuarios es necesario realizar una investigación sobre las mismas, así como también documentarlas de manera objetiva, de tal forma que no existan conflictos más adelante por malentendidos en cuanto a los requerimientos.

A continuación, se mostrará el proceso por medio del cual se desarrollaron los diagramas de secuencias y las historias de usuario.

### 4.2.1. Entrevistas a los usuarios

Como paso inicial en el desarrollo del proyecto, se realizaron entrevistas a usuarios potenciales de la Pontificia Universidad Javeriana Cali y la Universidad Icesi, con el objetivo de obtener información valiosa acerca de la problemática de los parqueaderos desde el punto de vista de estudiantes y profesores y ver cómo mejorar la plataforma que se deseaba desarrollar (Ver anexo 11.2. Grabaciones de las entrevistas).

Las entrevistas se realizaron teniendo en cuenta la metodología de diseño *Design Thinking*, por lo que las preguntas (Ver anexo 11.1. Preguntas de las entrevistas) no tenían un orden específico, es decir, se planteaban dependiendo de cómo fluyera la conversación.

### 4.2.2. Necesidades de los usuarios

Se listan las necesidades de los usuarios obtenidas en las entrevistas realizadas.

* El aplicativo debería mostrar información acerca del parqueadero de un establecimiento, es decir, la hora de apertura, la hora de cierre, el costo de parqueo por hora, en caso de que el parqueadero cobre.
* El aplicativo debería ser llamativo, usar colores que le generen a las personas recordación de este.
* El aplicativo debería ser intuitivo, mostrando información acerca de las opciones y botones disponibles.
* El aplicativo debería tener un marcador indicando la posición actual del usuario en el mapa.

### 4.2.3. Historias de usuario

|  |  |
| --- | --- |
| Historia de Usuario #1 | |
| Nombre historia: Registrarse en el aplicativo | |
| Prioridad: Alta | Riesgo en desarrollo: Baja |
| Puntos estimados: 2 | Iteración asignada: 1 |
| **Descripción:**  Quiero registrarme para poder hacer uso del aplicativo. | |
| **Validación:**  El usuario puede crear una cuenta registrándose directamente desde el aplicativo móvil o web. | |

|  |  |
| --- | --- |
| Historia de Usuario #2 | |
| Nombre historia: Iniciar sesión en el aplicativo | |
| Prioridad: Alta | Riesgo en desarrollo: Baja |
| Puntos estimados: 2 | Iteración asignada: 1 |
| **Descripción:**  Quiero iniciar sesión en el aplicativo con mi cuenta registrada. | |
| **Validación:**  El usuario puede ingresar al aplicativo haciendo uso del correo y contraseña con el que se haya registrado. | |

|  |  |
| --- | --- |
| Historia de Usuario #3 | |
| Nombre historia: Consultar un establecimiento | |
| Prioridad: Alta | Riesgo en desarrollo: Alta |
| Puntos estimados: 3 | Iteración asignada: 2 |
| **Descripción:**  Quiero consultar el estado de los parqueaderos de un establecimiento que seleccione. | |
| **Validación:**  El usuario puede realizar la consulta de los parqueaderos y su estado actual (ocupado o desocupado), de cualquier establecimiento que sea seleccionado. | |

|  |  |
| --- | --- |
| Historia de Usuario #4 | |
| Nombre historia: Consultar un establecimiento por voz | |
| Prioridad: Media | Riesgo en desarrollo: Alta |
| Puntos estimados: 3 | Iteración asignada: 3 |
| **Descripción:**  Quiero consultar un establecimiento haciendo uso del micrófono del celular, sin necesidad de tener que escribir el nombre. | |
| **Validación:**  El usuario puede consultar los establecimientos registrados en el aplicativo por medio de su voz, haciendo uso del micrófono del dispositivo móvil. | |

|  |  |
| --- | --- |
| Historia de Usuario #5 | |
| Nombre historia: Cambiar el idioma del aplicativo | |
| Prioridad: Baja | Riesgo en desarrollo: Baja |
| Puntos estimados: 1 | Iteración asignada: 4 |
| **Descripción:**  Quiero cambiar el idioma con el que se muestra el aplicativo. | |
| **Validación:**  El usuario puede seleccionar el idioma con el que quiere que el aplicativo se muestre. | |

|  |  |
| --- | --- |
| Historia de Usuario #6 | |
| Nombre historia: Cambiar el color del aplicativo | |
| Prioridad: Baja | Riesgo en desarrollo: Baja |
| Puntos estimados: 1 | Iteración asignada: 4 |
| **Descripción:**  Quiero cambiar los colores con los que se muestra el aplicativo. | |
| **Validación:**  El usuario puede seleccionar un tema de los predefinidos, el cual cambia los colores con los que se puede visualizar el aplicativo. | |

|  |  |
| --- | --- |
| Historia de Usuario #7 | |
| Nombre historia: Guiar el usuario hasta un parqueadero | |
| Prioridad: Alta | Riesgo en desarrollo: Media |
| Puntos estimados: 3 | Iteración asignada: 5 |
| **Descripción:**  Quiero llegar hasta un parqueadero disponible. | |
| **Validación:**  El usuario puede visualizar la ruta desde su posición hasta la de un parqueadero libre, dentro del aplicativo. | |

|  |  |
| --- | --- |
| Historia de Usuario #8 | |
| Nombre historia: Editar la información de una cuenta | |
| Prioridad: Media | Riesgo en desarrollo: Baja |
| Puntos estimados: 2 | Iteración asignada: 6 |
| **Descripción:**  Quiero modificar mi información personal. | |
| **Validación:**  El usuario puede cambiar cierta información de su cuenta de usuario. | |

|  |  |
| --- | --- |
| Historia de Usuario #9 | |
| Nombre historia: Cambiar configuración del mapa | |
| Prioridad: Media | Riesgo en desarrollo: Alta |
| Puntos estimados: 2 | Iteración asignada: 6 |
| **Descripción:**  Quiero cambiar la vista del mapa, el tipo del marcador para los parqueaderos y mi posición actual. | |
| **Validación:**  El usuario puede cambiar el tipo de vista del mapa, la forma de los marcadores y la posición actual en el mapa. | |

### 4.2.4. Diagramas de secuencia

Figura 9. Diagrama de inicio de sesión



Diagrama de secuencia para el inicio de sesión en el aplicativo.

Figura 10. Diagrama de cierre de sesión

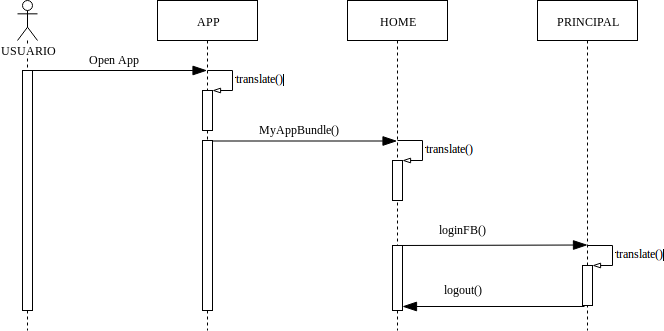


Diagrama de secuencia para el cierre de sesión o *logout* del aplicativo.

Figura 11. Diagrama de registro de usuario

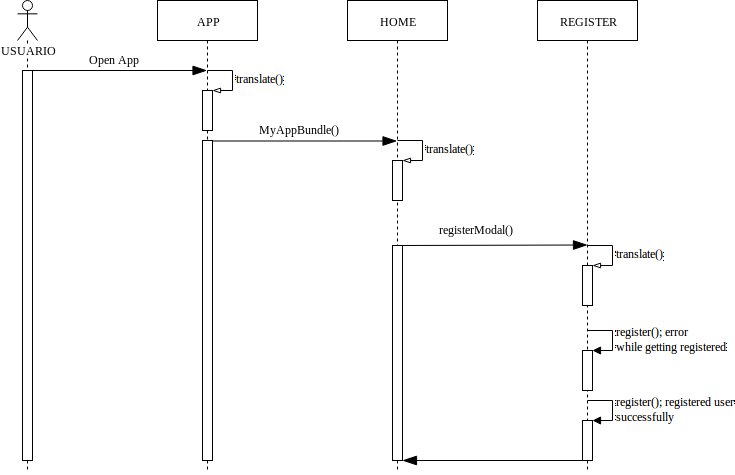


Diagrama de secuencia para el registro de un usuario desde el aplicativo.

Figura 12. Diagrama de consulta de establecimiento

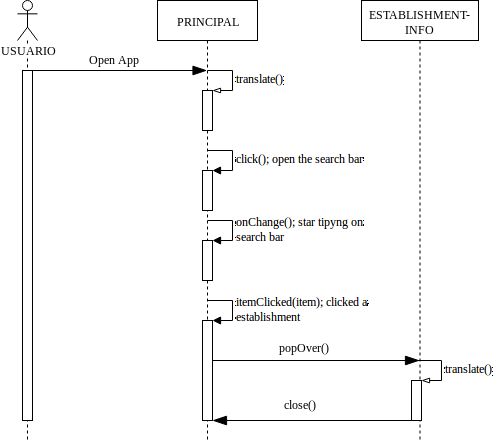


Diagrama de secuencia para la consulta de un establecimiento, digitando el nombre del establecimiento en la barra de búsqueda.

Figura 13. Diagrama de navegación de menú

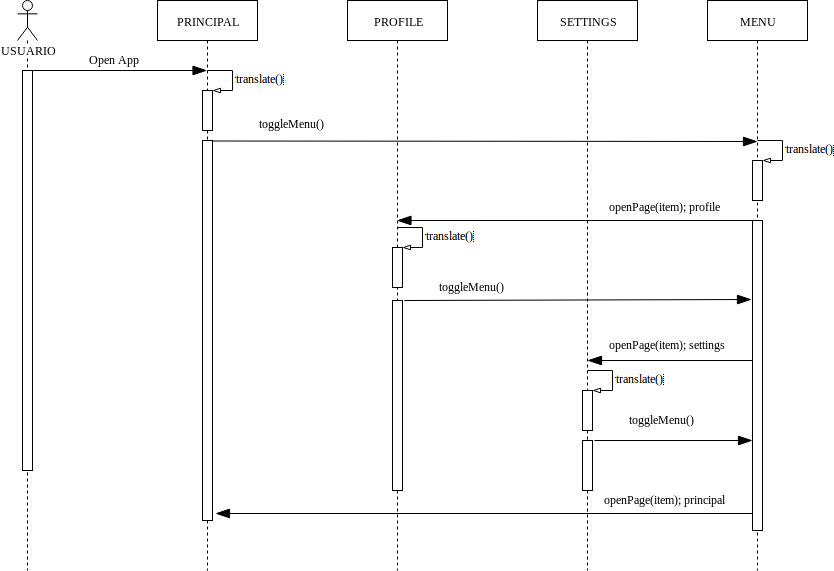


Diagrama de secuencia de navegación en el menú de opciones ubicado en el lado izquierdo del aplicativo.

Figura 14. Diagrama de administración de perfil

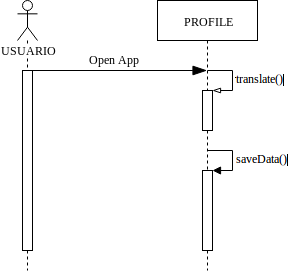


Diagrama de secuencia para la administración y modificación de información personal del perfil de usuario.

Figura 15. Diagrama de administración de configuración

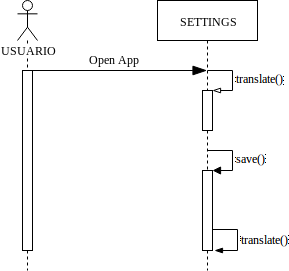
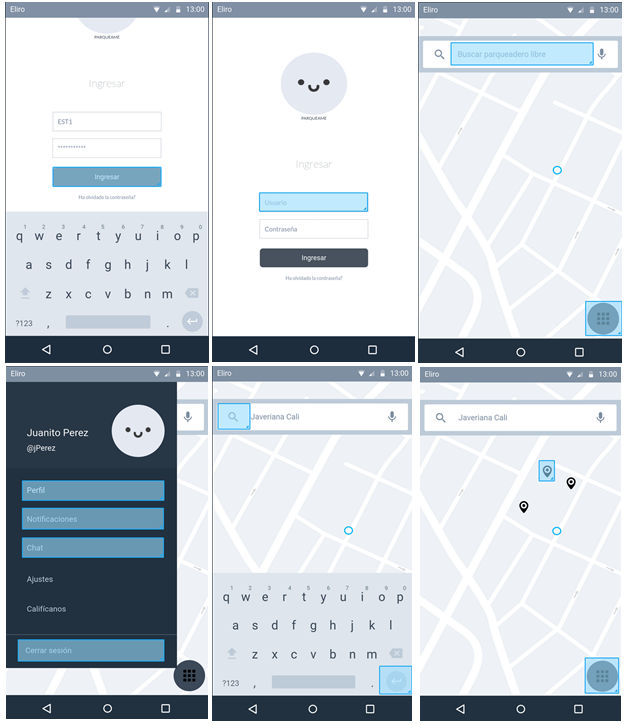


Diagrama de secuencia para la administración y configuración de las opciones del aplicativo, correspondientes al lenguaje, color y tipo de mapa a mostrar en el aplicativo.

## 4.3. Prototipo inicial

Para extraer las necesidades e información importante de los usuarios se hizo uso de la etapa de Prototipo de Design Thinking, realizando un mockup simple, el cual fue diseñado como una aproximación a lo que se creía podría llegar a lucir la app a futuro. Este prototipo inicial se llevó a cabo haciendo uso de la herramienta Marvel App, la cual permite crear y animar mockups que pueden ser ejecutados en dispositivos móviles.

Figura 16. Prototipo inicial



Después de tener diseñado el prototipo inicial, fueron llevadas a cabo pruebas de campo con las que se obtuvo información para el diseño y desarrollo del prototipo funcional final.

Los resultados obtenidos son los siguientes:

* El aplicativo debe ser colorido, para que capte la atención de los usuarios.
* El aplicativo deberá contar con comandos de voz para buscar los parqueaderos.
* Debería de mostrarse los espacios disponibles y ocupados.
* El aplicativo debe estar siempre atento, para captar cuando un parqueadero cambie su estado de ocupado a desocupado y viceversa.

## 4.4. Prototipo funcional / Prototipo final

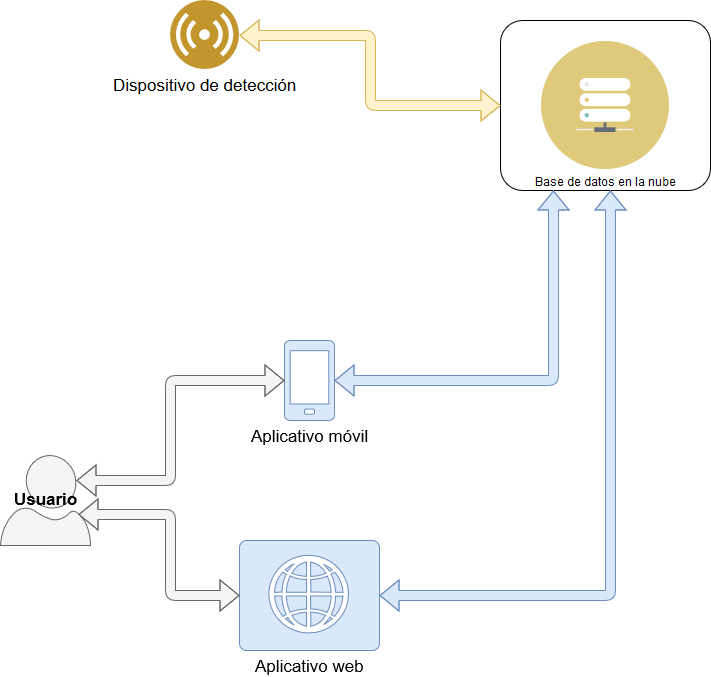
### 4.4.1. Tecnologías

Antes de llevarse a cabo el desarrollo del prototipo funcional, se estudiaron diferentes opciones tecnológicas que pudieran servir como punto de partida para el proyecto, teniendo en cuenta la arquitectura definida y las necesidades de los usuarios previamente analizadas.

Durante esta fase de diseño se revisó la viabilidad de diferente software, hardware y frameworks para el desarrollo del aplicativo móvil y del dispositivo que detectaría los vehículos.

A continuación, se presentan las tecnologías utilizadas para el Back-end, Front-end y dispositivo de detección que se utilizaron finalmente para el desarrollo del prototipo funcional final.

Figura 17. Arquitectura del proyecto

**

#### 4.4.1.1. Back-end

En un comienzo se había optado por trabajar con Node JS y la base de datos MySQL como posible Back-end del sistema, ya que debido a su estructura de datos rígida y concisa parecía un buen candidato para soportar la información del sistema. Lastimosamente, más adelante se encontró el problema de segregar y actualizar los datos en tiempo real para cada uno de los dispositivos conectados, por lo cual fue necesario cambiar de tecnología.

El Back-end del aplicativo final fue desarrollado con la API Firebase de Google, que brinda la posibilidad de trabajar con una base de datos almacenada en la nube permitiendo actualizar los datos en tiempo real en cualquier aplicativo o dispositivo que se encuentre sincronizado con esta (garantizando mayor fidelidad de datos), además de ser una base de datos no relacional que facilita el manejo de la información y la hace más flexible, pues todos sus registros son en formato JSON.

Figura 18. Primer diseño de la base de datos

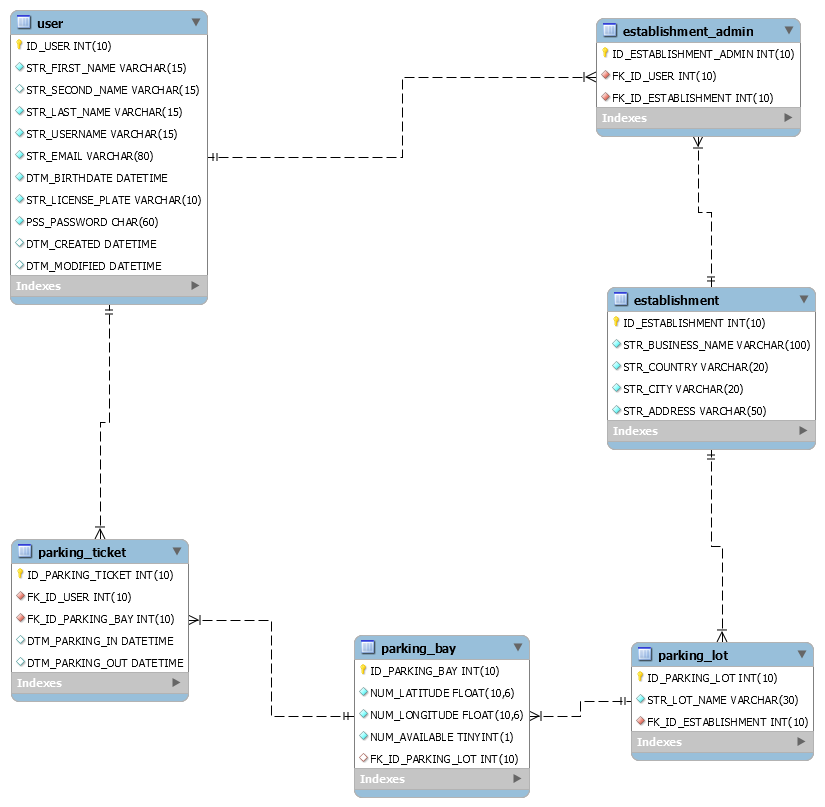
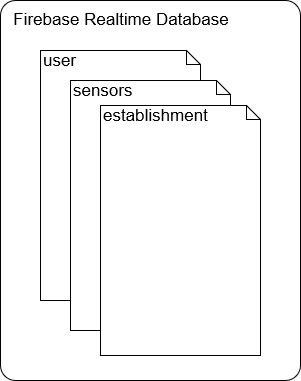


Figura 19. Diseño final de la base de datos

**

La estructura de la base de datos nosql de *Firebase* que utiliza el dispositivo funcional es la siguiente:

* **User:** Age - Email - Username - Name - Sex - Plates [Plate 1 - Plate 2 - Plate 3]
* **Sensors:** Id - Lat - Long - Status
* **Establishment:** Address - City - Country - Id - Lat - Long - Name - ParkingLots - Price - Schedule

(Ver anexo 11.3. Base de datos JSON)

#### 4.4.1.2. Front-end

Para el desarrollo del Front-end del sistema, se estudiaron varias posibilidades en las cuales se analizaron la dificultad de desarrollo, la amplitud con la cual abarcaba los distintos dispositivos y plataformas, el tiempo que existían en el mercado, el soporte que proveían, etc.

Las siguientes son algunas de las opciones que se estudiaron con mayor detenimiento:

* Flutter: Es un SDK para aplicativos móviles desarrollado por Google, con el objetivo de crear aplicaciones nativas para Android e iOS. El mayor inconveniente, es que se encuentra en una versión Beta y no tiene mucha documentación.
* Android Studio SDK y Xcode: Son SDK para desarrollar de forma nativa aplicativos móviles. Fueron desechados, pues sería necesario desarrollar el proyecto en dos lenguajes distintos, ocupando hasta el doble de tiempo, y en el caso particular de Xcode, la licencia de desarrollador tiene un costo de $99 USD.

El diseño Front-end se llevó a cabo finalmente con el framework Ionic, que permite desarrollar aplicativos móviles híbridos y web por medio de las herramientas HTML5, CSS3 y JavaScript.

Este framework fue seleccionado después de estudiarse diferentes opciones de desarrollo tanto nativas como híbridas, evaluando como punto principal, la integración y uso del mapa.

#### 4.4.1.3. Hardware

**Versión 1**

Para el desarrollo del dispositivo de detección de vehículos se analizaron varias alternativas, entre las cuales se encuentran:

* Sensores infrarrojos
* Sensores de contacto
* Sensores de campo magnético
* Sensores de ultrasonido

Después de estudiar cada una de ellas se optó por el uso de los sensores de ultrasonido para la detección del vehículo, ya que su implementación era un poco más fácil en comparación con los demás.

El dispositivo completo de hardware diseñado en la primera versión está compuesto por los siguientes componentes:

* **Chip NODEMCU-ESP8266:** Es el responsable del procesamiento de los datos, así como también de la conexión Wifi y envío de datos a la base de datos de Firebase.
* **Sensor ultrasónico HC SR-04:** Es el dispositivo encargado de la medición de distancia entre el dispositivo y el vehículo.
* **Led blanco:** Indica cuando el dispositivo está funcionando y si hay un obstáculo frente al sensor.
* **Regulador de voltaje ASM125:** Regula la corriente de una fuente de 9V y 12V a 5V para poder dar energía al sistema.
* **Batería 9V:** Fuente de poder del sistema.

**Versión 2**

Al diseñar el protector del dispositivo de hardware surgió el problema de que el sensor de ultrasonido no puede estar cubierto porque generaría error al detectar un vehículo, lo que causaba que el sensor tuviera una parte fuera del protector, arriesgándolo al medio ambiente y los vehículos.

Debido a lo anterior, se cambia el sensor de ultrasonido por un sensor de tiempo de vuelo o infrarrojo, y de igual manera se decide denominar el componente de hardware como PIM o Parking Integrated Module (en español, Módulo Integrado de Parqueo).

El dispositivo completo de hardware diseñado en la segunda versión está compuesto por los siguientes componentes:

* **Chip ESP8266:** Es el responsable del procesamiento de los datos, así como también de la conexión Wifi y envío de datos a la base de datos de Firebase.
* **Sensor de tiempo de vuelo o infrarrojo VL53LOX:** Sensor encargado de la detección del vehículo.
* **Step Down AC-DC:** Este dispositivo es un convertidor de voltaje de corriente alterna a corriente directa y al mismo tiempo un regulador de voltaje.

## 4.5. Modelo de negocio

Para poder lograr la sustentabilidad económica de este proyecto, se crearon ciertos escenarios, en los cuales se describen las posibles maneras de comercialización. Estos son:

**Escenario 1.**

El primer escenario propuesto es el de plataforma como un servicio o PaaS (por sus siglas en inglés, Platform as a Service), de tal forma que se realice un cobro mensual a los establecimientos, dependiendo de la cantidad de espacios de parqueo que estos posean o el tráfico de red que generen.

**Escenario 2.**

Como segunda propuesta se plantea la venta de licencias para los establecimientos, permitiéndoles a estos realizar pagos anuales que contemplen tanto el aplicativo como los detectores de vehículos que estos requieran para suplir todos los espacios de parqueo.

## 4.6. Marketing

Como tarea adicional, se desarrollaron, de manera paralela e independiente al avance del proyecto, ciertos diseños y herramientas visuales para poder dar a conocer al público general el trabajo que se está llevando a cabo en este proyecto, las cuales se describen a continuación:

* Diseño y desarrollo de una página web para informar al público sobre el funcionamiento básico de la app y el equipo involucrado. (http://www.parqueameapp.com/)
* Diseño e impresión de dos posters y/o volantes para atraer al público de manera más llamativa.

Figura 20. Volante promocional



Figura 21. Poster promocional



# 5. COSTOS Y PROYECCIÓN ECONÓMICA

Tabla 2. Costos para el desarrollo del proyecto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Producto | Descripción | Precio por unidad | Cantidad comprada | Total por producto |
| NODEMCU-ESP8266 | Chip para la conexión wifi y el manejo de los datos. | $38,000 | 6 | $228,000 |
| Sensor ultrasónico HC SR-14 | Sensor de ultrasonido para medir la distancia del objeto | $10,500 | 6 | $63,000 |
| Regulador de voltaje ASM125 | Reguladores de voltaje (12V o 9V a 5V) | $7,000 | 6 | $42,000 |
| Batería 9V | Batería alcalina para darle energía a los prototipos. | $7,000 | 6 | $42,000 |
| Protoboard | Para hacer pruebas y modelos del mecanismo. | $11,500 | 1 | $11,500 |
| Jumpler cables | Para hacer las conexiones en la board. | $2,300 | 2 | $4,600 |
| Cable broche | Cables de corriente para las baterías. | $600 | 6 | $3,600 |
| LED blanco | Parte del circuito que indica cuándo un objeto obstruye el sensor. | $200 | 6 | $1,200 |
| Placa para soldar | Placas para realizar las soldaduras de los prototipos. | $6,300 | 3 | $18,900 |
| Dominio web | Dominio público web para acceder a la página promocional | $42,495 | 1 | $42,495 |
| **Gasto total** | | | | **$457,295** |

Se detallan los costos obtenidos en el desarrollo del proyecto.

Tabla 3. Costos por diseño de sistema de detección

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Producto | 1 unidad | 10 unidades | 100 unidades | 1000 unidades |
| NODEMCU-ESP8266 | $38,000 | $380,000 | $3,800,000 | $38,000,000 |
| Sensor ultrasónico HC SR-14 | $10,500 | $105,000 | $1,050,000 | $10,500,000 |
| Regulador de voltaje ASM125 | $7,000 | $70,000 | $700,000 | $7,000,000 |
| LED blanco | $200 | $2,000 | $20,000 | $200,000 |
| Placa para soldar | $6,300 | $63,000 | $630,000 | $6,300,000 |
| **Total** | $62,000 | $620,000 | $6,200,000 | $62,000,000 |

Estos costos no contemplan la carcasa ni la soldada del dispositivo.

# 6. RESULTADOS OBTENIDOS

## 6.1. Pruebas funcionales

Conforme terminaba cada ciclo o sprint, se ejecutaron pruebas funcionales para asegurar el comportamiento deseado del aplicativo.

Los siguientes son casos de prueba funcionales ejecutados en el aplicativo.

**CP001 - Registrar un usuario nuevo en el aplicativo**

Descripción: Registrar un usuario nuevo en el aplicativo, haciendo uso de la funcionalidad de registro.

Precondiciones:

* El usuario por registrar no debe estar almacenado en la base de datos del aplicativo.
* No debe haber una sesión iniciada en el aplicativo.

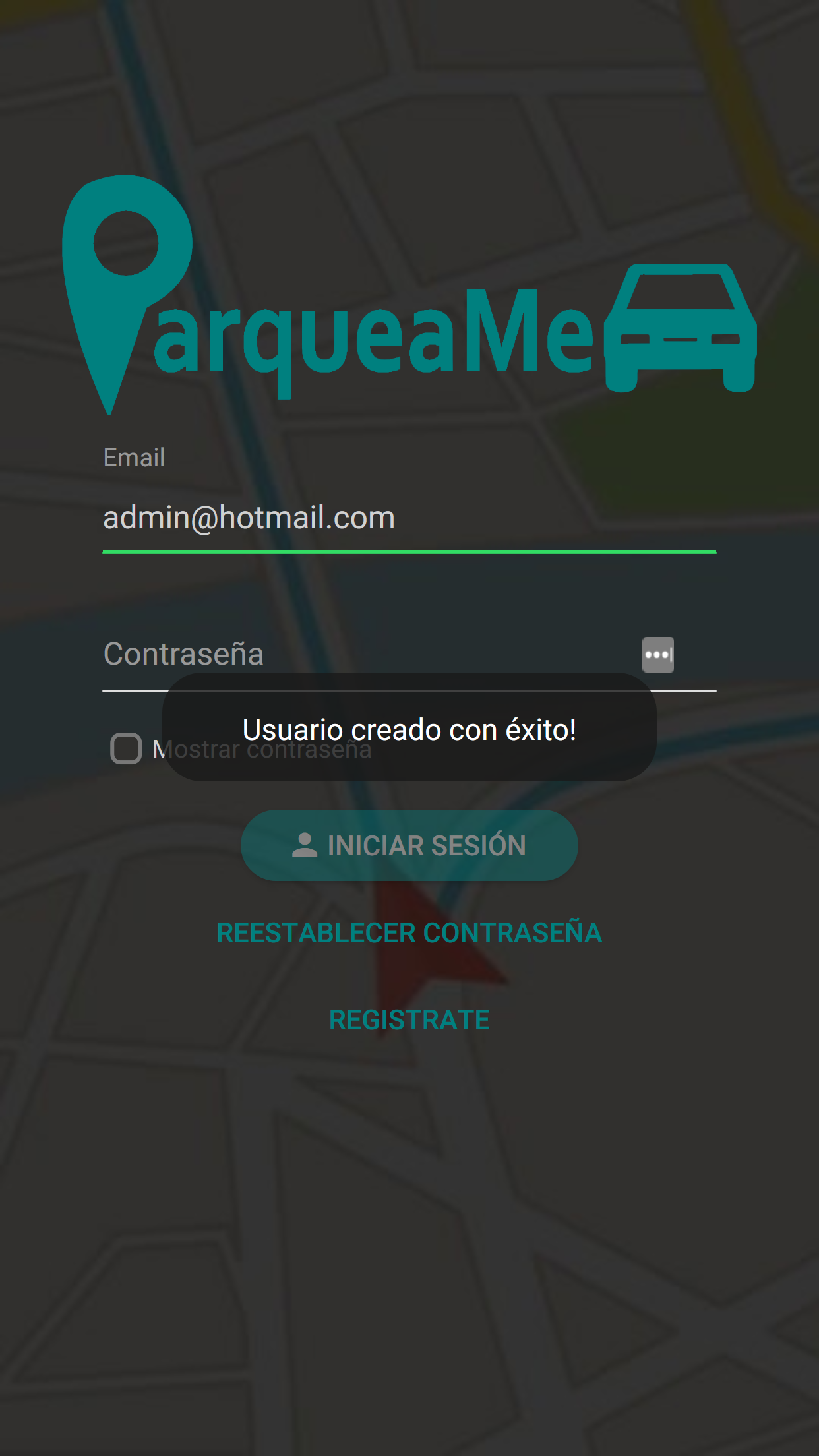
Paso a paso:

1. Abrir el aplicativo móvil.
2. Seleccionar la opción Registrate.
3. Diligenciar los campos obligatorios.
4. Seleccionar la opción Registrame!

Resultado esperado:

El registro se hizo con éxito, el usuario ya ingresar al aplicativo.

Figura 22. Registro de usuario en el aplicativo



**CP002 - Registrar un usuario existente en el aplicativo**

Descripción: Intentar registrar un usuario que ya existe en el aplicativo.

Precondiciones:

* El usuario que se intenta registrar ya debe estar creado en el aplicativo.
* No debe haber una sesión iniciada en el aplicativo.

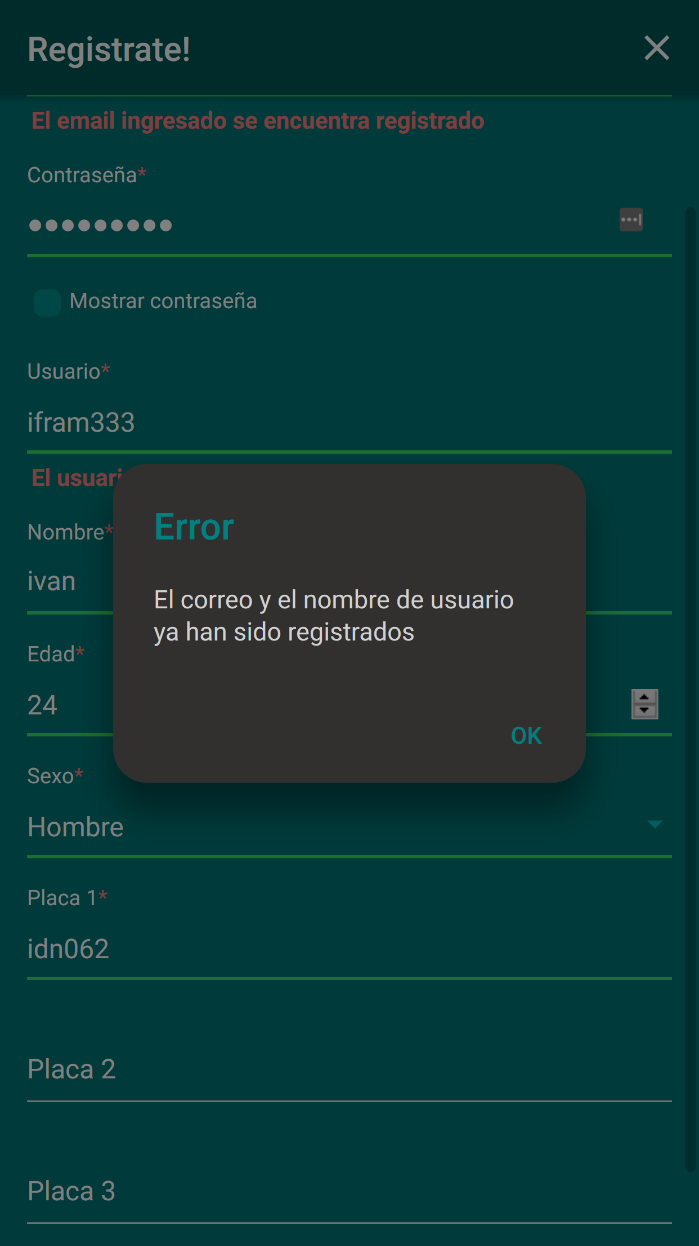
Paso a paso:

1. Abrir el aplicativo móvil.
2. Seleccionar la opción Registrate.
3. Diligenciar los campos obligatorios.
4. Seleccionar la opción Registrame!

Resultado esperado:

El aplicativo indica que el usuario ya está registrado.

Figura 23. Registro de usuario ya creado



**CP003 - Iniciar sesión en el aplicativo**

Descripción: Iniciar sesión en el aplicativo con un usuario ya registrado.

Precondiciones:

* El usuario debe estar registrado en el aplicativo.
* No debe haber una sesión iniciada en el aplicativo.

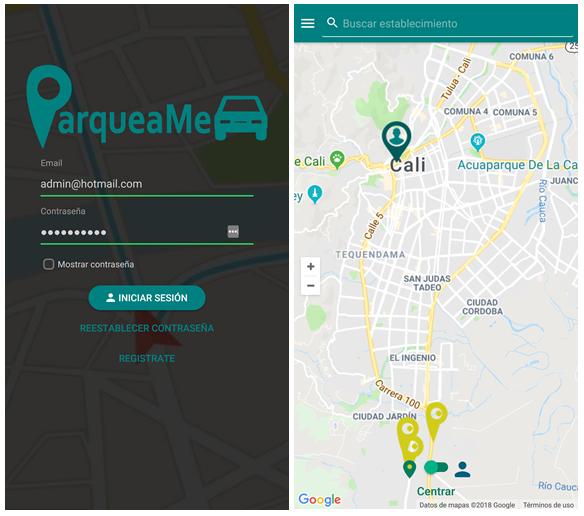
Paso a paso:

1. Abrir el aplicativo móvil.
2. Diligenciar los campos de inicio de sesión.
3. Seleccionar la opción Iniciar Sesión.

Resultado esperado:

El aplicativo inicia sesión mostrando la pantalla principal.

Figura 24. Inicio de sesión



**CP004 - Iniciar sesión en el aplicativo con un usuario no existente**

Descripción: Intentar iniciar sesión con un usuario que no está registrado.

Precondiciones:

* El usuario no debe estar registrado en el aplicativo.
* No debe haber una sesión iniciada en el aplicativo.

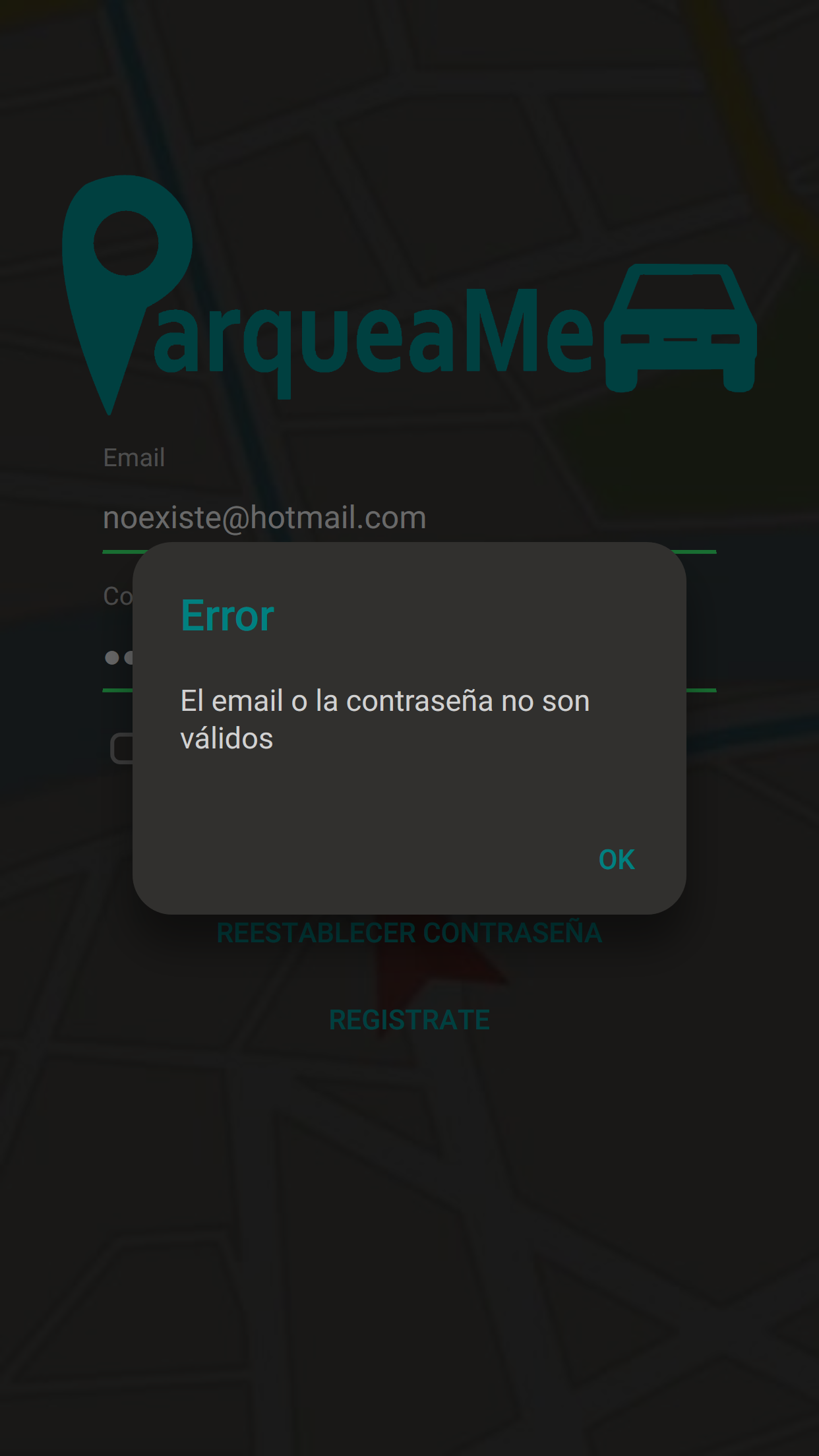
Paso a paso:

1. Abrir el aplicativo móvil.
2. Diligenciar los campos de inicio de sesión.
3. Seleccionar la opción Iniciar Sesión.

Resultado esperado:

El aplicativo muestra un mensaje y no inicia sesión en el aplicativo.

Figura 25. Inicio de sesión con un usuario no registrado

**

**CP005 - Iniciar sesión en el aplicativo con una contraseña incorrecta**

Descripción: Intentar iniciar sesión con un usuario que existe, pero con una contraseña que no corresponde al usuario.

Precondiciones:

* El usuario debe estar registrado en el aplicativo.
* No debe haber una sesión iniciada en el aplicativo.

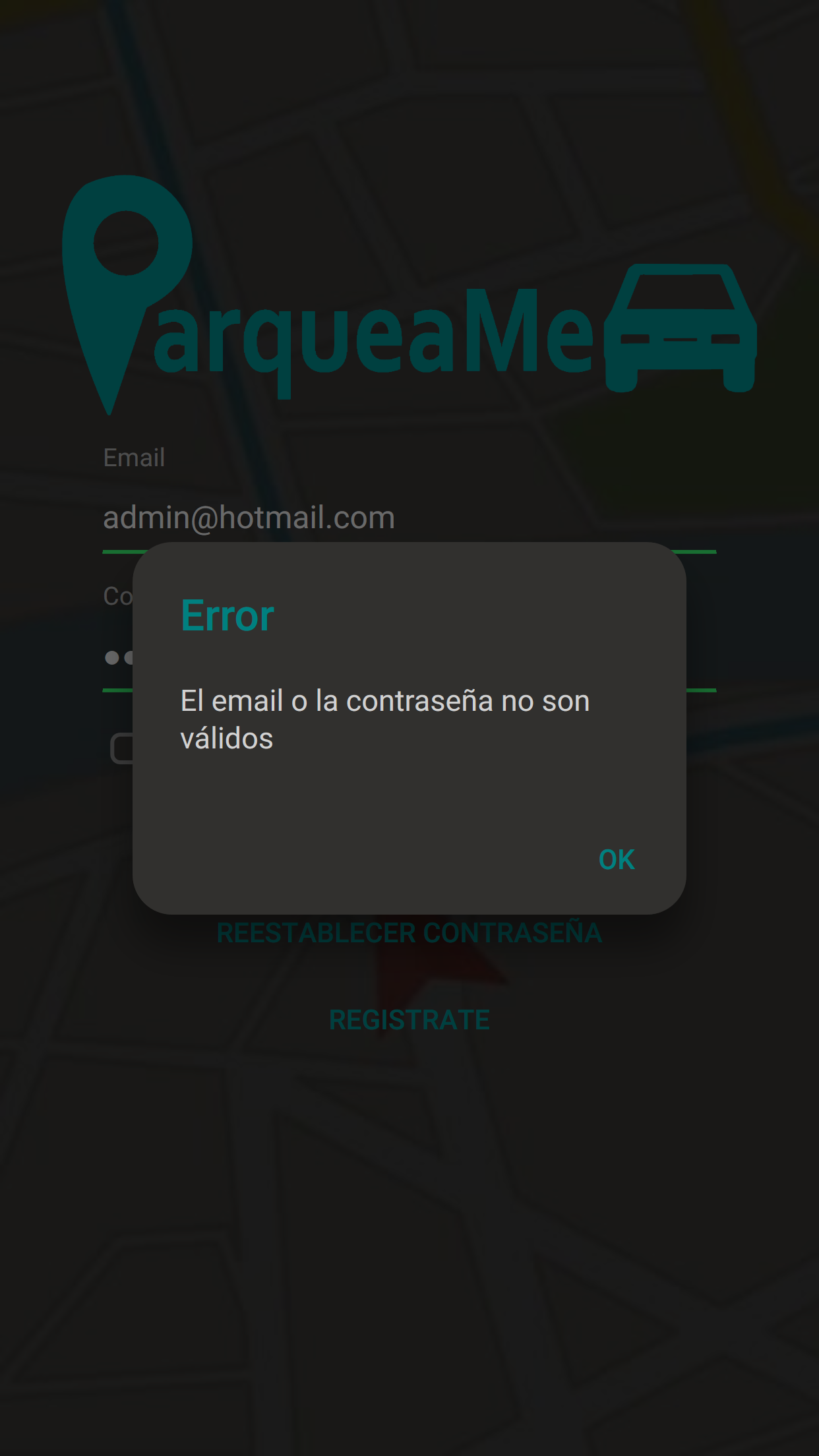
Paso a paso:

1. Abrir el aplicativo móvil.
2. Diligenciar el correo electrónico del usuario.
3. Diligenciar una contraseña que no corresponda al correo electrónico del usuario.
4. Seleccionar la opción Iniciar Sesión.

Resultado esperado:

El aplicativo muestra un mensaje y no inicia sesión en el aplicativo.

Figura 26. Iniciar sesión con contraseña inválida

**

**CP006 - Restablecer contraseña de una cuenta**

Descripción: Intentar iniciar sesión con un usuario que existe, pero con una contraseña que no corresponde al usuario.

Precondiciones:

* El usuario debe estar registrado en el aplicativo.
* No debe haber una sesión iniciada en el aplicativo.

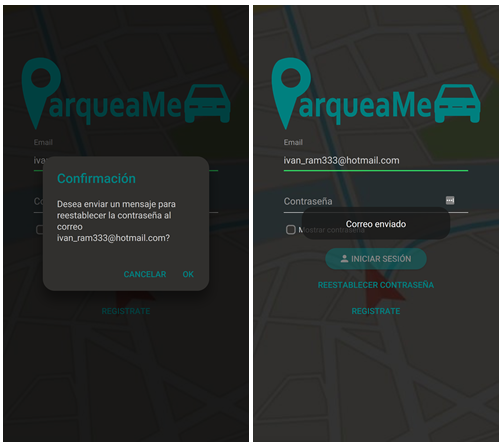
Paso a paso:

1. Abrir el aplicativo móvil.
2. Diligenciar el correo electrónico del usuario.
3. Seleccionar la opción Restablecer Contraseña.
4. Seleccionar la opción Ok.
5. Seleccionar el link enviado por el correo electrónico.
6. Diligenciar la nueva contraseña.
7. Seleccionar la opción Guardar.

Resultado esperado:

El aplicativo muestra un mensaje de información indicando si el usuario quiere recibir un correo electrónico con la información para cambiar la contraseña.

Figura 27. Restablecer contraseña



**CP007 - Consultar un establecimiento por nombre**

Descripción: Consultar un establecimiento ingresando su nombre en el buscador.

Precondiciones:

* Se debe haber iniciado sesión en el aplicativo.
* El usuario debe estar en la pantalla principal del aplicativo.

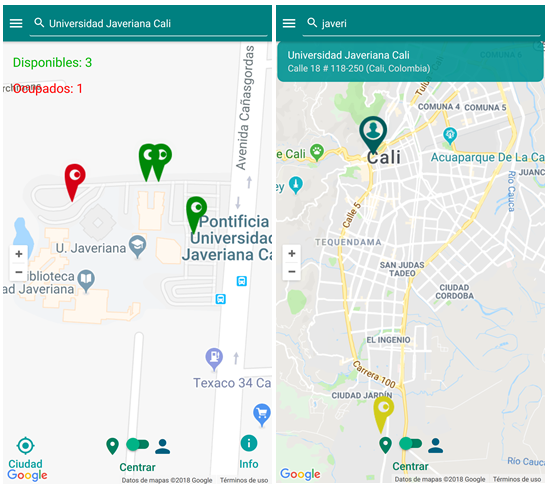
Paso a paso:

1. Seleccionar la barra de búsqueda.
2. Diligenciar el nombre del establecimiento a consultar.
3. Seleccionar el establecimiento consultado.

Resultado esperado:

El aplicativo muestra el establecimiento y los parqueaderos disponibles y ocupados.

Figura 28. Consultar establecimiento por nombre



**CP008 - Consultar un establecimiento seleccionándolo**

Descripción: Consultar un establecimiento seleccionándolo desde el mapa.

Precondiciones:

* Se debe haber iniciado sesión en el aplicativo.
* El usuario debe estar en la pantalla principal del aplicativo.

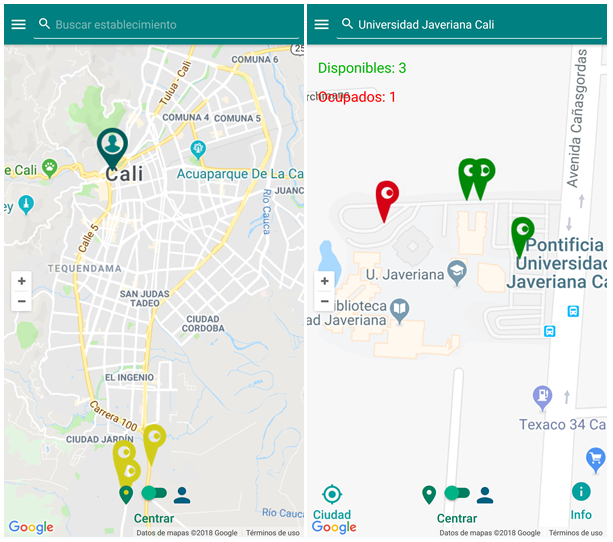
Paso a paso:

1. Seleccionar el establecimiento a consultar.

Resultado esperado:

El aplicativo muestra el establecimiento y los parqueaderos disponibles y ocupados.

Figura 29. Consultar establecimiento seleccionándolo



**CP009 - Consultar establecimiento por voz**

Descripción: Consultar un establecimiento haciendo uso del micrófono.

Precondiciones:

* Se debe haber iniciado sesión en el aplicativo.
* El usuario debe estar en la pantalla principal del aplicativo.

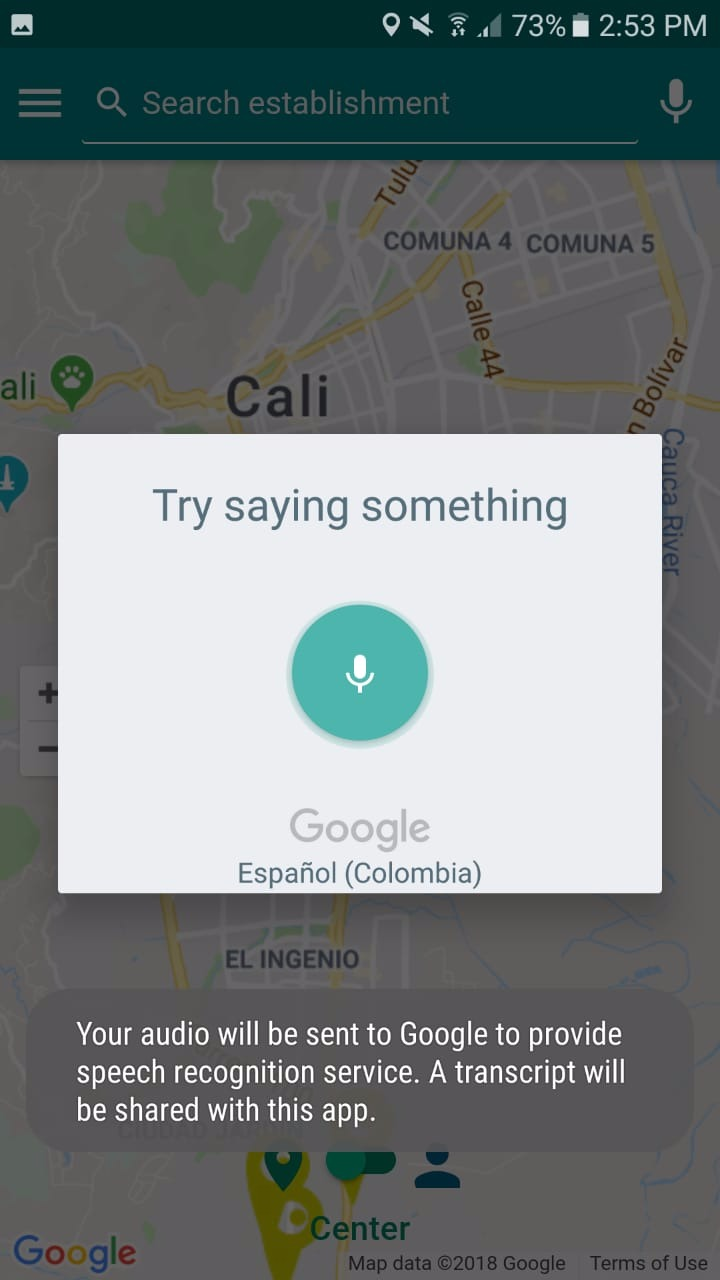
Paso a paso:

1. Seleccionar la opción de Micrófono.
2. Pronunciar el nombre del establecimiento a consultar.

Resultado esperado:

El aplicativo debe reconocer el nombre pronunciado y mostrar el establecimiento si existe.

Figura 30. Funcionalidad de micrófono



**CP010 - Guiar al usuario a un parqueadero disponible**

Descripción: Mostrar al usuario la ruta desde su posición actual hasta uno de los parqueaderos disponibles en cualquier establecimiento.

Precondiciones:

* Se debe haber iniciado sesión en el aplicativo.
* El usuario debe estar en la pantalla principal del aplicativo.
* El parqueadero por seleccionar debe estar disponible.

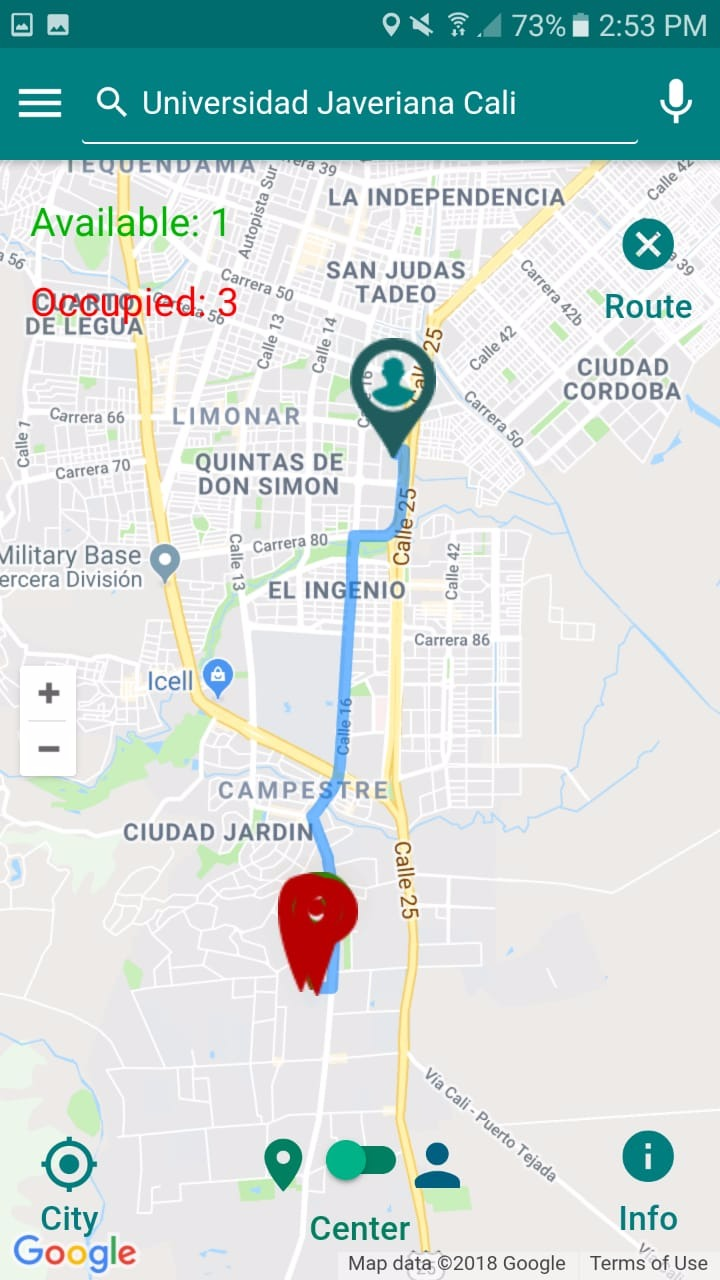
Paso a paso:

1. Seleccionar un parqueadero disponible.

Resultado esperado:

El aplicativo debe mostrar el camino desde la posición del usuario hasta el parqueadero seleccionado.

Figura 31. Ruta desde la posición del usuario

**

**CP011 - Cambiar el estado de un parqueadero a ocupado**

Descripción: Cambiar el estado de un parqueadero de libre a ocupado.

Precondiciones:

* Se debe haber iniciado sesión en el aplicativo.
* El usuario debe estar en la pantalla principal del aplicativo.
* Debe haber, por lo menos, un parqueadero desocupado.

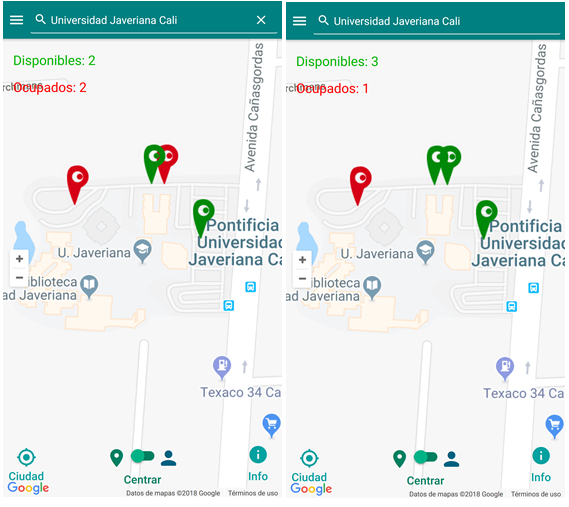
Paso a paso:

1. Cambiar el estado del parqueadero en la base de datos.

Resultado esperado:

El aplicativo debe actualizar la vista de los parqueaderos de tal forma que se identifique cuando un parqueadero cambia de libre a desocupado.

Figura 32. Cambio de estado del parqueadero a ocupado

**

**CP012 - Cambiar el estado de un parqueadero a desocupado**

Descripción: Cambiar el estado de un parqueadero de libre a ocupado.

Precondiciones:

* Se debe haber iniciado sesión en el aplicativo.
* El usuario debe estar en la pantalla principal del aplicativo.
* Debe haber, por lo menos, un parqueadero ocupado.

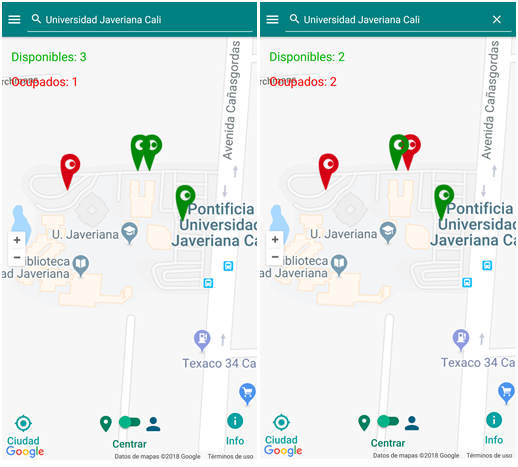
Paso a paso:

1. Cambiar el estado del parqueadero en la base de datos.

Resultado esperado:

El aplicativo debe actualizar la vista de los parqueaderos de tal forma que se identifique cuando un parqueadero cambia de ocupado a desocupado.

Figura 33. Cambio de estado del parqueadero a desocupado



**CP013 - Modificar el perfil de un usuario**

Descripción: Modificar el perfil de un usuario desde la aplicación.

Precondiciones:

* Se debe haber iniciado sesión en el aplicativo.
* El usuario debe estar en la pantalla principal del aplicativo.

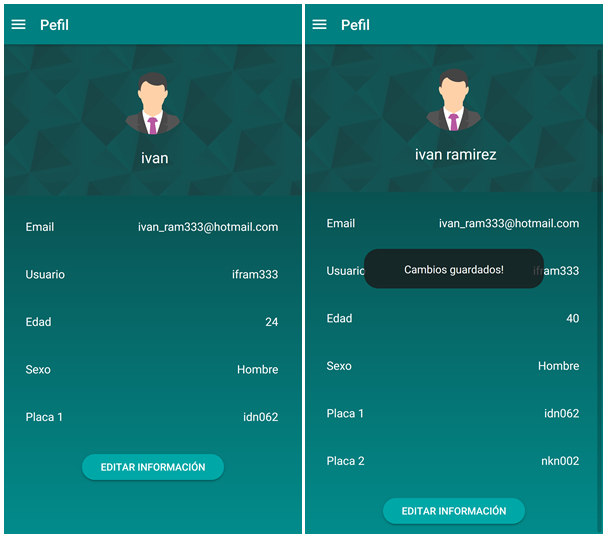
Paso a paso:

1. Seleccionar la opción de Menú.
2. Seleccionar la opción Perfil.
3. Seleccionar la opción Editar Información.
4. Cambiar información del usuario.
5. Seleccionar la opción Guardar Cambios.

Resultado esperado:

El aplicativo debe actualizar la información del usuario.

Figura 34. Modificar información



**CP014 - Cambiar el tema del aplicativo**

Descripción: Cambiar los colores del aplicativo con los temas predefinidos.

Precondiciones:

* Se debe haber iniciado sesión en el aplicativo.
* El usuario debe estar en la pantalla principal del aplicativo.

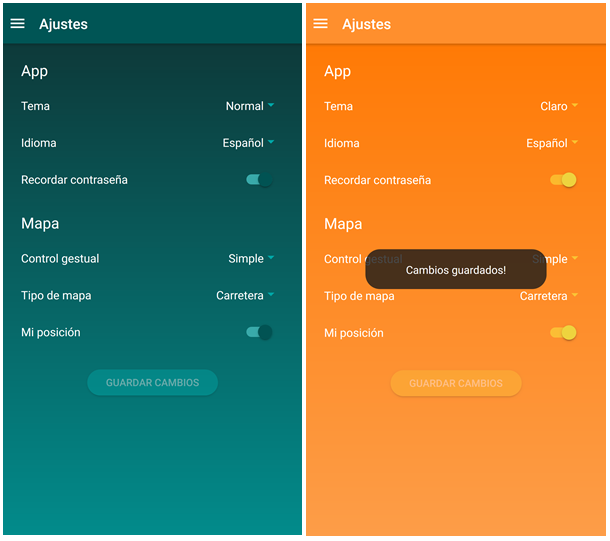
Paso a paso:

1. Seleccionar la opción de Menú.
2. Seleccionar la opción Ajustes.
3. Seleccionar la opción Temas.
4. Seleccionar un tema predefinido.
5. Seleccionar la opción Guardar Cambios.

Resultado esperado:

El aplicativo debe cambiar los colores según el tema seleccionado.

Figura 35. Cambiar el tema del aplicativo



**CP015 - Cambiar el idioma del aplicativo**

Descripción: Cambiar el idioma del aplicativo.

Precondiciones:

* Se debe haber iniciado sesión en el aplicativo.
* El usuario debe estar en la pantalla principal del aplicativo.

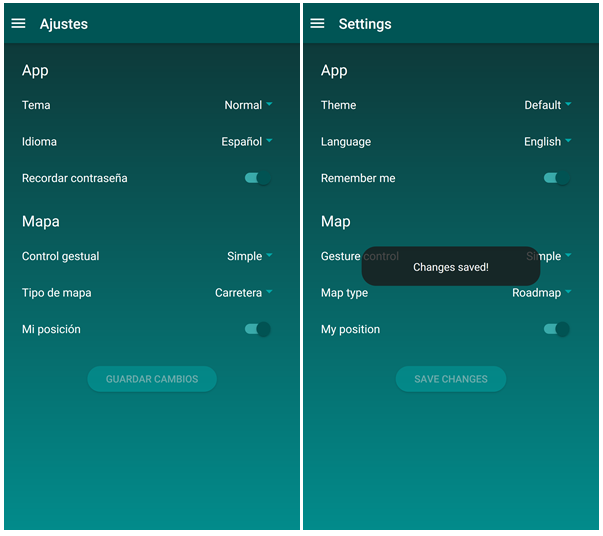
Paso a paso:

1. Seleccionar la opción de Menú.
2. Seleccionar la opción Ajustes.
3. Seleccionar la opción Idioma.
4. Seleccionar un idioma.
5. Seleccionar la opción Guardar Cambios.

Resultado esperado:

El aplicativo debe cambiar el idioma del aplicativo.

Figura 36. Cambiar el idioma del aplicativo



**CP016 - Cambiar la vista del mapa**

Descripción: Cambiar el idioma del aplicativo.

Precondiciones:

* Se debe haber iniciado sesión en el aplicativo.
* El usuario debe estar en la pantalla principal del aplicativo.

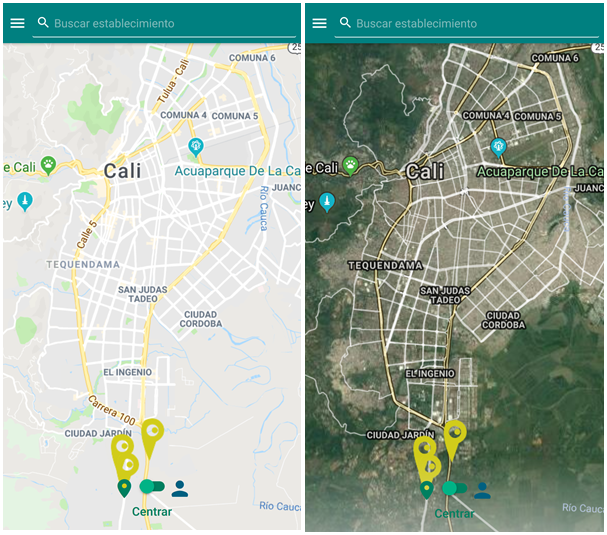
Paso a paso:

1. Seleccionar la opción de Menú.
2. Seleccionar la opción Ajustes.
3. Seleccionar la opción Tipo de mapa.
4. Seleccionar una opción de tipo de vista de mapa.
5. Seleccionar la opción Guardar Cambios.

Resultado esperado:

El aplicativo debe cambiar el tipo de vista del mapa.

Figura 37. Cambiar el tipo de mapa

****

## 6.2. Pruebas de usuario

Al finalizar el desarrollo, se llevaron a cabo pruebas de campo con los que podrían ser potencialmente usuarios finales. Estas pruebas de campo se dividieron en dos partes y fueron grabadas bajo los permisos de los mismos usuarios (Ver anexo 11.3. Grabaciones de pruebas de usuarios).

La primera parte de la prueba consistió en que los usuarios hicieran uso supervisado del aplicativo por un tiempo máximo de dos minutos, dando como única información el objetivo general por el que se desarrolló el aplicativo.

En la segunda parte de la prueba, se les daba una explicación de cada funcionalidad que permitía el aplicativo y se hacía un pequeño ejercicio de retroalimentación sobre lo que ellos experimentaron y se preguntaba además sugerencias adicionales que desearan aportar.

El resultado de las pruebas de usuario concluyó en lo siguiente:

* La aplicación debería cargar automáticamente la posición del usuario y no mostrar el mapa de la ciudad de Cali.
* La aplicación dejaba de funcionar al momento de cambiar de idiomas más de dos veces en una sola ejecución.
* La aplicación debería tener seccionada las zonas de los parqueaderos, de tal forma que facilite más el encontrar un parqueadero disponible.

Pese a algunas fallas con el prototipo, la mayoría de las entrevistas que fueron realizadas terminaron de manera satisfactoria, con varios de los participantes afirmando que esperaban ansiosos el lanzamiento de la aplicación.

Figura 38. Pruebas de usuario



## 6.3. Pruebas del dispositivo de detección

Según las especificaciones técnicas del sensor ultrasónico HC SR-14, se puede calcular distancia entre 2 y 450 centímetros. Las pruebas se realizaron con ajustes de medición para 10, 20, 40 y 70 centímetros; obteniendo resultados positivos al 100%.

El consumo de energía del dispositivo en conjunto fue calculado con una fuente de poder de 5V y un multímetro Fluke 117. Los resultados se encuentran en la siguiente tabla (Tabla 3).

Tabla 4. Consumo de energía del dispositivo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Estado del dispositivo | Descripción | 1 unidad | 100 unidades | 1000 unidades |
| Deep Sleep | El dispositivo detector entra en estado de suspensión y se reactiva cada 10 segundos. | 40 mAh | 4000 mAh | 40000 mAh |
| Funcionamiento continuo | El dispositivo detector está activo siempre sin descanso. | 70 mAh | 7000 mAh | 70000 mAh |
| Funcionamiento con delay de 7 segundos | El dispositivo detector está activo siempre, pero recolecta y envía los datos cada 7 segundos. | 52 mAh | 5200 mAh | 52000 mAh |

Los consumos de energía anteriores fueron calculados con una fuente de 5V. mAh = miliamperio por hora.

Debido a los resultados obtenidos por los cálculos realizados, se aconseja que los establecimientos que vayan a utilizar esta solución alimenten los componentes de hardware por medio de la red eléctrica, teniendo en cuenta que los dispositivos deben ser alimentados con corriente continua de 5V, lo cual hace necesaria la existencia de un convertidor de AC a DC.

# 7. CONCLUSIONES

Gracias a los resultados obtenidos por las pruebas realizadas, se puede concluir que la investigación, el diseño y el desarrollo del sistema ha sido un éxito, ya que se lograron los resultados que se esperaban en cuanto a la satisfacción de los 12 usuarios entrevistados a lo largo del proyecto, además del cumplimiento de la mayoría de los requisitos establecidos al iniciar el proyecto.

A pesar de esto, no se lograron dos objetivos de mediana importancia durante el desarrollo del proyecto, los cuales se han mencionado en trabajos futuros para poder ser implementados en una versión posterior. Estos objetivos son:

* La entrega de informes o información relevante a los establecimientos.
* Hacer al componente de hardware resistente al clima y al peso de los vehículos, en caso de que llegase a ocurrir un accidente con los mismos.

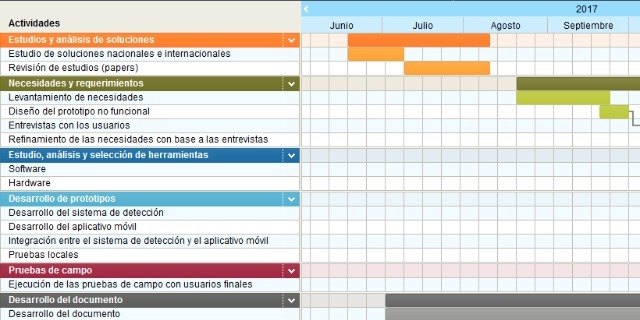
Por los cálculos realizados en la sección de Costos y Producción económica, se puede asegurar que tanto establecimientos grandes como pequeños pueden hacer uso de esta solución si así lo desean, pues esta solución prueba ser lo suficientemente modular para ser implementada sin ningún problema en la gran mayoría de los establecimientos, ya que no requiere una gran cantidad de infraestructura instalada.

# 8. TRABAJOS FUTUROS

A continuación, se listan las mejoras y adecuaciones que se desean realizar a futuro en el proyecto, con el objetivo de brindar una experiencia más completa para los usuarios finales y poder entregar una plataforma con mayor cobertura y estabilidad.

* Desplegar el aplicativo para iOS, windows phone y web.
* Servicio de *Chat*, con el cual se puedan comunicar varios usuarios entre ellos y con el establecimiento para resolver problemas que se puedan llegar a generar en el mismo (Choques, daños a propiedad, etc).
* *Dashboard* para el establecimiento, de tal forma que estos puedan recibir informes e información relevante acerca de los usuarios que visitan el mismo (Requiere permiso de recolección de datos por parte de los usuarios).
* Desarrollo de una API comercial para integrar la infraestructura propia del establecimiento con el aplicativo.
* Mejora del componente de hardware para que resiste riesgos medioambientales (lluvia, sobrecarga, etc).
* Adaptación del sistema para soportar vehículos de dos ruedas.

# 9. CRONOGRAMA DE ACTIVIDADES





# 10. BIBLIOGRAFÍA Y REFERENCIAS

*Mejoran las ventas de carros nuevos en Cali este año*. Cali, C. (2018). *Elpais.com.* Retrieved 12 May 2018, from http://www.elpais.com.co/economia/mejoran-las-ventas-de-carros-nuevos-en-cali-este-ano.html

*Eco parking revoluciona la forma de ‘parquear’*. (2018). *Dinero.com*. Retrieved 13 April 2018, from http://www.dinero.com/empresas/articulo/eco-parking-revoluciona-forma-parquear/168475

*Implementan parqueadero inteligente en centro comercial de Medellín*. (2017). *El Universal Cartagena*. Retrieved 5 February 2017, from http://www.eluniversal.com.co/tecnologia/implementan-parqueadero-inteligente-en-centro-comercial-de-medellin-150911

Propst, J., Poole, K., & Hallstrom, J. (2012). An embedded sensing approach to monitoring parking lot occupancy. *Proceedings Of The 50Th Annual Southeast Regional Conference On - ACM-SE '12*. http://dx.doi.org/10.1145/2184512.2184584

Mathew, S., Atif, Y., Sheng, Q., & Maamar, Z. (2013). Building sustainable parking lots with the Web of Things. *Personal And Ubiquitous Computing*, *18*(4), 895-907. http://dx.doi.org/10.1007/s00779-013-0694-7

Lambrinos, L. & Dosis, A. (2013). Applying mobile and internet of things technologies in managing parking spaces for people with disabilities. *Proceedings Of The 2013 ACM Conference On Pervasive And Ubiquitous Computing Adjunct Publication - Ubicomp '13 Adjunct*. http://dx.doi.org/10.1145/2494091.2494162

Herrera-Quintero, L., Maciá-Pérez, F., Marcos-Jorquera, D., & Gilart-Iglesias, V. (2012). Wireless sensor networks and service-oriented architecture, as suitable approaches to be applied into ITS. *Proceedings Of The 6Th Euro American Conference On Telematics And Information Systems - EATIS '12*. http://dx.doi.org/10.1145/2261605.2261650

Vora, A., Kumar, M., & Srinivasa, K. (2014). Low Cost Internet of Things based Vehicle Parking Information System. *Proceedings Of The 6Th IBM Collaborative Academia Research Exchange Conference (I-CARE) On I-CARE 2014 - I-CARE 2014*. http://dx.doi.org/10.1145/2662117.2662133

Elaouad, S., Benmakhlouf, S., Tobaji, N., Dmini, M., & Salih Alj, Y. (2015). Car parking management system using AMR-sensor technology. *2015 International Conference On Electrical And Information Technologies (ICEIT)*. http://dx.doi.org/10.1109/eitech.2015.7162985

Banerjee, S., Choudekar, P., & Muju, M. (2011). Real time car parking system using image processing. *2011 3Rd International Conference On Electronics Computer Technology*. http://dx.doi.org/10.1109/icectech.2011.5941663

Zheng, D., Zhang, X., Shu, Y., Fang, C., Cheng, P., & Chen, J. (2015). iParking: An intelligent parking system for large parking lots. *2015 IEEE Conference On Computer Communications Workshops (INFOCOM WKSHPS)*. http://dx.doi.org/10.1109/infcomw.2015.7179331

Bachani, M., Qureshi, U., & Shaikh, F. (2016). Performance Analysis of Proximity and Light Sensors for Smart Parking. *Procedia Computer Science*, *83*, 385-392. http://dx.doi.org/10.1016/j.procs.2016.04.200

Basavaraju, S. (2015). Automatic Smart Parking System using Internet of Things (IOT). *International Journal Of Scientific And Research Publications*, *5*(12).

Khedkar, S. and Thube, S. (2017). *Real Time Databases for Applications*. [online] Irjet.net. Available at: https://irjet.net/archives/V4/i6/IRJET-V4I6401.pdf [Accessed 19 Apr. 2018].

# 11. ANEXOS

Anexo 1. Preguntas de las entrevistas

## 11.1. Preguntas de las entrevistas

Las siguientes son las preguntas realizadas en las entrevistas a los usuarios.

* ¿Qué vehículo conduce?
* ¿Qué tan frecuentemente conduce usted?
* ¿Por qué áreas de la ciudad se moviliza usted?
* ¿Qué tal ha sido su experiencia manejando?
* ¿Le parece agradable conducir en Cali?
* ¿Cómo afronta las horas pico?
* ¿Suele llevar pasajeros en su vehículo?
* ¿Conduce por largos trayectos?
* ¿Cómo se entretiene usted mientras conduce?
* ¿Parquea usted en reversa o de frente?
* ¿Nos puede regalar una brece descripción de cómo es su día desde que sale de su vivienda hasta que vuelve a ella? (En cuanto a conducir y tráfico).
* ¿Cuántos vehículos posee? Si cuenta con más de uno, preguntar ¿Mejora la experiencia al conducir un vehículo u otro?
* ¿Cuándo va a salir a comer/bailar/cine usa su vehículo? Si la respuesta es positiva, preguntar ¿Parquea usted en la calle o en un establecimiento? ¿Cómo es su experiencia de parquear en un lugar u otro?
* ¿Qué lugares frecuenta cuando conduce? ¿Qué tal le parece aparcar en estos lugares? (Le afecta el lugar, el clima, la hora, la actitud).
* ¿En qué establecimientos le gusta aparcar? ¿Por qué?
* ¿En qué difiere el tener que parquear en horas pico y los fines de semana?
* ¿Qué tal le parece el costo de los parqueaderos? (Unicentro, Pontificia Universidad Javeriana Cali, Chipichape)
* ¿Cómo cree usted que se podría mejorar la experiencia de parqueo?
* Si existiera una aplicación para celulares que le ayude a encontrar un espacio de parqueadero ¿Estaría dispuesto a utilizarla?
* ¿Cómo le gustaría que fuera la aplicación?
* ¿Le parece justificable el pago del parqueadero si los establecimientos implementaran esta aplicación?
* ¿Estaría dispuesto a que los establecimientos recolecten información para mejorar su experiencia en ellos? (Sexo, edad, hora de parqueo)

## 11.2. Grabaciones de las entrevistas

Anexo 2. Grabaciones de las entrevistas

Se adjuntan las URLs correspondientes a los audios de las entrevistas realizadas a población estudiantes de la Pontificia Universidad Javeriana Cali y Universidad Icesi Cali.

* Entrevista #1 - https://goo.gl/1oCKKC
* Entrevista #2 - https://goo.gl/tMkb5f
* Entrevista #3 - https://goo.gl/opz4yK
* Entrevista #4 - https://goo.gl/uFfV3X
* Entrevista #5 - https://goo.gl/CtcvUf
* Entrevista #6 - https://goo.gl/vCw939
* Entrevista #7 - https://goo.gl/4gMSBM

## 11.3. Grabaciones de pruebas de usuario

Anexo 3. Grabaciones de pruebas de usuario

Se adjuntan las URLs correspondientes a los audios de las grabaciones realizadas en las pruebas de usuarios con el prototipo funcional final.

* Grabación #1 - https://goo.gl/VrmCAF
* Grabación #2 - https://goo.gl/NkjZAX
* Grabación #3 - https://goo.gl/mK7gkb
* Grabación #4 - https://goo.gl/sTAwK4
* Grabación #5 - https://goo.gl/Kt5Vo6

## 11.4. Base de datos JSON

Anexo 4. Base de datos JSON

Diseño de la base de datos utilizada para el proyecto.

{

"establishment":[

{

"address":"###",

"city":"###",

"country":"###",

"id":0,

"lat":###,

"long":###,

"name":"###",

"parkingLots":###,

"price":"###",

"schedule":"###"

}

],

"sensors":[

{

"id":###,

"name":"###",

"parkingLot":[

{

"id":###,

"lat":###,

"long":###,

"status":###

}

]

}

],

"user":{

"####":{

"age":###,

"email":"###@###.com",

"name":"###",

"plates":[

{

"number":"###"

}

],

"sex":"###",

"username":"###"

}

}

}