



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

## SISTEMA DE GESTIÓN DE RESERVA Y UTILIZACIÓN DE PARQUEOS

**Brandon Bredly Alvarez López**

**Joel Daniel Esaú Pérez Camey**

Asesorado por el Ing. Carlos Estuardo Coy de León

Guatemala, octubre de 2023



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**SISTEMA DE GESTIÓN DE RESERVA Y UTILIZACIÓN DE PARQUEOS**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**BRANDON BREDLY ALVAREZ LÓPEZ**  
ASESORADO POR LA MSC. ING. CARLOS COY

AL CONFERÍRSELES EL TÍTULO DE  
**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, OCTUBRE DE 2023



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**SISTEMA DE GESTIÓN DE RESERVA Y UTILIZACIÓN DE PARQUEOS**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**JOEL DANIEL ESAÚ PÉREZ CAMEY**  
ASESORADO POR LA MSC. ING. CARLOS COY

AL CONFERÍRSELES EL TÍTULO DE  
**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, OCTUBRE DE 2023



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO A.I.	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Álvaro Giovanni Longo Morales
EXAMINADOR	Ing. Alvaro Obrayan Hernández García
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**JOEL DANIEL ESAÚ PÉREZ CAMEY**





UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO A.I.	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Luis Fernando Espino Barrios
EXAMINADOR	Ing. Marlon Francisco Orellana López
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**BRANDON BREDLY ALVAREZ LÓPEZ**

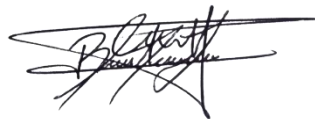


## HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### SISTEMA DE GESTIÓN DE RESERVA Y UTILIZACIÓN DE PARQUEOS

Tema que me fue asignado por la Dirección de Escuela de Estudios de Ingeniería en Ciencias y Sistemas, con fecha abril del 2023.



**Brandon Bredly Alvarez López**



**Joel Daniel Esaú Pérez Camey**



Guatemala, 21 de septiembre del 2023

Ingeniero  
**Carlos Alfredo Azurdia**  
**Coordinador de Privados y Trabajos de**  
**Tesis Escuela de Ingeniería en Ciencias y**  
**Sistemas Facultad de Ingeniería - USAC**

Respetable Ingeniero Azurdia:

Por este medio hago de su conocimiento que en mi rol de asesor del trabajo de investigación realizado por el estudiante **Joel Daniel Esaú Pérez Camey** con carné **201404268** y CUI **2994932450101** y al estudiante **Brandon Bredly Alvarez López** con carné **201403862** y CUI **2414819880101** titulado "**SISTEMA DE GESTIÓN DE RESERVA Y UTILIZACIÓN DE PARQUEOS**", luego de corroborar que el mismo se encuentra finalizado, lo he revisado y doy fe de que el mismo cumple con los objetivos propuestos en el respectivo protocolo, por consiguiente, procedo a la aprobación correspondiente.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



**Ing. Carlos Estuardo Coy de León**  
Colegiado No. 8079





Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 3 de octubre del 2023

Ingeniero  
**Carlos Gustavo Alonzo**  
**Director de la Escuela de Ingeniería**  
**En Ciencias y Sistemas**

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **JOEL DANIEL ESAÚ PÉREZ CAMEY** con carné **201404268** y CUI **2994 93245 0101** y **BRANDON BREDLY ALVAREZ LÓPEZ** con carné **201403862** y CUI **2414 81988 0101**, titulado: **“SISTEMA DE GESTIÓN DE RESERVA Y UTILIZACIÓN DE PARQUEOS”**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

A handwritten signature in black ink, appearing to read "Carlos Alfredo Azurdia".



**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados y Revisión  
de Trabajos de Graduación





UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA

SIST.LNG.DIRECTOR.11.EICSS.2023

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador de área y la aprobación del área de lingüística del trabajo de graduación titulado: **SISTEMA DE GESTIÓN DE RESERVA Y UTILIZACIÓN DE PARQUEOS**, presentado por: **Brandon Bredly Alvarez López**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería.

“ID Y ENSEÑAD A TODOS”

Ingeniero Carlos Gustavo Alonzo  
DIRECTOR  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, octubre de 2023

Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, -Escuela de Ciencias, Regional de Ingeniería Sanitaria y Recursos Hidráulicos (ERIS), Maestría en Sistemas Mención construcción y Mención Ingeniería Vial. Carreras: Ingeniería Mecánica, Ingeniería Electrónica, Ingeniería en Ciencias y Sistemas, Licenciatura en Matemática, Licenciatura en Física. Centros: de Estudios Superiores de Energía y Minas (CESEM). Guatemala, Ciudad Universitaria, Zona 12, Guatemala, Centroamérica



UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA

SIST.LNG.DIRECTOR.8.EICSS.2023

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador de área y la aprobación del área de lingüística del trabajo de graduación titulado: **SISTEMA DE GESTIÓN DE RESERVA Y UTILIZACIÓN DE PARQUEOS**, presentado por: **Joel Daniel Esau Perez Camey**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería.

“ID Y ENSEÑAD A TODOS”



Ingeniero Carlos Gustavo Alonzo  
DIRECTOR  
Escuela de Ingeniería en Ciencias y Sistemas

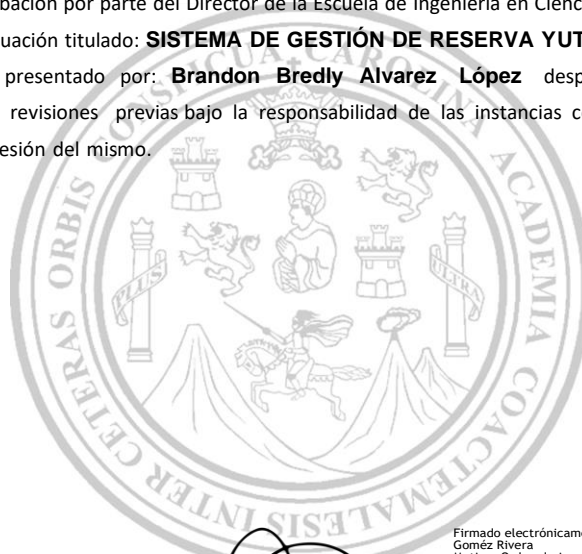
Guatemala, octubre de 2023

Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, -Escuela de Ciencias, Regional de Ingeniería Sanitaria y Recursos Hidráulicos (ERIS), Maestría en Sistemas Mención construcción y Mención Ingeniería Vial. Carreras: Ingeniería Mecánica, Ingeniería Electrónica, Ingeniería en Ciencias y Sistemas, Licenciatura en Matemática, Licenciatura en Física. Centros: de Estudios Superiores de Energía y Minas (CESEM). Guatemala, Ciudad Universitaria, Zona 12, Guatemala, Centroamérica



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **SISTEMA DE GESTIÓN DE RESERVA Y UTILIZACIÓN DE PARQUEOS**, presentado por: **Brandon Bredly Alvarez López** después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Firmado electrónicamente por: José Francisco Gómez Rivera  
Motivo: Orden de impresión  
Fecha: 25/10/2023 18:16:40  
Lugar: Facultad de Ingeniería, USAC.

Ing. José Francisco Gómez Rivera  
Decano a.i.



Guatemala, octubre de 2023

Para verificar validez de documento ingrese a <https://www.ingenieria.usac.edu.gt/firma-electronica/consultar-documento>

Tipo de documento: Correlativo para orden de impresión Año: 2023 Correlativo: 98 CUI: 2414819880101

Escuelas: Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, - Escuela de Ciencias, Regional de Ingeniería Sanitaria y Recursos Hidráulicos (ERIS). Postgrado Maestría en Sistemas Mención Ingeniería Vial. Carreras: Ingeniería Mecánica, Ingeniería Electrónica, Ingeniería en Ciencias y Sistemas. Licenciatura en Matemática. Licenciatura en Física. Centro de Estudios Superiores de Energía y Minas (CESEM). Guatemala, Ciudad



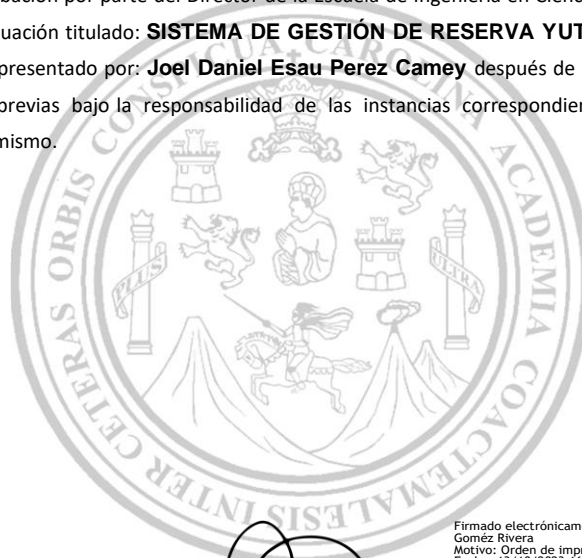


Decanato  
Facultad e Ingeniería  
24189101- 24189102

LNG.DECANATO.OIE.89.2023

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **SISTEMA DE GESTIÓN DE RESERVA Y UTILIZACIÓN DE PARQUEOS**, presentado por: **Joel Daniel Esau Perez Camey** después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



A handwritten signature in black ink, appearing to read 'J. Gómez'.

Firmado electrónicamente por: José Francisco Gómez Rivera  
Motivo: Orden de impresión  
Fecha: 13/10/2023 16:29:30  
Lugar: Facultad de Ingeniería, USAC.

Ing. José Francisco Gómez Rivera  
Decano a.i.



Guatemala, octubre de 2023

Para verificar validez de documento ingrese a <https://www.ingenieria.usac.edu.gt/firma-electronica/consultar-documento>

Tipo de documento: Correlativo para orden de impresión Año: 2023 Correlativo: 89 CUI: 2994932450101

Escuelas: Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, - Escuela de Ciencias, Regional de Ingeniería Sanitaria y Recursos Hidráulicos (ERIS). Postgrado Maestría en Sistemas Mención Ingeniería Vial. Carreras: Ingeniería Mecánica, Ingeniería Electrónica, Ingeniería en Ciencias y Sistemas. Licenciatura en Matemática. Licenciatura en Física. Centro de Estudios Superiores de Energía y Minas (CESEM). Guatemala, Ciudad





## **ACTO QUE DEDICO A:**

### **Dios**

Por brindarme la fuerza e inteligencia para poder alcanzar esta meta.

### **Mis padres**

Sandra Camey y Roni Pérez, por ser un pilar fundamental en mi vida, y ser mi apoyo durante todos los días de mi vida, que con su ejemplo me han enseñado el valor de hacer las cosas con esfuerzo, dedicación y honestidad.

### **Mis hermanos**

Pamela, Abraham y Angie Pérez, por estar siempre a mi lado, por su apoyo, compañía, y ser una fuente de inspiración.

### **Mi pareja**

Evelin Palacios, incondicional apoyo durante muchas facetas de mi vida.

**JOEL DANIEL ESAÚ PÉREZ CAMEY**



## **ACTO QUE DEDICO A:**

- Dios** Por brindarme la sabiduría para culminar esta meta.
- Mis padres** Norma López y Rigoberto Alvarez (q. e. p. d.), por creer siempre en mí, ser un apoyo incondicional en mi vida y por todo el sacrificio que hicieron antes y durante este proceso. Por enseñarme a esforzarme para conseguir cada una de mis metas.
- Mi hermano** Brayán Alvarez, por siempre estar para mí y nunca negar un favor siempre que lo necesité, gracias por tu paciencia.
- Mi pareja** Cristina Reynoso, por tener paciencia y brindarme su apoyo en cada una de mis decisiones.

**BRANDON BREDLY ALVAREZ LÓPEZ**



## **AGRADECIMIENTOS A:**

<b>Universidad de San Carlos de Guatemala</b>	Por brindarme la oportunidad de desarrollarme académicamente.
<b>Facultad de Ingeniería</b>	Por ser el centro de tantos aprendizajes y relaciones durante la carrera.
<b>Mis amigos</b>	Por su compañía, fidelidad, compartir sus conocimientos, risas, y desvelos a lo largo de la carrera.
<b>Mi asesor</b>	Carlos Coy, por creer en mi capacidad y acompañarme en el proceso de graduación.

**JOEL DANIEL ESAÚ PÉREZ CAMEY**



## **AGRADECIMIENTOS A:**

<b>Universidad de San Carlos de Guatemala</b>	Por darme la oportunidad de formarme como profesional.
<b>Facultad de Ingeniería</b>	Por brindarme los conocimientos necesarios para desarrollarme como ingeniero.
<b>Mis amigos</b>	Por su apoyo incondicional y desinteresado durante la carrera.
<b>Mi asesor</b>	Carlos Coy, por creer en mi capacidad y acompañarme en el proceso de graduación.

**BRANDON BREDLY ALVAREZ LÓPEZ**





## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES .....	V
TABLAS .....	VII
LISTA DE SÍMBOLOS .....	IX
GLOSARIO .....	XI
RESUMEN .....	XV
1. OBJETIVOS .....	1
2. INTRODUCCIÓN .....	3
3. ANTECEDENTES .....	5
4. ALCANCES Y LÍMITES .....	7
5. PLANTEAMIENTO DEL PROBLEMA .....	9
6. JUSTIFICACIÓN .....	11
7. Aplicaciones móviles .....	13
7.1. Aspectos generales .....	14
7.2. Funcionamiento .....	15
7.3. Componentes .....	15
8. Arquitectura .....	17
8.1. Nube .....	17
8.2. Backend .....	19
8.2.1. Springboot .....	20
8.2.2. Certbot .....	21
8.3. Aplicación móvil .....	21
8.3.1. React Native .....	22
8.3.2. Características .....	22
8.3.3. Requerimientos .....	23
9. Herramientas externas .....	25

9.1.	Jira .....	25
9.2.	Github .....	26
9.3.	Namecheap .....	26
9.4.	AWS .....	27
10.	Funcionalidades .....	29
10.1.	Aplicación móvil .....	29
10.1.1.	Registro .....	29
10.1.1.1.	Datos personales .....	29
10.1.1.2.	Confirmar correo electrónico y nueva contraseña .....	31
10.1.2.	Iniciar sesión .....	31
10.1.3.	Cerrar sesión .....	32
10.1.4.	Recuperar contraseña .....	33
10.1.4.1.	Verificar correo electrónico .....	33
10.1.4.2.	Crear nueva contraseña .....	35
10.1.5.	Perfil .....	35
10.1.5.1.	Actualizar información .....	36
10.1.6.	Chat .....	36
10.1.6.1.	Visualizar parqueos .....	37
10.1.6.2.	Visualizar mensajes enviados .....	37
10.1.6.3.	Conversación con parqueo .....	38
10.1.7.	Listado de reservaciones .....	39
10.1.7.1.	Reservaciones por tipo .....	40
10.1.7.2.	Cancelar reservación .....	41
10.1.8.	Configuraciones .....	42
10.1.8.1.	Cambiar contraseña .....	43
10.1.8.2.	Administrar vehículos .....	44
10.1.8.2.1.	Listado de vehículos .....	44
10.1.8.2.2.	Eliminar vehículo .....	45

10.1.8.2.3.	Agregar vehículo .....	46
10.1.8.3.	Método de pago .....	48
10.1.8.3.1.	Listado de cuentas bancarias .....	48
10.1.8.3.2.	Eliminar cuenta bancaria .....	49
10.1.8.3.3.	Agregar cuenta bancaria .....	50
10.1.8.4.	Eliminar usuario .....	52
10.1.9.	Mapa .....	53
10.1.9.1.	Parqueos .....	56
10.1.9.2.	Reservaciones .....	58
10.1.10.	Configuración de idioma .....	60
10.2.	Consola de administración .....	61
10.2.1.	Pantalla de inicio de sesión .....	61
10.2.2.	Pantalla inicial .....	63
10.2.3.	Administración de parqueos .....	64
10.2.3.1.	Creación de parqueo .....	66
10.2.3.2.	Edición de parqueo .....	67
10.2.3.3.	Activar y desactivar parqueos .....	68
10.2.4.	Reservaciones .....	71
10.2.4.1.	Desactivar reservaciones .....	71
10.2.3.	Pagos .....	72
11.	Estructura del proyecto.....	75
11.1.	Configuración de ambiente de desarrollo para backend .....	75
11.2.	Descripción de archivo application.yml .....	83
11.3.	Estructura de carpetas en servidor de producción .....	85
11.4.	Base de datos .....	87
11.4.1.	Script SQL con esquema de la base de datos .....	88
11.4.2.	Descripción de tablas .....	92
11.4.2.1.	Entidad bank_account .....	93
11.4.2.2.	Entidad chat .....	94

11.4.2.3.	Entidad message .....	95
11.4.2.4.	Entidad parking .....	96
11.4.2.5.	Entidad reservation .....	97
11.4.2.6.	Entidad user .....	98
11.4.2.7.	Entidad vehicle .....	99
11.5.	Configuración de ambiente de desarrollo para frontend .....	100
11.5.1.	Modo desarrollador .....	100
11.5.2.	Código fuente y dependencias .....	101
11.5.3.	Almacén de claves .....	103
11.5.4.	Configuración para AWS .....	105
11.5.5.	Ejecutar proyecto .....	106
12.	Presentación de resultados .....	109
12.1.	Beneficio de uso de la plataforma .....	109
12.2.	Oportunidades .....	109
12.3.	Consideraciones .....	110
CONCLUSIONES .....		113
RECOMENDACIONES .....		115
REFERENCIAS .....		117

## ÍNDICE DE ILUSTRACIONES

### FIGURAS

Figura 1.	Arquitectura en la nube .....	19
Figura 2.	Pantalla de registro .....	30
Figura 3.	Código de verificación para registro .....	30
Figura 4.	Ingreso de código de verificación y nueva contraseña .....	31
Figura 5.	Inicio de sesión con correo electrónico y contraseña .....	32
Figura 6.	Cierre de sesión .....	33
Figura 7.	Recuperar contraseña de usuario nuevo .....	34
Figura 8.	Código para confirmar nueva contraseña .....	34
Figura 9.	Confirmar nueva contraseña .....	35
Figura 10.	Pantalla de perfil, con dato del usuario .....	36
Figura 11.	Listado de parqueos disponibles para conversación .....	37
Figura 12.	Listado de mensajes enviados .....	38
Figura 13.	Conversación activa con parqueo .....	39
Figura 14.	Listado de reservaciones .....	40
Figura 15.	Filtrado de reservación .....	41
Figura 16.	Cancelar reservación .....	42
Figura 17.	Menú de configuraciones .....	43
Figura 18.	Cambio de contraseña .....	44
Figura 19.	Listado de vehículos .....	45
Figura 20.	Eliminar vehículo .....	46
Figura 21.	Agregando nuevo vehículo .....	47
Figura 22.	Confirmando nuevo vehículo .....	48
Figura 23.	Listado de cuentas bancarias .....	49

Figura 24.	Confirmando eliminación de cuenta bancaria .....	50
Figura 25.	Formulario de nueva cuenta bancaria .....	51
Figura 26.	Confirmando nueva cuenta bancaria .....	52
Figura 27.	Confirmación de eliminación de cuenta de usuario .....	53
Figura 28.	Mapa de parqueos .....	54
Figura 29.	Mapa de parqueos disponibles .....	55
Figura 30.	Crear reservación .....	57
Figura 31.	Reservación creada .....	58
Figura 32.	Reservaciones por día .....	59
Figura 33.	Filtrar reservaciones por fecha .....	60
Figura 34.	Cambio de idioma en aplicación .....	61
Figura 35.	Inicio de sesión .....	62
Figura 36.	Pantalla inicial en consola de administración .....	63
Figura 37.	Listado de parqueos de usuario .....	64
Figura 38.	Acciones habilitadas para un parqueo .....	65
Figura 39.	Creación de parqueo .....	66
Figura 40.	Edición de parqueo .....	67
Figura 41.	Datos para editar parqueo .....	68
Figura 42.	Desactivar parqueo .....	69
Figura 43.	Restaurar parqueo deshabilitado .....	70
Figura 44.	Listado de reservaciones .....	71
Figura 45.	Desactivar reservaciones .....	72
Figura 46.	Listar pagos .....	73
Figura 47.	Detalle de pagos .....	73
Figura 48.	Instalación jdk de java .....	75
Figura 49.	Validación de versión de java .....	76
Figura 50.	Configuración de repositorio para docker .....	77
Figura 51.	Instalación de docker .....	77
Figura 52.	Ejecutando comando de hola mundo .....	78

Figura 53.	Mensaje de éxito con docker .....	78
Figura 54.	Creación de contenedor .....	79
Figura 55.	Iniciando y parando contenedor .....	80
Figura 56.	Configuración de conexión a base de datos .....	80
Figura 57.	Mensaje de conexión exitosa .....	81
Figura 58.	Inicializar proyecto de springboot .....	82
Figura 59.	Archivo de entorno .....	83
Figura 60.	Directorio de carpetas de servidor .....	86
Figura 61.	Diagrama entidad relación de base de datos .....	88
Figura 62.	Script de base de datos parte 1 .....	89
Figura 63.	Script de base de datos parte 2 .....	90
Figura 64.	Script de base de datos parte 3 .....	91
Figura 65.	Estructura general de código de aplicación móvil .....	101
Figura 66.	Generación de almacén de claves .....	104
Figura 67.	Ubicación de archivo de almacén de claves .....	105
Figura 68.	Archivo de conexión con AWS .....	105
Figura 69.	Comando para ejecutar proyecto .....	107

## TABLAS

Tabla 1.	Variables en fórmula para calcular precio de reservación ....	56
Tabla 2.	Descripción archivo yml .....	84
Tabla 3.	Descripción estructura de carpetas .....	86
Tabla 4.	Descripción de tablas de base de datos .....	92
Tabla 5.	Detalle de campos de tabla bank_account .....	93
Tabla 6.	Detalle de campos de tabla chat .....	94
Tabla 7.	Detalle de campos de tabla message .....	95
Tabla 8.	Detalle de campos de tabla parking .....	96
Tabla 9.	Detalle de campos de tabla reservation .....	97

Tabla 10. Detalle de campos de tabla user ..... 98

Tabla 11. Detalle de campos de tabla vehicle ..... 99

Tabla 12. Pasos para modo desarrollador ..... 100

Tabla 13. Comandos para clonar repositorio e instalación de librerías. 101

Tabla 14. Descripción general de carpetas y archivos del proyecto..... 102

Tabla 15. Parámetros al ejecutar comando de generación de almacén  
de claves..... 104

Tabla 16. Descripción de contenido de archivo aws-exports.js ..... 106



## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>yml</b>	Extensión para archivos de datos.
<b>http</b>	Protocolo de transferencia de hipertexto.
<b>https</b>	Protocolo seguro de transferencia de hipertexto.
<b>Mb</b>	Mega Bite
<b>DB</b>	Base de datos
<b>SSL</b>	Seguridad de capa de transporte
<b>\$</b>	Dólar



## GLOSARIO

<b>API</b>	Una API es un conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar el software de las aplicaciones. Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados.
<b>Base de datos</b>	Una base de datos es una colección organizada de información estructurada, o datos, típicamente almacenados electrónicamente en un sistema de computadora.
<b>DDL</b>	Lenguaje de definición de datos, las sentencias DDL son aquellas que sirven para definir objetos dentro de la base de datos.
<b>DML</b>	lenguaje de manipulación de datos, las sentencias DML son aquellas utilizadas para insertar, borrar, modificar y consultar los datos de una base de datos.
<b>ESLint</b>	Herramienta para analizar el código del desarrollador e identifica patrones y problemas en el lenguaje de JavaScript
<b>Framework</b>	El Framework es una especie de plantilla, un esquema conceptual, que simplifica la elaboración de una tarea,

ya que solo es necesario complementarlo de acuerdo a lo que se quiere realizar.

## **HTML**

HTML es el lenguaje con el que se define el contenido de las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web.

## **HTTP**

El protocolo de transferencia de hipertexto, es el protocolo que utilizan los navegadores para comunicarse con los servidores web, y el que deben seguir el resto de software perteneciente a la arquitectura de la web.

## **HTTPS**

Es un protocolo de comunicación de Internet que protege la integridad y la confidencialidad de los datos de los usuarios entre sus ordenadores y el sitio web.

## **Metro Bundler**

Herramienta que permite empaquetar todos los archivos de React native en uno solo para instalarlo en un dispositivo

## **Patrones de diseño de software**

Los patrones de diseño son soluciones habituales a problemas comunes en el diseño de software.

## **Scripting**

Los lenguajes de scripts pueden realizar diferentes acciones dentro de un entorno de tiempo de ejecución particular, como automatizar la ejecución de tareas, mejorar la funcionalidad del software principal, realizar

configuraciones, extraer datos de conjuntos de datos y otros.

## **SQL**

Es un lenguaje de programación estandarizado que se utiliza para administrar bases de datos relacionales y realizar diversas operaciones con los datos que contienen.

## **TypeScript**

Tecnología utilizada en el lenguaje de programación de JavaScript que permite agregar el tipado de datos y objetos basado en clases, manteniendo así un código más ordenado



## RESUMEN

El sistema de gestión de reserva y utilización de parqueos con una aplicación móvil y múltiples consolas de administración nace en la necesidad de aportar a la sociedad guatemalteca, una plataforma segura y confiable para poder acceder y utilizar los distintos parqueos.

Se buscó con este proyecto mejorar el viaje que realizan los pilotos y al mismo tiempo proporcionar una herramienta a los administradores para poder promover, administrar y monitorear sus negocios, enfocado principalmente en tener un proceso claro y transparente para todos.

Se inició el proyecto indagando en los distintos procesos y maneras de llevar el control de tiempos en los parqueos, muchos se contaban simplemente con un papel anotado la hora de ingreso, hasta generación de boletos con cinta magnética de forma automática, se realizó una investigación para generar un sistema de gestión, que no agregue costo, ni complejidad a los negocios.

Durante el transcurso del proyecto fue necesario tener retroalimentación de los usuarios finales, así como realizar correcciones, se realizaron presentaciones, manuales de usuario y videos explicativos de la plataforma creada.

El proyecto implementará múltiples pantallas donde cada usuario puede visualizar solamente los aspectos donde él debe interactuar o donde debe ser notificado del estado de su proceso.





# 1. OBJETIVOS

## **General**

Desarrollo de una plataforma que permita promover, administrar y monitorear los distintos parqueos localizados en la república de Guatemala.

## **Específicos**

1. Facilitar la gestión de espacios, tiempos y costos por parte de los administradores de parqueos.
2. Geolocalizar distintos parqueos proporcionando información relevante a usuarios que deseen encontrar un parqueo.
3. Automatizar el control de tiempo dentro de parqueos.



## 2. INTRODUCCIÓN

El presente trabajo está constituido por diferentes aspectos concentrados en el desarrollo de una aplicación que ayude a con el parque vehicular en crecimiento.

Para poder crear un buen plan y llevar a cabo el proyecto se elaboró un plan de trabajo que contaba con tres etapas, la primera fue una fase de investigación, seguido de la fase técnico profesional y finalizando con la fase de enseñanza, documentación y aprendizaje.

El sitio oficial del Departamento de tránsito de la Policía Nacional Civil reporta basado en el Observatorio Nacional de Seguridad del Tránsito (ONSET) un incremento en la cantidad de vehículos en los últimos años, dado que, para mayo del 2021, se contabilizaron 4 millones 289 792 vehículos en toda la república, mientras que a finales de 2020 había 4 millones 096 mil 800, lo que significa un aumento de 192 mil 912, un 4.50% más.

Es necesario debido a este aumento del parque, una mejor coordinación y planificación de los pocos espacios para poder parquear los vehículos, esta aplicación plantea una propuesta de solución a esta problemática.



### **3. ANTECEDENTES**

En el año 2017 los vecinos y visitantes en Antigua Guatemala mostraron sus molestias debido a que fueron multados con hasta Q.100.00 por estacionarse en lugares no autorizados, esto debido a que las autoridades con el objetivo de liberar la desde la 3ra hasta la 6ta calle, así como la 2da y 5ta avenida, colocaron señalizaciones para que los vehículos pudieran circular en doble carril, disminuyendo así el espacio para los parqueos en la vía pública. Esto provocó molestias en las personas porque no se les dio una solución para el estacionamiento de sus vehículos, buscando lugares lejanos lo que provoca que estos sean vulnerables ante la delincuencia.

En el Instituto Guatemalteco de Seguridad Social de Pamplona en la zona 12 de la ciudad capital, al ser un lugar visitado por una gran cantidad de personas y quienes llegan en vehículos propios se les dificulta ubicar un parqueo, ya que se ven obligados a dejar al paciente y utilizar los lugares que están reservados por personas quienes cobran por estos espacios públicos, cuyos precios pueden ir desde Q10.00 hasta Q.30.00 siendo esto un riesgo, debido a que si no pagan sus vehículos pueden sufrir daños parciales o totales. Asimismo, estas personas utilizan botes plásticos, vallas, conos para reservar estos espacios evitando que las personas puedan estacionar sus vehículos. Esto se ve comúnmente en lugares que son altamente concurridos, actividades tales como la Cuaresma, Semana Santa, conciertos y actividades de fin año.

Durante el periodo de la pandemia Covid-19 se produjo un aumento en la adquisición de vehículos por parte de las personas, debido a que unos fueron despedidos y tuvieron que trabajar por su propia cuenta; otras personas

afectadas por la falta de transporte público y poder dirigirse a su área de trabajo, aplicando así para vehículos y motocicletas.

## 4. ALCANCES Y LÍMITES

Utilizar un sistema de gestión de reservas y utilización de parqueos, puede brindar un gran alcance para optimizar procesos de reservas, mejorar la experiencia de usuarios y calidad del medio ambiente, así como la optimización de recursos y la parte administrativa en los estacionamientos, por lo que a continuación se presentan los siguientes casos:

1. Evitar el desconocimiento o incertidumbre de las personas al momento de dirigirse a un lugar y desconocer si existe algún estacionamiento seguro y su disponibilidad.
2. Manejo automatizado para la asignación de espacios disponibles al momento de las reservaciones.
3. Obtener información y brindar datos estadísticos de los parqueos para tomar mejores decisiones administrativas, así como estratégicas.
4. Poder integrar distintos métodos de pagos para brindar tranquilidad a los usuarios.

Asimismo, también se cuentan con limitantes o desafíos para el desarrollo de este sistema, de los cuales se mencionan los siguientes:

1. Algunos usuarios no estarán en disposición o no tendrán la capacidad de poder utilizar este sistema, por cómo se ha venido manejando los parqueos recientemente, lo que dificultará el alcance o utilidad.
2. El sistema requerirá de cambios, mantenimiento o actualizaciones por lo que, los usuarios deben estar al día con estos cambios.

3. Se requiere una inversión inicial, dado que cada administrador debe contar con una computadora física para poder consultar datos administrativos.
4. El sistema no está exento a fallas técnicas, lo que puede afectar a una mala experiencia de los usuarios.
5. Se debe contar con una infraestructura controlada para tener un manejo exacto de los espacios a utilizar en los estacionamientos.



## 5. PLANTEAMIENTO DEL PROBLEMA

Citando el artículo 26 de la Ley de Tránsito de Guatemala, indica que “el estacionamiento de vehículos en la vía pública se hará conforme a las disposiciones de la autoridad de tránsito correspondiente”, por lo que se puede determinar que las autoridades pueden distribuir el espacio público como mejor les parezca posible con el fin de mejorar la afluencia de vehículos, cambiar el sentido de los carriles en un rango de horarios, evitar que personas estacionen sus vehículos en determinados lugares y horarios.

El artículo 24 de la Ley de Tránsito de Guatemala, indica que “la autoridad de tránsito está facultada para retirar de la vía pública cualquier cosa, vehículo, material, propaganda u otro tipo que obstaculice la circulación de personas y vehículos y para trasladarla y depositarla, a costa del propietario, en los predios autorizados para tal efecto”, por lo que la acción de las personas que colocan objetos para cobrar una cuota por su utilización para estacionamiento está totalmente prohibido por la ley, y si alguna persona obliga a pagar dicha cuota puede considerarse como una extorsión lo cual, puede notificarse a la policía.

El artículo 27 de la Ley de Tránsito de Guatemala, indica que “se autoriza construir y habilitar parqueos subterráneos o por elevación en calles, parques u otros bienes nacionales o municipales de uso común”. Si bien, la utilización de parqueos ayuda a mantener seguros los vehículos por daños parciales o totales, esto no resuelve la problemática de las personas, al no poder determinar en qué horarios estarán habilitados los parqueos cercanos a su destino o si se encuentran en la disponibilidad de espacios para poder reservar cuando deseen

utilizarlos, por lo que se requiere también, que dentro de un establecimiento haya un encargado responsable para saber qué usuarios van a llegar con un vehículo.

## 6. JUSTIFICACIÓN

La importancia que se tiene al tener un sistema que permita la gestión de reservas y utilización de parqueos, es debido a que en muchos lugares en Guatemala, con el aumento en la población, así es el aumento que se tiene de vehículos, por lo que la demanda de estacionamientos se ve incrementada, lo que genera problemas de congestionamiento en las calles con vehículos estacionados en la vía pública por la falta de espacios disponibles y habilitados. Utilizando un sistema de gestión permitirá optimizar el uso de los espacios disponibles, pudiendo darles a las personas una forma distinta de conocer el lugar que dejarán su vehículo con un tiempo de anticipación a la llegada a su destino, reduciendo así los tiempos de espera lo que disminuye la emisión de dióxido de carbono de los vehículos ayudando también al medio ambiente permitiría mejorar también el flujo vehicular.

Utilizar un sistema de gestión de parqueos, mejorará la experiencia al usuario al momento de buscar un estacionamiento y reducirá los niveles de estrés, así como la incertidumbre de las personas al momento de buscar un lugar para estacionar su vehículo.

Se puede brindar mayor seguridad a las personas como a sus vehículos, ya que estos al estar estacionados en lugares públicos pueden ser dañados parcial o totalmente, así como hurtados. Reduciría la cantidad de asaltos a las personas cuando estos salen o regresan a su vehículo y tiene un mayor control y seguridad en los vehículos que ingresan y salen del estacionamiento.



## 7. APLICACIONES MÓVILES

Es una aplicación informática que es creada con el objetivo de ser instalada y ejecutada en un dispositivo móvil inteligente y estas pueden ser desarrolladas en un lenguaje de programación ya sea nativo, lo que significa que la aplicación está desarrollada en el lenguaje propio del dispositivo como Objective-C o Swift para dispositivos iOS, Java, Kotlin o C++ para dispositivos Android y .Net para Windows Phone; y lenguajes de programación híbridos que, utilizando un mismo lenguaje de programación y posterior compilación, se generará una aplicación tanto para dispositivos Android como iOS. Dentro de estos lenguajes híbridos se encuentra React Native que fue desarrollado por Facebook y utiliza como base JavaScript o TypeScript en versiones más recientes; también se puede mencionar Ionic y fue creada por Drifty Co. en el año 2013 utilizando como base para su codificación TypeScript; y Flutter que fue desarrollada por Google y se usa el lenguaje de programación Dart.

Por lo general, estas aplicaciones móviles pueden encontrarse de manera gratuita o ya sea de paga en su respectiva plataforma de distribución, las cuales pueden ser App Store, que es administrada por Apple para dispositivos iOS; Play Store administrada por Google para dispositivos con sistema operativo Android con servicios de Google y App Gallery, que es administrada por Huawei para dispositivos Android de marca Huawei. Por políticas en la tienda Play Store, no pueden existir aplicaciones que tras su proceso de compilación estas contengan código de Huawei, sin embargo; en App Gallery pueden existir aplicaciones con servicios de Google sin ninguna restricción. Así también, pueden encontrarse aplicaciones en sitios web para su instalación y ejecución en dispositivos Android, pero no se recomienda instalarlas ya que, pueden contener alguna funcionalidad

para el robo de información o utilización de datos de forma maliciosa para el usuario.

## **7.1. Aspectos generales**

Para poder obtener una aplicación, estas deben incluir el sufijo *.apk*, que es el paquete o archivo de almacenamiento que incluye todo el contenido de la aplicación, y es el utilizado para instalarse dentro de un dispositivo con sistema operativo Android, previamente necesita de un kit de desarrollo de software (SDK en inglés) para poder compilar correctamente el código de la aplicación y así generar este archivo APK.

Dado que un sistema operativo Android es un sistema operativo Linux de múltiples usuarios, cada una de las aplicaciones que el dispositivo tenga instaladas, estas son interpretadas como usuarios diferentes. El sistema es el encargado de asignarle un identificador de Linux a cada aplicación, por lo que solo el sistema lo conoce y no es posible obtenerlo por la misma aplicación. Y cada una de las aplicaciones es ejecutada en su propia máquina virtual, por lo que una aplicación no depende de otra para su funcionamiento.

Con esto se garantiza que la aplicación dentro del sistema Android pueda acceder de forma predeterminada únicamente a los recursos que necesita para poder ejecutarse de forma correcta, por lo que la aplicación no puede acceder a funciones del sistema Android, a los cuales no tiene permiso, sin embargo, es posible añadir permisos en un archivo en específico, el cual será solicitado en cierta parte de la aplicación y definida por el desarrollador, para que este pueda ser aceptado o denegado por el usuario y así, en caso de ser aceptado poder acceder a la información únicamente del permiso solicitado.

## **7.2. Funcionamiento**

Dado que existen aplicaciones móviles gratuitas y otras que son de paga, para que las aplicaciones gratuitas puedan tener algún beneficio estas pueden tener espacios publicitarios, realizar micro transacciones para obtener algún añadido como se da mayormente en juegos; también están las aplicaciones que brindan una capa gratuita pero es necesario el pago mensual o anual para poder utilizar todas sus funcionalidades, tal y como funciona Spotify; o aplicaciones que tienen relación con los bancos del país donde se encuentren para realizar transacciones y así obtener comisiones como lo son Acreditame, Nexa, Kash, entre otras, que a mayor cantidad de transacciones puedan generar su ganancia. Y las aplicaciones de paga, que si bien su descarga es gratuita es necesario de una membresía mensual o anual para poder mantener en uso cada uno de los servicios que estos ofrecen, tal y como funcionan las aplicaciones como Netflix, Star Plus, entre otras.

## **7.3. Componentes**

Cada componente son puntos de entrada por el que un usuario puede entrar a una aplicación, por lo que unos componentes pueden necesitar de otros para su funcionamiento. Y cada aplicación está conformada por cuatro tipo de componentes distintos los que pueden ser:

- Una actividad es un punto de entrada en una aplicación, por lo que puede ser representada como una pantalla mostrada hacia el usuario; aunque la aplicación interactúe con distintas actividades, cada una de estas es independiente de las demás.
- Un servicio es un punto de entrada que permite a la aplicación mantenerse en ejecución, aunque la aplicación se encuentre en segundo plano, por lo que, este componente no brinda una interfaz de usuario. Un ejemplo claro

es un reproductor de música, que sigue ejecutando sus procesos aunque no se esté dentro de la aplicación.

- Un receptor de emisión permite que el sistema entregue eventos a la aplicación fuera de un flujo normal de ejecución. Aquí se puede mencionar la aplicación de alarma, que puede ejecutarse aunque no se esté dentro de la aplicación, ni se encuentre en segundo plano. También se encuentra la creación de notificaciones en la barra de estados, que muestra alguna información aunque la aplicación no se esté ejecutando, lo que ocurre cuando llegan mensajes de Whatsapp.
- Un proveedor de contenido es el encargado de administrar el conjunto compartido de datos que pueden ser almacenados en la web, sistema de archivos, en una base de datos SQLite o cualquier otra ubicación a la que la aplicación tenga acceso.



## 8. ARQUITECTURA

### 8.1. Nube

La infraestructura fue realizada en la plataforma de Amazon Web Service(AWS), haciendo uso de diferentes servicios que esta ofrece. La arquitectura ofrece la posibilidad de poder escalar los servidores que se utilizan para proporcionar siempre una alta disponibilidad y bajos tiempos de respuesta.

La base fundamental sobre la cual trabajan los servidores es una red virtual aislada, también llamada VPC(Virtual private cloud), esta proporciona la posibilidad de construir una infraestructura de forma totalmente aislada a cualquier otro proyecto desarrollado dentro del sistema de Amazon Web Service. Una VPC está conformada de subredes, que pueden ser redes públicas o privadas para el caso se debe configurar una subred privada, la cual contendrá los servidores donde se ejecutarán nuestros servicios desarrollados en Java con Spring boot.

El servicio de API Gateway que es el que expone nuestro API al internet, utiliza varias herramientas para poder funcionar de forma correcta, Cognito es una herramienta que se utiliza para el manejo de usuarios dentro de la plataforma, esta herramienta se encarga de realizar el registro de usuarios, tokens de sesión, contraseñas, registro de dispositivos desde donde se autentica un usuario, así como validar los tokens de sesión que son enviados de parte del usuarios, que pertenezcan a un usuario existente y recientemente autenticado dentro del sistema. API Gateway al ser el punto de exposición a internet, debe

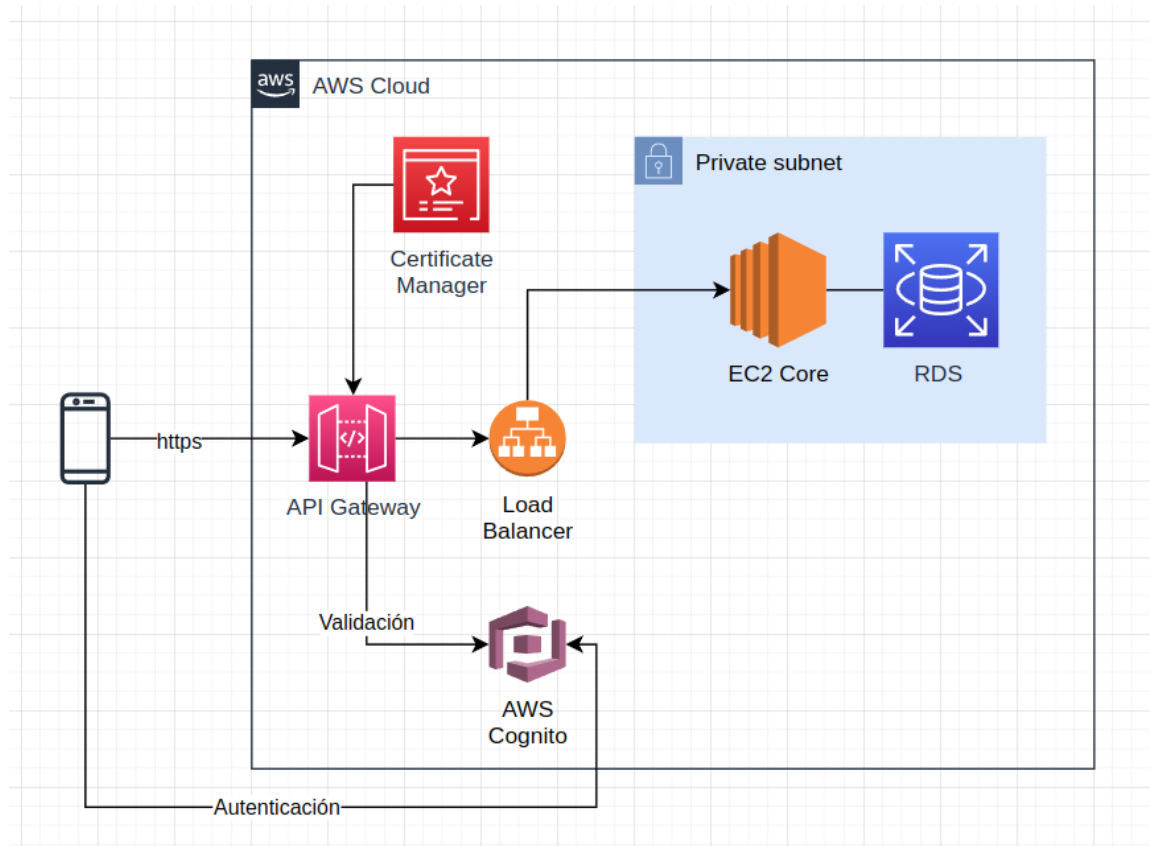
poseer un certificado SSL(Secure Socket Layer), el cual garantiza que el dominio el cual se está consumiendo está encriptado y es seguro.

El balanceador de carga es la herramienta encargada de direccionar las solicitudes que realizan los usuarios finales a los servidores, basándose en los objetivos que se registraron previamente, se realizó la configuración para que funcionara como un Proxy, por el cual los servidores reciben las peticiones con todos sus parámetros.

Los servidores son proveídos por medio de la herramienta EC2, esta herramienta permite la configuración de grupos de seguridad para poder configurar los puertos expuestos que tendrá este servidor, lo cual nos provee una capa extra de seguridad.

**Figura 1.**

*Arquitectura en la nube*



*Nota.* Diagrama de arquitectura en la nube. Elaboración propia, realizado en draw.io.

## 8.2. Backend

El desarrollo de la plataforma en la cual se desarrolló toda la lógica del sistema se encuentra constituido tanto de herramientas de transporte de datos, así como de análisis de datos.

### **8.2.1. Springboot**

Spring boot es una de los frameworks más utilizados para el lenguaje de programación Java, este facilita el desarrollo de aplicaciones web, y también de microservicios.

Los archivos de configuración permiten poder parametrizar aspectos como límite tiempos de respuesta, conexiones a base de datos, zona horaria, puerto de escucha y variables de entorno que se utilizaran dentro del sistema, lo cual nos permite poder llevar la configuración a un nivel muy específico a lo que necesitemos.

Los archivos controladores, son archivos en los cuales se configuran las rutas, parámetros, variables de ruta y cuerpo de las peticiones que recibira nuestro microservicio, en este maneja la logica de respuestas que recibira el usuario, asi como las respuestas que obtendrá en caso de fallar alguna petición dentro de nuestro microservicio.

Los archivos DAO(Data Access Object) son los encargados de la lógica de acceso a la base de datos, básicamente estos archivos se encargan de recibir ciertos parámetros y preparar una respuesta estructurada en base a los parámetros recibidos, ya sea por medio de un archivo repository para acceder a base de datos o por medio de datos obtenidos de otros medios.

Los archivos repositorios son los archivos encargados de describir la forma en que se obtienen información de la base de datos, pudiendo configurar que objeto se irá a traer de base de datos, así como el manejo de paginaciones y uso de sentencias SQL.

Los archivos entity se utilizan para realizar un mapeo a las distintas tablas dentro de la base de datos, estas contienen como variables, los campos que existen en la tabla a la cual están haciendo referencia, se debe especificar el tipo de dato que es almacenado en base de datos, así como si existe alguna relación con otras tablas para un mejor manejo.

### **8.2.1. Certbot**

El uso de peticiones a los servidores se hace por medio del protocolo de transferencia de hipertexto(Http), lo cual permite que los usuarios puedan obtener información de los servidores, por lo cual se desarrollo una mejora para este protocolo, proporcionando un encriptado el cual permite que los datos no puedan ser visibles, más que solamente por el emisor y el receptor, ya que son los únicos que poseen la clave con la cual se puede realizar la descripción de la información. La mejora de seguridad requiere que se genere un certificado de capa de socket(SSL), el cual se utilizara para la encriptación, este certificado se puede generar si se posee acceso a el servidor de dominios del cual se desea generar el certificado.

La plataforma Certbot, proporciona las herramientas para poder generar certificados de una forma fácil y rápida, con lo cual se pueden generar, y exportar dichos certificados a la herramienta Certificate Manager, el cual se encargará de la renovación de forma automática de los certificados cada dos meses y medio.

### **8.3. Aplicación móvil**

Para desarrollar una aplicación móvil es necesario utilizar un lenguaje de programación híbrido o nativo, tener una computadora física configurada para la implementación del código y un dispositivo físico o un emulador para realizar las pruebas de la aplicación.

### **8.3.1. React Native**

En el año 2015 Facebook lanza la librería de React Native para el lenguaje de programación de JavaScript y actualmente continúa recibiendo actualizaciones para mejorar su desempeño. En el año 2018 es donde obtiene el segundo lugar a nivel mundial, con el mayor número de contribuciones por parte de desarrolladores en su repositorio de GitHub. Actualmente cuenta con el respaldo de empresas como Expo, Microsoft, Infinity Red y Software Mansion. También cuenta con aplicaciones desarrolladas con esta librería, que ya se encuentran publicadas en su respectiva tienda de distribución como pueden ser: Facebook, Microsoft Office, Xbox Game Pass, Tesla, Discord, PlayStation App, PUMA, NFL, entre otras, y todas estas disponibles para dispositivos Android y iOS en Google Play Store y App Store respectivamente.

### **8.2.1. Características**

React Native también implementa funcionalidades que ya se encuentran integradas en React, que también es una librería de JavaScript, pero ésta orientada al desarrollo de aplicaciones web. A partir de la versión 0.71.0 de React Native se añade TypeScript para el desarrollo de la aplicación lo que permite un mejor manejo en el tipo de datos. Se pueden crear aplicaciones completamente nuevas o se puede usar en proyectos existentes tanto de Android como iOS. Cuenta con actualización rápida, lo que significa que al programar en la aplicación, realizar cambios y guardarlos podrán ser visualizados de forma inmediata a través del emulador o dispositivo físico, sin necesidad de compilar por completo la aplicación, lo que disminuye el tiempo de desarrollo.

### **8.2.3. Requerimientos**

Previo a poder utilizar React Native, es necesario instalar el entorno de desarrollo integrado (IDE en inglés) de Android Studio, el cual se encuentra disponible para Windows, Linux y Mac. Se recomienda que, si una aplicación se publicará en la tienda App Store es requisito indispensable utilizar una computadora con sistema operativo MacOS, ya que es necesario utilizar XCode para realizar pruebas y posterior publicación, de lo contrario se puede utilizar una computadora con Windows o Linux. También es necesario instalar Node.js, Java JDK 11, Visual Studio Code, Git y tener un dispositivo Android físico. En el sitio web oficial de React Native se encuentra detalladamente el proceso para poder configurar el equipo a utilizar y ejecutar las aplicaciones.





## 9. HERRAMIENTAS EXTERNAS

### 9.1. Jira

Es un software diseñado para la gestión de proyectos, seguimiento de errores, priorización de tareas, presentación de resultados. La plataforma Jira, ayudó a desglosar todas las actividades que fueron necesarias en el desarrollo del proyecto.

La gestión de proyectos se puede realizar por cuatro tipos de incidencias, empezando desde la que representa una mayor carga de trabajo épica, esta incidencia representa un desarrollo o esfuerzo que conlleva un grupo de errores, historias y tareas dentro de ella.

Las historias dentro de la plataforma Jira, se utilizan para poder describir un segmento del proyecto, así como algún entregable o meta esperada dentro del proyecto. Las historias muchas veces necesitan ser compuestas de actividades más pequeñas que se unirán para lograr el objetivo de la historia, estas actividades reciben el nombre de tarea.

Las actividades *bug*, se utiliza este tipo de actividad para llevar un seguimiento de algún comportamiento no esperado ni deseado dentro de la plataforma, Jira permite asignar la prioridad a cada una de estas actividades.

## **9.2. Github**

Durante el desarrollo del proyecto fue necesario poder manejar un control del código del proyecto, el cual se realizó por medio de la plataforma Github, esta herramienta permite gestionar las versiones del proyecto.

Se utilizaron tres repositorios los cuales reciben el nombre de parking-app, parking-console y parking-backend, esta herramienta trabaja bajo comandos los cuales ayudan a llevar un control de los cambios que se realizaron, los principales son la creación de ramas, hacer posteo de código nuevo o modificado, unificación de cambios.

## **9.3. Namecheap**

El sistema de nombres de dominio (DNS) es utilizado como un directorio del internet, el cual facilita el poder acceder a distintos recursos sin tener que memorizar direcciones IP, esto nos ayuda a poder buscar páginas web o servicios de forma fácil, para poder utilizar un nombre de dominio se debe de registrar por medio de alguna plataforma que provea este sistema. La plataforma de namecheap es una de las más utilizadas, para poder registrar un dominio se debe de validar primero la disponibilidad de este, luego realizar un proceso de compra.

Un dominio por sí solo no representa estar conectado a algún servidor, es necesario realizar la configuración y apuntar a la dirección IP en la cual se encuentra la página web o el API que se desea consumir, en este caso al estar siendo configurado junto con API Gateway, este proporciona un DNS interno, al cual se redirecciona.

#### **9.4. AWS**

La plataforma de Amazon Web Service (AWS) que provee la capacidad de tener cómputo en la nube, AWS cuenta con más de doscientos servicios disponibles, los cuales abarcan computo, base de datos, almacenamiento de archivos, facilidades para aplicaciones móviles y páginas web, tecnología sin servidor. La gran variedad de servicios que la plataforma provee permite poder crear una infraestructura que se adapte a las necesidades del proyecto.

Los servicios que se utilizaron en el proyecto fueron Cognito, el cual permite administrar usuarios de la aplicación, poder realizar un seguimiento, autenticación y registro de eventos relacionados con el usuario, también puede ayudar con la seguridad del sistema realizando un análisis en tiempo real, y determinar si algún intento de autenticación pudiera ser peligroso. AWS provee un sistema por el cual se puede exponer al internet un API, por medio de la herramienta API Gateway, la cual trabaja en conjunto con balanceadores de carga, instancias de EC2, grupos de seguridad, subredes, servidores NAT, para llevar a cabo el funcionamiento correcto del proyecto.



## **10. FUNCIONALIDADES**

### **10.1. Aplicación móvil**

La aplicación móvil cuenta con funcionalidades básicas e intuitivas, que permiten a los usuarios registrarse, iniciar sesión, para luego realizar reservaciones en los parqueos disponibles mostrados en el mapa.

#### **10.1.1. Registro**

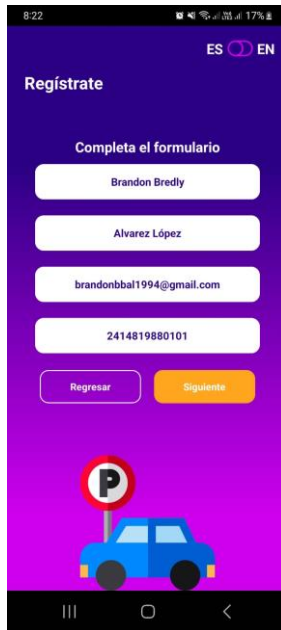
Para poder utilizar la aplicación, es necesario que los usuarios se registren utilizando la aplicación móvil, para esto es necesario que posean un correo activo.

##### **10.1.1.1. Datos personales**

Se debe ingresar al menos un nombre y un apellido, correo electrónico y su número de CUI de 13 dígitos. El correo electrónico se debe poder acceder, debido a que a este llegará un código de confirmación, y este sirve para poder completar el registro correctamente.

## Figura 2.

### *Pantalla de registro*



*Nota.* Primer formulario en el registro. Elaboración propia, realizado en Figma.

Verificar en la bandeja de entrada o en la bandeja de correos no deseados en su defecto, del correo anteriormente escrito y verificar que se ha recibido el código de verificación, dicho código de verificación tiene una vigencia de 24 horas, y el emisor del correo será bydtesis@gmail.com.

## Figura 3.

### *Código de verificación para registro*

PARKING APP: Código de seguridad.

bydtesis@gmail.com

para mí ▼

Tu nuevo código de seguridad es: 631683

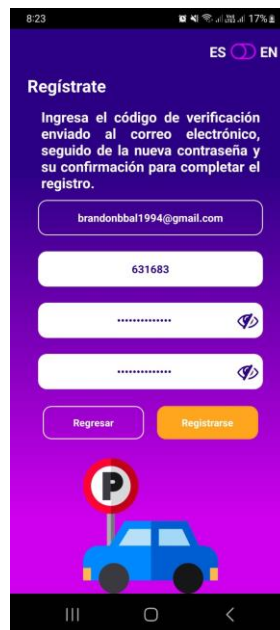
*Nota.* Código de verificación en bandeja de entrada de correo. Elaboración propia, realizado con herramienta de captura de pantalla.

### 10.1.1.2. Confirmar correo electrónico y nueva contraseña

Se debe ingresar el código de verificación de 6 dígitos, la contraseña y repetirla para confirmar, escribiéndola dos veces, para comprobar que se está ingresando la misma contraseña. Esta debe tener al menos 8 caracteres y debe contener mínimo una letra mayúscula y minúscula, un dígito y un carácter especial.

#### Figura 4.

*Ingreso de código de verificación y nueva contraseña*



*Nota.* Segundo formulario en el registro. Elaboración propia, realizado en Figma.

### 10.1.2. Iniciar sesión

Una vez activado el correo electrónico del usuario, se podrá acceder a la aplicación iniciando sesión utilizando el correo y la contraseña, es necesario

ingresar correctamente la contraseña ya que, si se ingresa incorrectamente 4 ocasiones consecutivamente, se bloqueará temporalmente el correo, y deberá esperar 4 horas para poder intentar nuevamente. En esta pantalla de iniciar sesión se podrá encontrar en la parte superior la versión de la aplicación, y esta irá cambiando conforme se añadan nuevas funcionalidades o se corrijan errores.

### Figura 5.

*Inicio de sesión con correo electrónico y contraseña*



*Nota.* Pantalla de inicio de sesión. Elaboración propia, realizado en Figma.

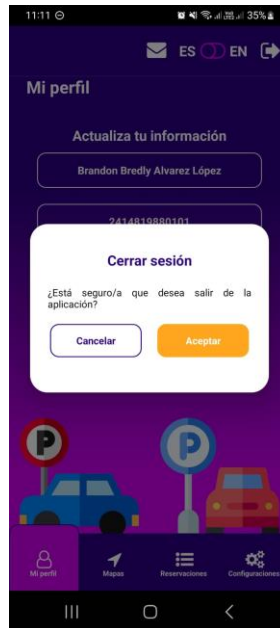
#### 10.1.3. Cerrar sesión

Al estar dentro de la aplicación con una sesión activa, se podrá cerrar sesión en cualquiera de las pantallas principales, se debe confirmar para completar esta acción o se puede cancelar para continuar dentro de la aplicación. Para mostrar esta opción se debe presionar el icono con la flecha de la parte superior derecha.



## Figura 6.

### Cierre de sesión



*Nota.* Confirmación de cierre de sesión. Elaboración propia, realizado en Figma.

### 10.1.4. Recuperar contraseña

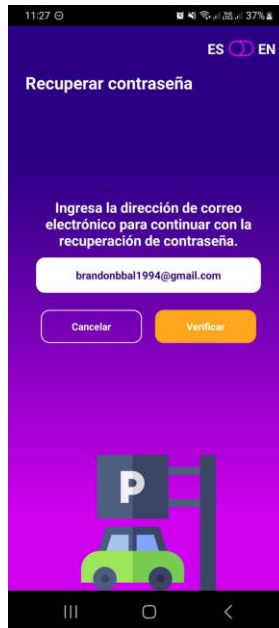
Si el usuario se encuentra activo, y por algún motivo no recuerda la contraseña puede usar la opción de “Olvidé mi contraseña” que se encuentra en la pantalla de inicio de sesión.

#### 10.1.4.1. Verificar correo electrónico

Se debe ingresar el correo electrónico con el que se registró en la aplicación ya que, en este se recibirá el código de verificación de 6 dígitos para crear una nueva contraseña.

## Figura 7.

### Recuperar contraseña de usuario activo



*Nota.* Ingreso de correo electrónico para recuperar contraseña. Elaboración propia, realizado en Figma.

Verificar en la bandeja de entrada o mensajes no deseados, el correo con el código de verificación de 6 dígitos, siendo el emisor del correo `no-reply@verificationemail.com`.

## Figura 8.

### Código para confirmar nueva contraseña

PARKING APP: Código de seguridad

`no-reply@verificationemail.com`

para mí ▾

Tu nuevo código de seguridad es: 189872.

*Nota.* Código de verificación para nueva contraseña. Elaboración propia, realizado con herramienta de captura de pantalla.

#### 10.1.4.2 Crear nueva contraseña

Por último, se debe ingresar el código de verificación, la nueva contraseña y su confirmación para garantizar que se esté escribiendo la misma contraseña y esta debe contener al menos 8 caracteres que tengan al menos una mayúscula y una minúscula, un dígito numérico y algún símbolo especial.

#### Figura 9.

*Confirmar nueva contraseña*



*Nota.* Verificar código y nueva contraseña para iniciar sesión. Elaboración propia, realizado en Figma.

#### 10.1.5. Perfil

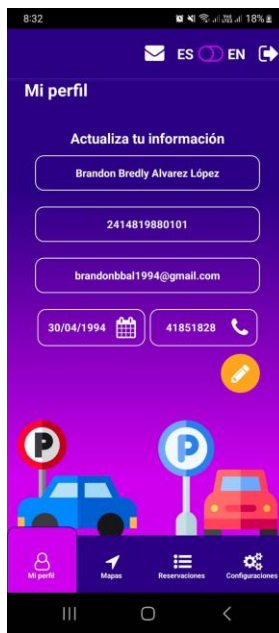
Cada vez que se inicie sesión, esta será la pantalla la que se mostrará por defecto, en la que se podrá visualizar los datos del usuario que inició sesión.

### 10.1.5.1. Actualizar información

El usuario podrá ingresar su fecha de nacimiento, así como su número de teléfono y debe ser de 8 dígitos, esto para poder contactarlo más fácilmente en caso de ser necesario, se debe presionar el icono del lápiz y elegir el campo a modificar.

#### Figura 10.

*Pantalla de perfil, con dato del usuario*



*Nota.* Datos en perfil actualizados. Elaboración propia, realizado en Figma.

### 10.1.6. Chat

Se podrá tener una conversación con mensajes en vivo con un parqueo si se encuentra habilitado únicamente, se puede acceder a ver los parqueos disponibles o conversaciones previas desde cualquiera de las pantallas principales presionando el icono de mensaje en la bandeja superior.

### 10.1.6.1. Visualizar parqueos

Al entrar a la opción de mensajes se mostrará un listado con los parqueos disponibles, y presionando el símbolo de “+”, se mostrará la información del parqueo, y una caja de texto para iniciar una conversación.

**Figura 11.**

*Listado de parqueos disponibles para conversación*



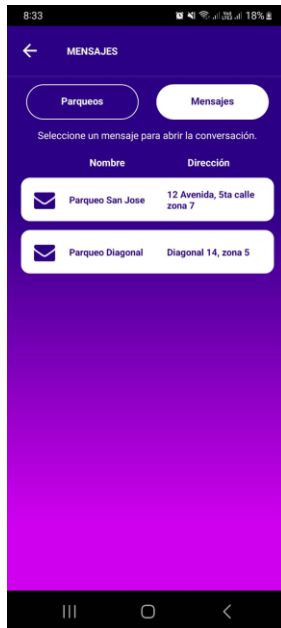
*Nota.* Texto por enviar a Parqueo San José. Elaboración propia, realizado en Figma.

### 2.1.6.2. Visualizar mensajes enviados

Una vez iniciada la conversación se podrán visualizar cada uno de los mensajes enviados, seleccionando la opción de “Mensajes”, en el cual hay que seleccionar un mensaje en específico para abrir la conversación.

## Figura 12.

*Listado de mensajes enviados*



*Nota.* Listado de conversaciones activas e inactivas. Elaboración propia, realizado en Figma.

### 10.1.6.2. Conversación con parqueo

Si se ha iniciado una conversación con un parqueo y este se encuentra habilitado, se toma en cuenta como una conversación activa, pero si el parqueo es dado de baja en algún momento por alguna razón, ya no se podrá continuar la conversación, únicamente se podrán visualizar los mensajes enviados.

### Figura 13.

#### Conversación activa con parqueo



*Nota.* Conversación activa con Parque San José. Elaboración propia, realizado en Figma.

### 10.1.7. Listado de reservaciones

Al presionar la opción “Reservaciones” del menú inferior se mostrará inicialmente todas las reservaciones, no importando si son pagadas, reservadas o canceladas. Se mostrarán las últimas 10 reservaciones, y si hay existen más de 10, se podrá presionar un botón para obtener las siguientes 10 reservaciones, hasta obtener todas.

**Figura 14.**

*Listado de reservaciones*



*Nota.* Listado de todas las reservaciones y su detalle. Elaboración propia, realizado en Figma.

### 10.1.7.1. Reservaciones por tipo

Se tiene un menú superior en el cual se puede elegir qué tipo de reservación se desea obtener, el filtrado se hace por su código de identificación “id”, en forma descendente. Se pueden filtrar por reservaciones pagadas, reservadas o canceladas, y se debe presionar el símbolo de “+” para obtener el detalle completo de cada reservación. Al intercambiar entre opciones de filtrado se reinicia el contador de obtención de últimas 10 reservaciones.



**Figura 15.**

*Filtrado de reservación*



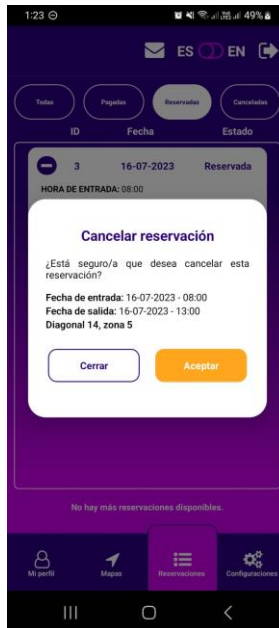
*Nota.* Filtrado de tipo reservadas. Elaboración propia, realizado en Figma.

### 10.1.7.2. Cancelar reservación

Al utilizar el filtro de reservaciones de “Todas” el estado de estas puede ser como “Reservada”, o ubicándose específicamente en la opción del menú superior de “Reservadas”, cada reservación al mostrar el detalle, contiene la opción de poder cancelar dicha reservación, únicamente se pueden cancelar reservaciones que no han sido pagadas.

**Figura 16.**

*Cancelar reservación*



*Nota.* Confirmación para cancelar reservación. Elaboración propia, realizado en Figma.

### **10.1.8. Configuraciones**

En esta opción del menú se puede elegir entre cambiar contraseña, ver los vehículos disponibles para poder crear una reservación, crear o eliminar vehículos, ver los métodos de pago, agregar o eliminar un método de pago, y eliminar cuenta de usuario.

## Figura 17.

### Menú de configuraciones



*Nota.* Opciones disponibles del menú de configuraciones. Elaboración propia, realizado en Figma.

#### 10.1.8.1. Cambiar contraseña

El usuario podrá cambiar su contraseña en cualquier momento, pero es necesario ingresar su contraseña actual, seguido de la nueva contraseña, así como su confirmación para garantizar que se esté ingresando la misma. También debe ser de al menos 8 caracteres, los cuales deben incluir al menos una mayúscula y minúscula, un dígito y un carácter especial.

**Figura 18.**

*Cambio de contraseña*



*Nota.* Formulario con contraseña actual y nueva contraseña. Elaboración propia, realizado en Figma.

### **10.1.8.2. Administrar vehículos**

Al seleccionar esta opción, por defecto se podrán visualizar cada uno de los vehículos del usuario, estos a su vez serán utilizados para elegir uno al momento de hacer una reservación.

#### **10.1.8.2.1. Listado de vehículos**

Al presionar el símbolo de "+", se podrá ver el detalle completo del vehículo seleccionado, y al presionar "-", se ocultará la información.

## Figura 19.

### Listado de vehículos



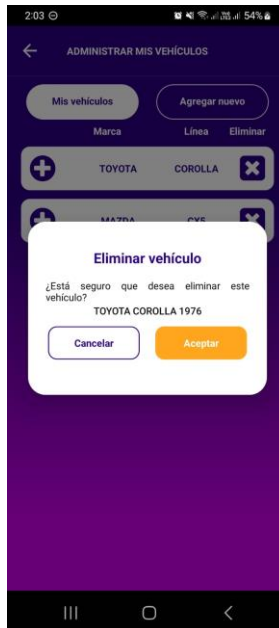
*Nota.* Vehículo disponible para poder crear reservación. Elaboración propia, realizado en Figma.

#### 10.1.8.2.2. Eliminar vehículo

Cada uno de los vehículos disponibles para reservación se pueden eliminar, esto se hace presionando el símbolo "X", lo cual muestra una opción de confirmación y cancelar eliminación de vehículo. Si se acepta la eliminación del vehículo, ya no se podrá hacer una reservación utilizando este vehículo.

## Figura 20.

### Eliminar vehículo



*Nota.* Confirmación de eliminación de vehículo Toyota. Elaboración propia, realizado en Figma.

### 10.1.8.2.3. Agregar vehículo

Al seleccionar la opción de “Agregar nuevo”, se mostrará un formulario el cual debe llenarse con la información del vehículo, que debe ser marca del vehículo, línea a la que pertenece, color, año o modelo y número de placas particulares en el caso de vehículos convencionales o matrícula para las motos.

## Figura 21.

*Agregando nuevo vehículo*



The screenshot shows a mobile application interface with a purple-to-pink gradient background. At the top, there is a status bar with the time 1:54 and a battery level of 53%. Below the status bar is a header with a back arrow and the text "ADMINISTRAR MIS VEHÍCULOS". Underneath the header are two buttons: "Mis vehículos" and "Agregar nuevo". The "Agregar nuevo" button is highlighted. Below these buttons is a section titled "Agregar nuevo vehículo". This section contains four buttons: "Toyota", "Naranja", "Corolla", and "1976". Below these buttons is a text input field containing "P048dtr". At the bottom of the form are two buttons: "Cancelar" and "Guardar". The "Guardar" button is highlighted in orange. The bottom of the screen shows the Android navigation bar with three icons: a square, a circle, and a triangle.

*Nota.* Formulario con datos de nuevo vehículo Toyota. Elaboración propia, realizado en Figma.

Al completar el formulario, se deben guardar los cambios, previamente se tiene una visualización con los datos para corroborar la información ingresada.

## Figura 22.

Confirmando nuevo vehículo



*Nota.* Detalle de nuevo vehículo previo a guardarlo. Elaboración propia, realizado en Figma.

### 10.1.8.3. Método de pago

El único método de pago disponible actualmente para la aplicación móvil será con cuentas bancarias, se podrá elegir entre cuentas monetarias y cuentas de ahorro, y cuenta bancaria principal o secundaria. La cuenta principal será la utilizada para hacer los pagos de reservaciones.

#### 10.1.8.3.1. Listado de cuentas bancarias

Al presionar el símbolo de “+” se puede ver el detalle completo de la cuenta bancaria, sin embargo, únicamente se pueden observar los últimos dígitos del número de la cuenta de banco.



## Figura 23.

### Listado de cuentas bancarias



*Nota.* Cuenta bancaria principal y secundaria. Elaboración propia, realizado en Figma.

#### 10.1.8.3.2. Eliminar cuenta bancaria

Las cuentas bancarias que no sean principales serán las únicas que se podrán eliminar y esto se hace presionando el símbolo de “X”, el cual muestra una confirmación previa a su eliminación definitiva.

## Figura 24.

*Confirmando eliminación de cuenta bancaria*



*Nota.* Eliminando cuenta bancaria secundaria. Elaboración propia, realizado en Figma.

### 10.1.8.3.3. Agregar cuenta bancaria

Si es la primera cuenta bancaria que se agrega, por defecto será la cuenta bancaria principal, por lo que el usuario no podrá modificar el valor de cuenta principal. Se debe seleccionar si es cuenta bancaria de ahorro o monetaria, y agregar el número de cuenta bancaria. Si es la segunda cuenta bancaria se podrá elegir como cuenta secundaria o cuenta principal, por lo que si se elige como cuenta principal, la que anteriormente era cuenta principal pasará a ser cuenta secundaria.

## Figura 25.

### *Formulario de nueva cuenta bancaria*



The screenshot shows a mobile application interface with a purple-to-pink gradient background. At the top, there is a status bar with the time 2:30 and battery level 57%. Below the status bar is a navigation bar with a back arrow and the text "MÉTODO DE PAGO". Underneath are two buttons: "Mis cuentas" and "Agregar". The main content area is titled "Completa el formulario:" and contains a text input field with the value "500600700". Below the input field are two radio button options: "Cta. ahorro" (unselected) and "Cta. monetaria" (selected). Below these are two more radio button options: "No" (unselected) and "Si" (selected) for the question "¿Cuenta principal?". At the bottom of the form are two buttons: "Cancelar" and "Guardar". The bottom of the screen shows the standard Android navigation bar with three icons: a square, a circle, and a triangle.

*Nota.* Completando formulario con nueva cuenta bancaria principal. Elaboración propia, realizado en Figma.

Al presionar “Guardar”, se muestra una vista previa con la información de la nueva cuenta bancaria antes de guardarla de forma definitiva.

## Figura 26.

*Confirmando nueva cuenta bancaria*



*Nota.* Vista previa de nueva cuenta bancaria principal. Elaboración propia, realizado en Figma.

### 10.1.8.4. Eliminar usuario

Última opción del menú de configuraciones que permite desactivar al usuario, mostrando previamente una confirmación, la cual debe ser aceptada por el usuario, por lo que, si el usuario acepta, debe hacer el proceso de registrarse nuevamente en la aplicación.

## Figura 27.

*Confirmación de eliminación de cuenta de usuario*



*Nota.* Vista previa de eliminación de usuario de aplicación. Elaboración propia, realizado en Figma.

### 10.1.9. Mapa

Se utiliza API de Google para poder mostrar cada una de las ubicaciones en tiempo real de los parqueos, que se encuentren más cercanas a la ubicación del usuario. También se puede visualizar la ubicación de parqueos en las que se han realizado reservaciones.

#### 10.1.9.1. Parqueos

Esta funcionalidad requiere de la utilización de la geolocalización del dispositivo, por lo que debe ser activado presionando el ícono en la barra de buscador de parqueos por texto y aceptando el permiso de ubicación, o

accediendo manualmente a la configuración de la aplicación, activando el permiso.

## Figura 28.

### Mapa de parqueos

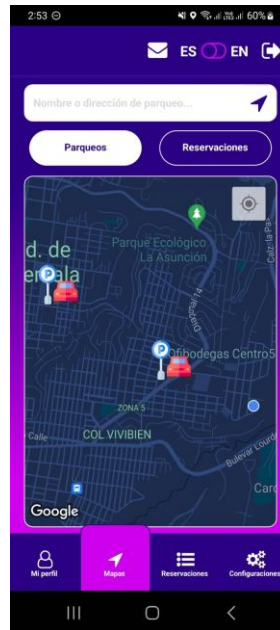


*Nota.* Mapa de parqueos con geolocalización desactivada. Elaboración propia, realizado en Figma.

Al tener habilitado el permiso de geolocalización en el dispositivo se mostrará el mapa con los parqueos cercanos disponibles para reservar. Se podrá visualizar su ubicación real en el mapa.

## Figura 29.

### Mapa de parqueos disponibles



*Nota.* Parqueos cercanos a la ubicación del usuario. Elaboración propia, realizado en Figma.

Al seleccionar un parqueo se mostrará un formulario el cual contiene la información del parqueo, así como los datos necesarios para poder realizar una reservación. La fecha de entrada será la misma fecha de salida, por lo que, cambiará será la hora de entrada y hora de salida. El intervalo de tiempo de selección de minutos es de 15 minutos, y se tomará en cuenta la fracción dependiendo del precio por hora del parqueo. Se debe seleccionar el vehículo, dentro del listado de vehículos disponibles, y elegir si desea pagar la reserva o no.

La ecuación para calcular el total a pagar se realizó de la siguiente forma:

$$T = [PH * (HF - HI)] + [PH * (MF - MI)] / 60 \quad (1)$$

Cada una de las variables se detalla en la siguiente tabla, si y solo sí la hora de salida es mayor o igual a la hora de entrada y los minutos de salida son mayores a los minutos de entrada, durante la misma hora o minuto de salida es mayor o igual al minuto de salida cuando la hora de salida es mayor a la hora de entrada:

**Tabla 1.**

*Descripción de variables en ecuación 1 para calcular precio de reservación*

<b>Variable</b>	<b>Descripción</b>
<b>T</b>	Precio total para pagar en quetzales
<b>PH</b>	Precio por hora del parqueo
<b>HF</b>	Hora de salida seleccionada por el usuario
<b>HI</b>	Hora de entrada seleccionada por el usuario
<b>MF</b>	Minuto de salida seleccionada por el usuario
<b>MI</b>	Minuto de salida seleccionada por el usuario

*Nota.* Variables para cálculo de distancia. Elaboración propia, realizado con Microsoft Word.



## Figura 30.

### Crear reservación



The screenshot shows a mobile application interface for creating a reservation. The screen is titled "Crear reservación" and features a purple header with language options "ES" and "EN". The form includes the following fields and options:

- Nombre parqueo:** Parqueo Diagonal
- Total a pagar:** Q.30.00
- Dirección parqueo:** Diagonal 14, zona 5
- Fecha:** 13/09/2023
- Vehículo:** MAZDA-P56K
- Hora entrada:** 07:00
- Hora salida:** 09:00

Below the form, there is a checkbox labeled "¿Desea pagar el monto total de la reservación?" which is checked. Underneath, there are two lines of explanatory text:

- Si no paga el monto de la reservación y elimina el método de pago configurado, la reservación será cancelada.
- Si paga ésta reservación, podrá realizar una nueva reservación.

At the bottom of the form are two buttons: "Cancelar" and "Reservar". The bottom navigation bar of the app includes icons for "Mi perfil", "Mapas", "Reservaciones", and "Configuraciones".

*Nota.* Reservación por crear con 2 horas de diferencia. Elaboración propia, realizado en Figma.

Si se paga la reserva, se mostrará el detalle de esta, al finalizar el pago, de forma automática.

**Figura 31.**  
*Reservación creada*



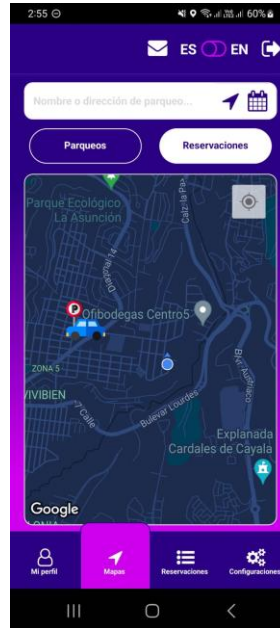
*Nota.* Detalle de reservación al finalizar pago. Elaboración propia, realizado en Figma.

### 10.1.9.2. Reservaciones

Al igual que la funcionalidad de parqueos, esta también requiere del permiso de geolocalización del dispositivo, y se activa de la misma forma. Acá se pueden filtrar por nombre de parqueos utilizando el buscador de parqueos.

## Figura 32.

### *Reservaciones por día*



*Nota.* Reservaciones con fecha actual. Elaboración propia, realizado en Figma.

También se puede utilizar el filtro por fecha de reserva. Al filtrar por fecha, la fecha inicial será el día actual, pero se puede cambiar seleccionando el ícono del calendario ubicado en el buscador.

### Figura 33.

*Filtrar reservaciones por fecha*



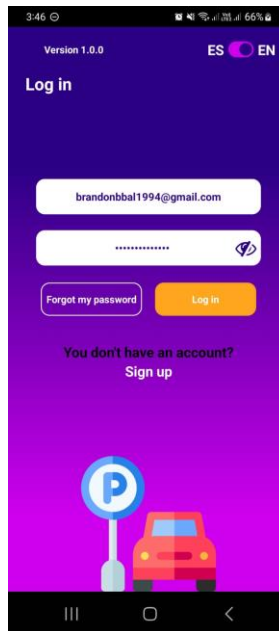
*Nota.* Filtrando reservaciones con fecha 13 de septiembre 2023. Elaboración propia, realizado en Figma.

#### 10.1.10. Configuración de idioma

Personas que no hablen español, pero tengan conocimiento del idioma inglés también podrán utilizar la aplicación ya que, se encuentra disponible en 2 idiomas el cual, puede ser cambiado en cualquier momento.

## Figura 34.

*Cambio de idioma en aplicación.*



*Nota.* Cambio a idioma inglés en pantalla de inicio de sesión. Elaboración propia, realizado en Figma.

## 10.2. Consola de administración

En la consola de administración se encuentran todas las herramientas con el cual un dueño de parqueo podrá administrar todos los pagos, parqueos, reservaciones y demás servicios.

### 10.2.1. Pantalla de inicio de sesión

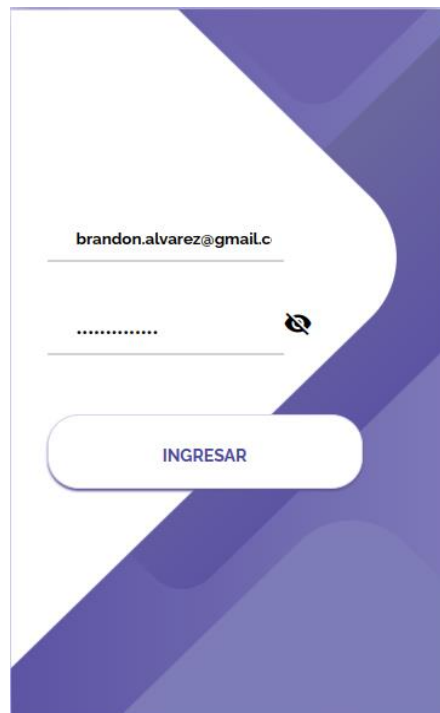
Pantalla en la que el usuario utilizará su correo y contraseña por medio del cual podrá acceder a las demás configuraciones para poder administrar sus parqueos, la autenticación se realiza por medio de dos partes, la primera se realiza con Cognito, el cual se encarga de validar que los datos sean correctos y

generar un token de sesión válido y retornar parámetros que se configuraron en el usuario, al momento que Cognito valida al usuario, se utiliza uno de los parámetros, para poder obtener la información completa del usuario almacenados en la base de datos.

Esta pantalla solo se podrá visualizar si el usuario no se ha autenticado previamente y si el token de sesión es válido aún, los tokens de sesión tienen un tiempo aproximado de un día, se realizó la configuración de manera que la sesión solamente sea válido en una pestaña del navegador, si el usuario cierra la pestaña o trata de ingresar desde otra pestaña del navegador, deberá de autenticarse nuevamente ingresando su correo y contraseña.

### **Figura 35.**

*Inicio de sesión*



*Nota.* Inicio de sesión en consola de administración. Elaboración propia, realizado en Figma.

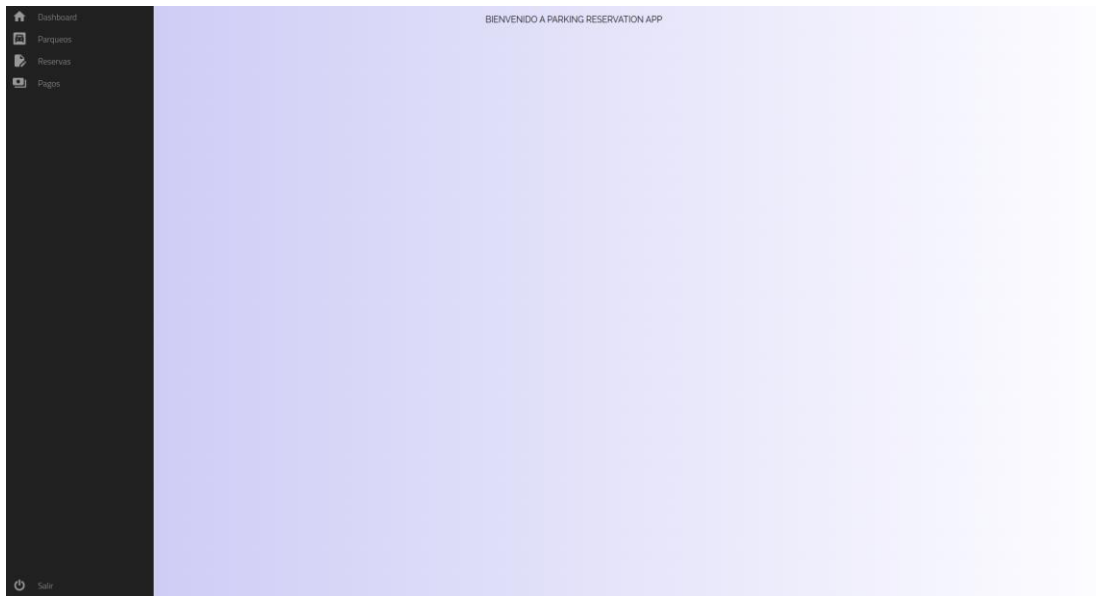
### 10.2.2. Pantalla inicial

Al momento de un usuario haber ingresado su correo y contraseña válidos, el sistema realizará una redirección a la pantalla inicial del sistema de configuración de parqueo, en esta pantalla se puede visualizar un menú en la parte izquierda con diferentes opciones:

- Administración de parqueos
- Administración de reservas
- Administración de pagos

#### Figura 36.

*Pantalla inicial en consola de administración*



*Nota.* Pagina inicial consola de administración. Elaboración propia, realizado en Figma.

Estas configuraciones permiten de manera muy general administrar su parqueo, ya sea haciendo inhabilitaciones, validando datos, creación de parqueos, actualización de datos, validar reservaciones, etc.

### 10.2.3. Administración de parqueos

Se habilitará una pantalla en la cual los usuarios podrán inicialmente visualizar todos los parqueos que estén relacionados a él, en esta lista de parqueos, se puede realizar paginación, búsqueda por nombre, selección de cantidad de ítems a visualizar.

**Figura 37.**

*Listado de parqueos de usuario*



Id.	Nombre	Lugar	Latitud	Longitude	Precio	Estado	
4	Parqueo 3 usuario prueba 2	Ubicaicon de prueba	154.45	0.1564	10	Habilitado	⋮
3	2 usuario prueba 2 editado	Ubicacion prueba editado	0.214444	0.154444	13.51	Habilitado	⋮
1	Parqueo 1 de Usuario 2	Ubicación en texto	0.1545	0.4567	15	Habilitado	⋮

Items per page: 5 1 - 3 of 3 < >

*Nota.* Listado de parqueos que pertenecen a un usuario. Elaboración propia, realizado en Figma.

Esta pantalla muestra los distintos datos que un parqueo necesita al momento de ser registrado, como son:

- Id: identificador autogenerado al momento de crear un parqueo
- Nombre: Dato de tipo texto el cual se le mostrará al usuario en la aplicación.
- Lugar: Dato con la descripción de la ubicación del parqueo.
- Latitud y longitud: datos para localizar un parqueo.



- Precio: Dato de tipo numérico, el cual representa el costo por hora de un parqueo.
- Estado

La pantalla permite que por cada uno de los parqueos se puedan realizar acciones en el parqueo que se desee realizar, siendo estas, visualizar detalle, editar, desactivar en caso de que esté habilitado, y deshabilitar en caso que se encuentre deshabilitado.

### Figura 38.

#### Acciones habilitadas para un parqueo

Mis parqueos Nuevo +

Búsqueda ^

Nombre de lugar Filtrar

Id.	Nombre	Lugar	Latitud	Longitud	Precio	Estado
4	Parqueo 3 usuario prueba 2	Ubicaicon de prueba	154.45	0.1564	10	Habilitado
3	2 usuario prueba 2 editado	Ubicacion prueba editado	0.214444	0.154444	1351	Habilitado
1	Parqueo 1 de Usuario 2	Ubicación en texto	0.1545	0.4567	15	Habilitado

Items per page: 5 1 - 3 of 3

- Ver
- Editar
- Desactivar

*Nota.* Acciones que puede realizar sobre un parqueo. Elaboración propia, realizado en Figma.

### 10.2.3.1 Creación de parqueo

Para poder acceder a esta opción se debe de presionar el botón “Nuevo +”, el cual hará una redirección al formulario en el cual el usuario puede ingresar los datos necesarios para la creación del parqueo.

#### Figura 39.

*Creación de parqueo*

El formulario de creación de un nuevo parqueo se muestra en un recuadro con un fondo púrpura claro. En la parte superior izquierda del recuadro hay un botón azul con el texto "Crear". El formulario mismo tiene un título "Parqueo" y un subtítulo "Datos de parqueo". Debajo de esto hay cinco campos de texto con líneas horizontales para escribir, cada uno etiquetado con un asterisco que indica que es obligatorio: "Ubicación \*", "Latitud \*", "Longitud \*", "Precio por hora \*" y "Nombre \*".

*Nota.* Creación de un nuevo parqueo. Elaboración propia, realizado en Figma.

Al momento del usuario terminar de ingresar los datos, puede presionar el botón “Crear” el cual enviará una petición de tipo POST, al servidor el cual validará que todos los datos sean correctos, y creará el registro en base de datos, para que pueda ser obtenido por cualquier usuario dentro de la aplicación.

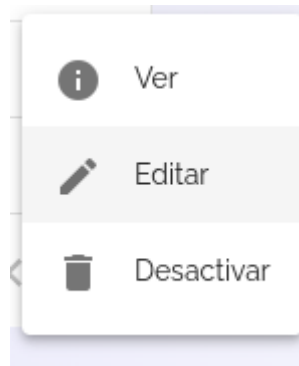
En caso de que la información no sea correcta, se mostrará una notificación indicando que los datos del formulario se deben validar, por lo que no se creará el registro del nuevo parqueo en base de datos.

### 10.2.3.2. Edición de parqueo

Los usuarios dentro de la consola podrán seleccionar editar la información de un parqueo.

#### Figura 40.

*Edición de parqueo*



*Nota.* Proceso para editar parqueo. Elaboración propia, realizado en Figma.

Al momento de presionar el botón “Editar” serán redirigidos a un formulario similar que el que se utiliza para crear un parqueo, pero este se encontrara pre rellenado con la información del parque, lista para ser editada.

**Figura 41.**

*Datos para editar parqueo*



Guardar

**Parqueo**  
Datos de parqueo

Ubicación \*  
Ubicaicon de prueba

Latitud \*  
154.45

Longitud \*  
0.1564

Precio por hora \*  
10

Nombre \*  
Parqueo 3 usuario prueba 2

*Nota.* Edición de parqueo. Elaboración propia, realizado en Figma.

El usuario podrá modificar cualquiera de los datos que se le mostraran. Al momento de presionar guardar, si todos los datos son correctos, se realizará una actualización dentro de la base de datos, si los datos llegarán a ser incorrectos, la información no se actualizará y se mostrará una notificación indicando que existe un error en los datos actualizados.

### **10.3.3.3. Activar y desactivar parqueos**

Los usuarios podrán activar o desactivar los parqueos de forma sencilla, si un parqueo se encuentra activado, se mostrará una opción que indica que se puede deshabilitar, al momento de presionar deshabilitar, se mostrará un

formulario con la información del parqueo, pero sin poderse editar, el usuario podrá revisar si es correcto el parqueo que desea deshabilitar, y presionar nuevamente el botón deshabilitar. Al momento de desactivar un parqueo, este ya no se mostrará dentro de la aplicación hasta que se habilite nuevamente.

**Figura 42.**

*Desactivar parqueo*



*Nota.* Desactivación de parqueo. Elaboración propia, realizado en Figma.

Si un comercio está deshabilitado ya no se mostrará la opción de deshabilitar, el botón que se podrá visualizar será el de deshabilitar, al presionar esta opción se mostrará la información del comercio que se desea habilitar, y de confirmar que se desea habilitar, se modifica en base de datos, y ya se podrá visualizar dentro de la aplicación.

**Figura 43.**

*Restaurar parqueo deshabilitado*

Restaurar

Parqueo

Datos de parqueo

Ubicación  
Ubicación de prueba

Latitud  
154.45

Longitud  
0.1564

Precio por hora  
10

Nombre  
Parqueo 3 usuario prueba 2

Estado  
Deshabilitado

Nota. Restaurando parqueo deshabilitado. Elaboración propia, realizado en Figma.

## 10.2.4. Reservaciones

Los usuarios en la aplicación móvil podrán realizar búsqueda de parqueos, y de ser conveniente para ellos, generar una reservación con fecha y hora del parqueo. Las reservaciones están relacionadas a un vehículo en específico, así como a un método de pago que el usuario ya debe tener previamente agregado.

### Figura 44.

#### Listado de reservaciones



Id.	Fecha/hora de inicio	Fecha/hora de fin	Parqueo	Vehiculo	Estado
3	2023-07-16 08:00:00	2023-07-16 13:00:00	1-Parqueo 1 de Usuario 2	ACR763 Mazda 2013 Blanco	Pagado
1	2023-08-15 10:38:02	2023-08-15 12:38:09	1-Parqueo 1 de Usuario 2	ACR763 Mazda 2013 Blanco	Pagado

*Nota.* Listado de reservaciones. Elaboración propia, realizado en Figma.

Los usuarios podrán realizar búsqueda de reservaciones por placa de vehículo para un filtrado más rápido, también podrán ver el detalle de la reservación y si una reservación aún se encuentra en estado pendiente, podrán deshabilitarse, para que la reservación ya no sea válida. Al momento de que una reservación sea pagada o deshabilitada, ya no se podrá modificar el estado de esta.

### 10.2.4.1. Desactivar reservación

Al momento de querer deshabilitar una reservación se mostrará la información detallada de la reservación, y el usuario podrá confirmar que desea

deshabilitar la reservación, una vez confirmado, se realizará la modificación en base de datos, y el usuario ya no podrá ver dicha reservación en su aplicación móvil.

### Figura 45.

*Desactivar reservaciones.*



*Nota.* Deshabilitar reservaciones. Elaboración propia, realizado en Figma.

### 10.2.3. Pagos

Los usuarios podrán visualizar los pagos recibidos de cualquiera de sus reservaciones, podrán realizar búsquedas por identificador de reservación.



**Figura 46.**

*Listar pagos*

Id.	Total	Id Reservación	Id Cliente	Estado	
3	154.45	3	3	Pagado	⋮
1	300	1	2	Habilitado	⋮

Items per page: 5 1 - 2 of 2 < >

*Nota.* Listar pagos recibidos de un parqueo. Elaboración propia, realizado en Figma.

Los usuarios tienen la opción de visualizar el detalle de sus pagos.

**Figura 47.**

*Detalle de pagos*

**Pagos**  
Datos de pagos

Monto total  
154.45

Estado  
Pagado

Reservación  
3

Id de cliente  
3

*Nota.* Detalle de pago recibido. Elaboración propia, realizado en Figma.



## 11. ESTRUCTURA DEL PROYECTO

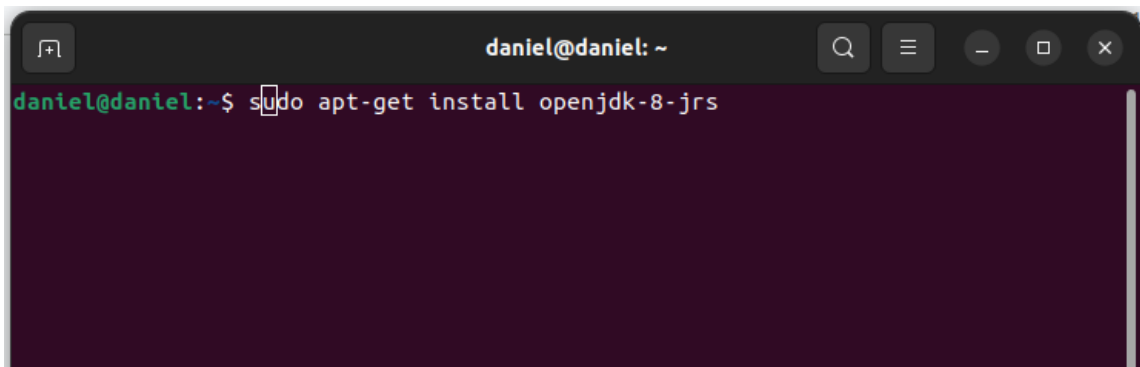
### 11.1. Configuración de ambiente de desarrollo para backend

El microservicio desarrollado en Springboot junto con Java requiere de varios factores necesarios para poder ejecutarse. El proyecto es ejecutado usando la versión de Java 8, una conexión a base de datos, y diferentes paquetes instalados por medio de maven.

El proceso de instalación de Java en un servidor de linux inicia desde la consola, ejecutando la instalación con el comando `sudo apt-get install openjdk-8-jre`.

#### Figura 48.

*Instalación jdk de java*

A screenshot of a terminal window with a dark background. The window title is 'daniel@daniel: ~'. The prompt is 'daniel@daniel:~\$' and the command 'sudo apt-get install openjdk-8-jrs' is being typed. The terminal shows the command being entered, with a cursor at the end of the line. The window has standard Linux window controls (minimize, maximize, close) and a search icon in the top right corner.

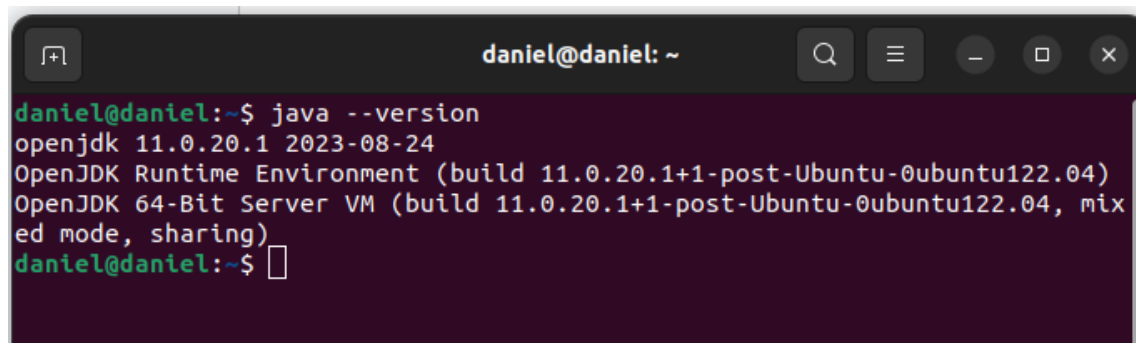
*Nota.* Comando para instalar jdk desde terminal. Elaboración propia, realizado con herramienta de recortes.

Una vez ejecutado el comando, se mostrará el espacio de almacenamiento que se utilizara, y comenzará su instalación. Para poder utilizar Java de manera global en el sistema se debe de crear la variable `export JAVA_HOME=path_to_java_home` luego de esto Java se podrá utilizar de manera general.

Luego podremos ejecutar el comando `java --version`, si la instalación se realizó de forma correcta, mostrará la versión que se instaló y se encuentra funcionando en el servidor.

#### **Figura 49.**

*Validación de versión de java*

A screenshot of a terminal window with a dark background. The window title is "daniel@daniel: ~". The terminal shows the command "java --version" being executed, resulting in the following output: "openjdk 11.0.20.1 2023-08-24", "OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04)", and "OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)". The prompt "daniel@daniel:~\$" is visible at the end of the output.

```
daniel@daniel:~$ java --version
openjdk 11.0.20.1 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
daniel@daniel:~$
```

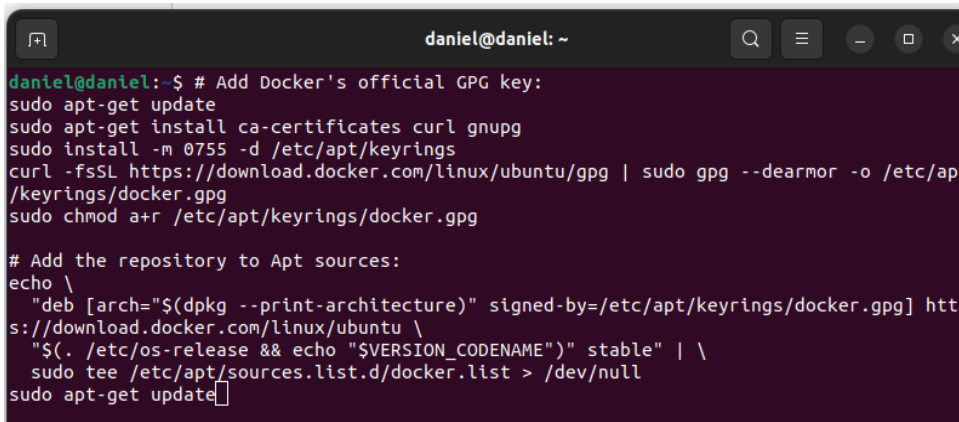
*Nota.* Código para validar versión de Java. Elaboración propia, realizado con herramienta de recortes.

El microservicio debe poseer una conexión directa a la base de datos, por lo que se debe poseer una base de datos dedicada al proyecto. Se utilizó el sistema de Docker para ejecutar el sistema de base de datos, sin la necesidad de tener configurada una base de datos ejecutándose directamente en nuestro servidor.

La instalación del sistema de contenedores Docker debe ser instalado en el servidor donde se desee ejecutar dichos contenedores, la instalación de Docker se puede realizar desde una terminal, agregando la configuración del repositorio de Apt que contiene los paquetes de Docker, ejecutando los comandos:

### Figura 50.

#### *Configuración de repositorio para docker*

A terminal window with a dark background and light text. The prompt is 'daniel@daniel: ~'. The user enters several commands to set up the Docker repository: adding the GPG key, updating apt, installing certificates and gnupg, downloading the Docker GPG key from the official website, and adding the repository to the sources list. The commands are: 'sudo apt-get update', 'sudo apt-get install ca-certificates curl gnupg', 'sudo install -m 0755 -d /etc/apt/keyrings', 'curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg', 'sudo chmod a+r /etc/apt/keyrings/docker.gpg', '# Add the repository to Apt sources:', 'echo \'', '"deb [arch=\$(dpkg --print-architecture)] signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \'', '"\$(. /etc/os-release && echo "\$VERSION\_CODENAME")" stable" | \'', 'sudo tee /etc/apt/sources.list.d/docker.list > /dev/null', and 'sudo apt-get update'.

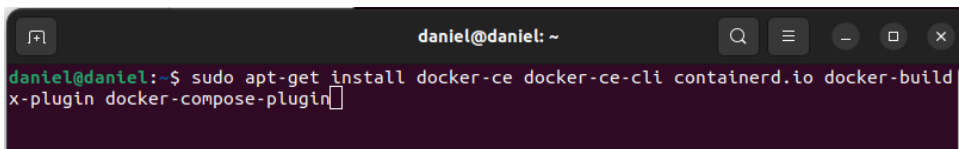
```
daniel@daniel: ~  
daniel@daniel:~$ # Add Docker's official GPG key:  
sudo apt-get update  
sudo apt-get install ca-certificates curl gnupg  
sudo install -m 0755 -d /etc/apt/keyrings  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg  
sudo chmod a+r /etc/apt/keyrings/docker.gpg  
  
# Add the repository to Apt sources:  
echo \  
"deb [arch=$(dpkg --print-architecture)] signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \  
"$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

*Nota.* Código para configurar repositorio de docker. Elaboración propia, realizado con herramienta de recortes.

Luego se debe de realizar la instalación de los paquetes de Docker.

### Figura 51.

#### *Instalación de docker*

A terminal window with a dark background and light text. The prompt is 'daniel@daniel: ~'. The user enters the command 'sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin'.

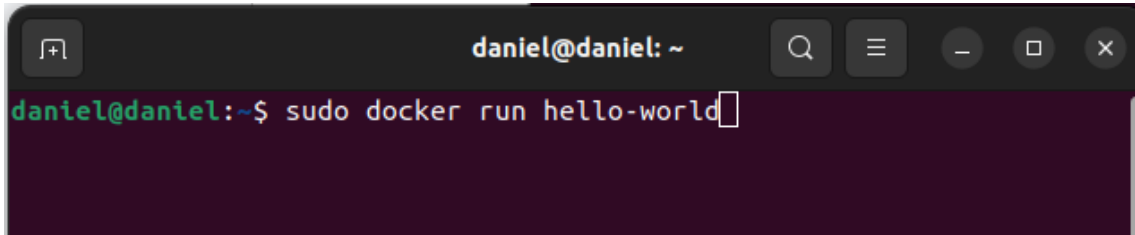
```
daniel@daniel: ~  
daniel@daniel:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

*Nota.* Código para instalar Docker. Elaboración propia, realizado con herramienta de recortes.

Por último, para validar que Docker se encuentre funcionando de forma correcta se puede realizar una prueba ejecutando una imagen básica del sistema.

## Figura 52.

*Ejecutando comando de hola mundo*

A terminal window with a dark purple background. The title bar shows 'daniel@daniel: ~'. The prompt is 'daniel@daniel:~\$' and the command 'sudo docker run hello-world' is being typed, with a white cursor at the end of the line.

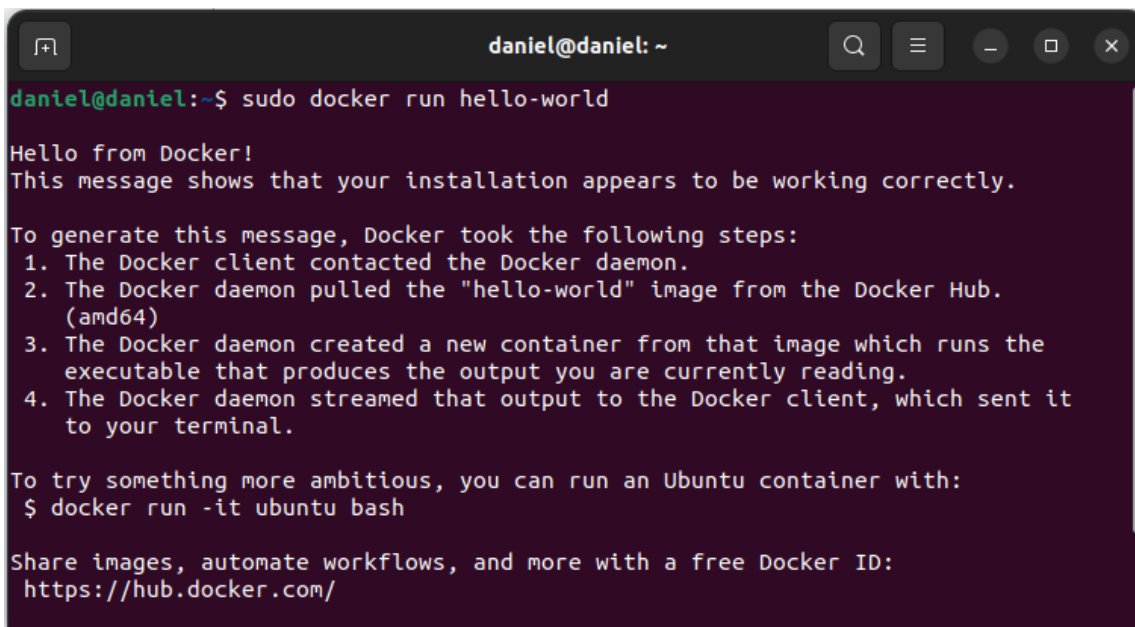
```
daniel@daniel:~$ sudo docker run hello-world
```

*Nota.* Código para validar que Docker se instaló. Elaboración propia, elaborado con herramienta de recortes.

La que nos mostrará un mensaje indicando el éxito en su ejecución.

## Figura 53.

*Mensaje de éxito con docker*

A terminal window with a dark purple background. The title bar shows 'daniel@daniel: ~'. The prompt is 'daniel@daniel:~\$' and the command 'sudo docker run hello-world' has been executed. The output is displayed in white text.

```
daniel@daniel:~$ sudo docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
```

*Nota.* Mensaje de éxito de Docker. Elaboración propia, realizado con herramienta de recortes.

Una vez configurado Docker, podemos ingresar el comando por el cual se levantará una imagen de mysql con la versión más reciente, la cual se ejecutará en el puerto 33060, por la cual se podrá acceder a la base de datos utilizando alguna herramienta como podría ser Jira y Mysql workbench.

### Figura 54.

*Creación de contenedor*

A terminal window titled 'daniel@daniel: ~' with search, menu, and window control icons. The command entered is 'sudo docker run -d -p 33060:3306 --name mysql-db -e MYSQL\_ROOT\_PASSWORD=test123 mysql'. The prompt is followed by a cursor. The terminal background is dark purple.

```
daniel@daniel:~$ sudo docker run -d -p 33060:3306 --name mysql-db -e MYSQL_ROOT_PASSWORD=test123 mysql
```

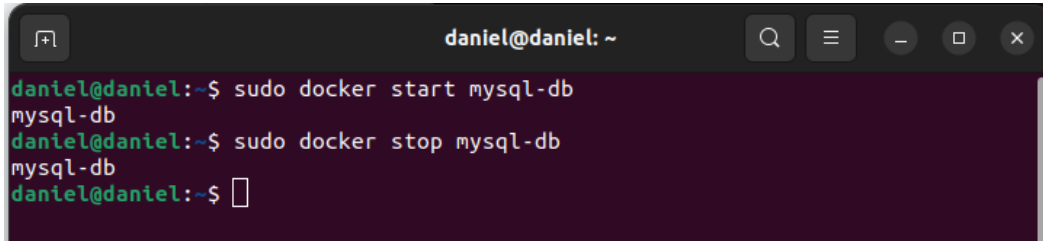
*Nota.* Ejecución de un contenedor. Elaboración propia, realizado con herramienta de recortes.

El comando indica que puerto es el que se ejecutará y a cual puerto del contenedor está apuntando, qué nombre se le asignará al contenedor, contraseña que se utilizara en la base de datos.

Muchas veces es necesario pausar o ejecutar de nuevo el contenedor, por lo que es importante recalcar dos comandos, que nos permitirán realizar estas acciones.

**Figura 55.**

*Iniciando y parando contenedor*



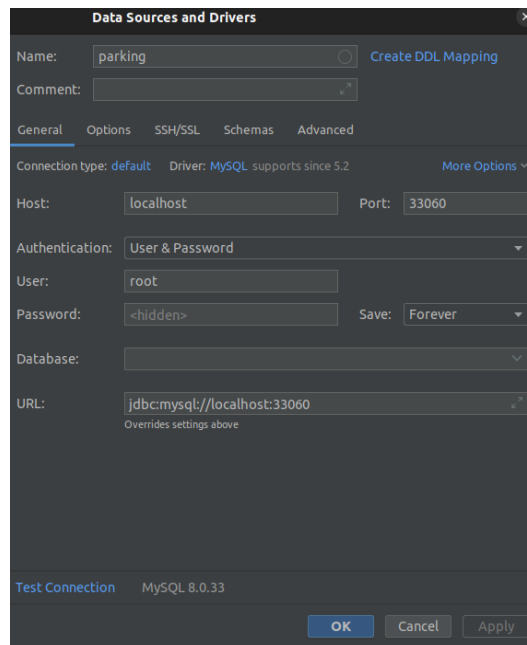
```
daniel@daniel: ~  
daniel@daniel:~$ sudo docker start mysql-db  
mysql-db  
daniel@daniel:~$ sudo docker stop mysql-db  
mysql-db  
daniel@daniel:~$
```

*Nota.* Inicialización y finalización de contenedor. Elaboración propia, realizado con herramienta recortes.

Una vez configurado el sistema de base de datos, podremos conectarnos a ella, la configuración necesaria para la herramienta DataGrip consiste en indicar puerto, usuario, dirección de servidor y contraseña de la base de datos.

**Figura 56.**

*Configuración de conexión a base de datos*



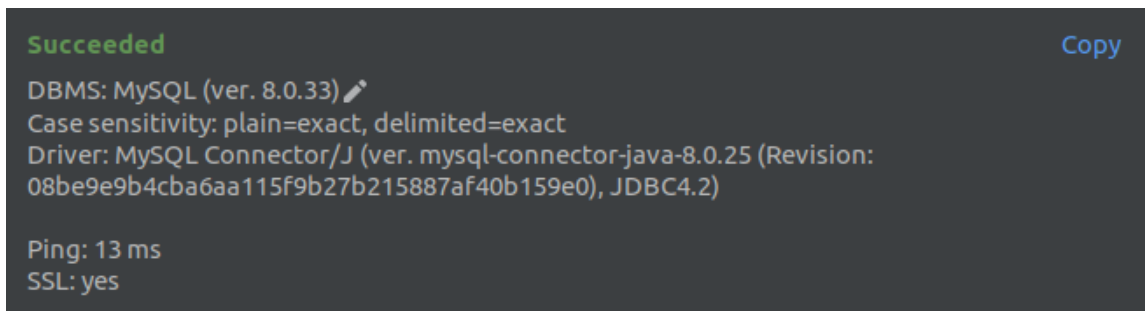
*Nota.* Datos para conexión a base de datos. Elaboración propia, realizado con herramienta de recortes.



Si la configuración se realizó de forma exitosa, la herramienta nos mostrará un mensaje de éxito al presionar Test Connection.

### Figura 57.

*Mensaje de conexión exitosa*

A screenshot of a dark-themed notification box. At the top left, the word "Succeeded" is written in green. At the top right, there is a blue "Copy" button. The main text is white and provides technical details about the database connection: "DBMS: MySQL (ver. 8.0.33) ✎", "Case sensitivity: plain=exact, delimited=exact", "Driver: MySQL Connector/J (ver. mysql-connector-java-8.0.25 (Revision: 08be9e9b4cba6aa115f9b27b215887af40b159e0), JDBC4.2)", "Ping: 13 ms", and "SSL: yes".

```
Succeeded Copy  
DBMS: MySQL (ver. 8.0.33) ✎  
Case sensitivity: plain=exact, delimited=exact  
Driver: MySQL Connector/J (ver. mysql-connector-java-8.0.25 (Revision:  
08be9e9b4cba6aa115f9b27b215887af40b159e0), JDBC4.2)  
  
Ping: 13 ms  
SSL: yes
```

*Nota.* Mensaje de éxito. Elaboración propia, realizado con herramienta de recortes.

Una vez configurado Java y nuestra base de datos, podremos visualizar nuestro proyecto. La forma más fácil de inicializar un proyecto de Springboot es por medio de la página Spring Initializer, la que nos permite realizar configuraciones del tipo de proyecto, lenguaje que se utilizara, versión de Spring boot que se utilizara, tipo de empaquetado y versión de Java que se utilizara en el proyecto.

**Figura 58.**

*Inicializar proyecto de springboot*

The image shows a dark-themed configuration window for a Spring Boot project. It is divided into several sections:

- Project:** Radio buttons for  Gradle - Groovy,  Gradle - Kotlin, and  Maven.
- Language:** Radio buttons for  Java,  Kotlin, and  Groovy.
- Spring Boot:** Radio buttons for versions 3.2.0 (SNAPSHOT), 3.2.0 (M2), 3.1.4 (SNAPSHOT), 3.1.3, 3.0.11 (SNAPSHOT), 3.0.10, 2.7.16 (SNAPSHOT), and  2.7.15.
- Project Metadata:** Text input fields for:
  - Group: com.parking
  - Artifact: parking
  - Name: parking
  - Description: Demo project for Spring Boot
  - Package name: com.parking.parking
- Packaging:** Radio buttons for  Jar and  War.
- Java:** Radio buttons for versions 20, 17, 11, and  8.

*Nota.* Configuración inicial. Elaboración propia, realizado con herramienta de recortes.

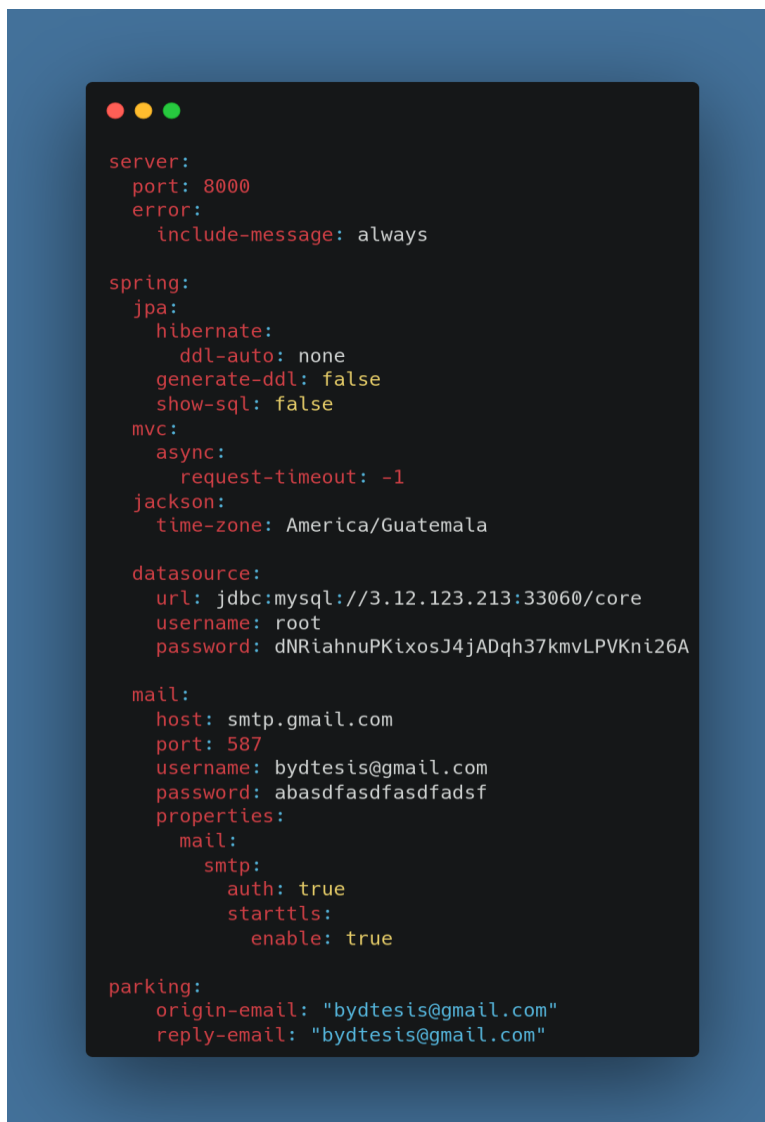
El proyecto se configura con un proyecto Maven, utilizando un lenguaje Java, versión de Spring boot 2.7.15, empaquetado por Jar, y Java 8. Una vez configurado el proyecto, se puede descargar, y nos proporciona los archivos necesarios para empezar el desarrollo de nuestro microservicio.

## 11.2. Descripción de application.yml

El proyecto necesita de un archivo en el cual se puede configurar las variables de entorno, esto nos permitirá indicar a detalle todos los parámetros que se necesitan. Se detalla cada uno de los valores y funcionamiento.

### Figura 59.

*Archivo de entorno*

A screenshot of a terminal window with a dark background and light blue text. The terminal shows the configuration for an application.yml file. The configuration is organized into several sections: 'server', 'spring', 'datasource', 'mail', and 'parking'. Each section contains key-value pairs for various settings. The 'server' section includes port and error handling. The 'spring' section includes JPA and MVC settings. The 'datasource' section includes database connection details. The 'mail' section includes SMTP configuration. The 'parking' section includes email-related settings.

```
server:
  port: 8000
  error:
    include-message: always

spring:
  jpa:
    hibernate:
      ddl-auto: none
      generate-ddl: false
      show-sql: false
  mvc:
    async:
      request-timeout: -1
  jackson:
    time-zone: America/Guatemala

datasource:
  url: jdbc:mysql://3.12.123.213:33060/core
  username: root
  password: dNRiahnuPKixosJ4jADqh37kmvLPVKni26A

mail:
  host: smtp.gmail.com
  port: 587
  username: bydtesis@gmail.com
  password: abasdfasdfasdfadsf
  properties:
    mail:
      smtp:
        auth: true
        starttls:
          enable: true

parking:
  origin-email: "bydtesis@gmail.com"
  reply-email: "bydtesis@gmail.com"
```

*Nota.* Archivo de configuración. Elaboración propia, con herramienta carbon.now.sh

**Tabla 2.***Descripción archivo yml*

<b>Comando</b>	<b>Descripción</b>
port: 8000	Indica el puerto que se utilizara para escuchar las peticiones que se realizan al microservicio
include-message: always	Configuración que habilita la muestra de mensajes de error en el log siempre
ddl-auto: none	Especifica cómo se realiza la configuración del esquema de base de datos.
generate-ddl: false	Indica que Spring boot no generará ningún esquema en base de datos, todo se tomará ya creado de la base de datos.
request-timeout: -1	Parámetro que se utilizará para determinar el tiempo que el microservicio tomara como valido, si se excede este tiempo, retornará una respuesta timeout, el tiempo se debe escribir en milisegundos, y si se quiere un tiempo de forma indefinida se debe colocar como -1
time-zone: America/Guatemala	Indica la zona horaria, esto se utiliza en el microservicio para poder conocer la fecha actual de forma correcta y almacenamiento de datos relacionados a fechas en base de datos de forma correcta.
url: jdbc:mysql://3.12.123.213:33060/core	Parámetro que contiene el string de conexión a base de datos, se debe de colocar la dirección del servidor, puerto y esquema al que se desea acceder.

Continuación de la tabla 2

<b>Comando</b>	<b>Descripción</b>
username: root	Usuario que se utiliza en base de datos
password: asdfasdfasdf	contraseña que se utiliza para autenticar a un usuario en base de datos.
host: smtp.gmail.com	Indica el servidor de correos que se utilizara
port: 587	Indica el puerto por el cual trabaja el servidor de correos
username: bydtesis@gmail.com password: abasdfasdfadsf	Parámetros utilizados para indicar los el usuario por el cual se enviaran los correos electrónicos
origin-email: "bydtesis@gmail.com"	Variable de entorno por medio de la cual se pasa al microservicio el valor del email origen de los correos electrónicos
reply-email: "bydtesis@gmail.com"	Variable de entorno por medio de la que se pasa al microservicio el valor de

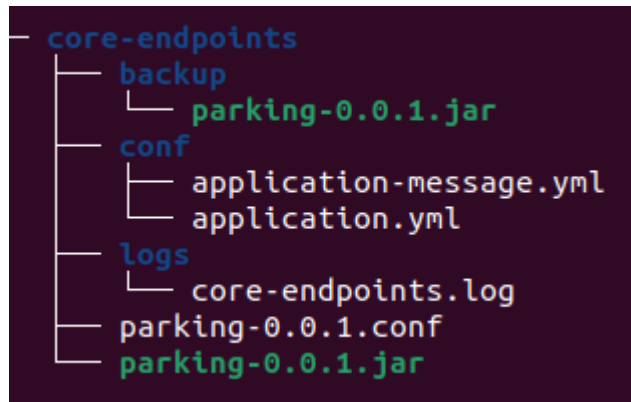
*Nota.* Descripción de archivo de configuración. Elaboración propia, realizado con Microsoft Word.

### **11.3. Estructura de carpetas en servidor de producción**

Al momento de configurar el microservicio en el servidor es necesario realizar algunas configuraciones para su correcto funcionamiento. Los archivos mas importantes hacen referencia a archivos de variables de entorno, logs, y jar.

**Figura 60.**

*Directorio de carpetas de servidor*



*Nota.* Directorio de carpetas. Elaboración propia, realizado con herramienta de recortes.

Descripción de la estructura de carpetas en el servidor.

**Tabla 3.**

*Descripción estructura de carpetas*

<b>Nombre</b>	<b>Descripción</b>
application-message.yml	Los mensajes que retorna el microservicio como respuesta fallida o exitosa, se pueden parametrizar por medio de este archivo, de esta forma si es necesario cambiar algún texto, no requerirá crear una nueva versión de código, ni compilar nuevamente.
application.yml	Archivo donde se parametrizan las variables de entorno que utilizara la aplicación.
core-endpoints.log	Archivo que contiene todas las impresiones que se realizan desde el microservicio, este nos ayuda a poder determinar si existió un error en el microservicio.

Continuación de la tabla 3

<b>Nombre</b>	<b>Descripción</b>
parking-0.0.1.conf	Archivo donde se especifica la ruta en la que se encuentra el archivo de logs, y el archivo de variables de entorno.
parking-0.0.1.jar	Archivo el cual contiene el código fuente de la aplicación.

*Nota.* Descripción de estructura de carpetas. Elaboración propia, realizado con Microsoft Word.

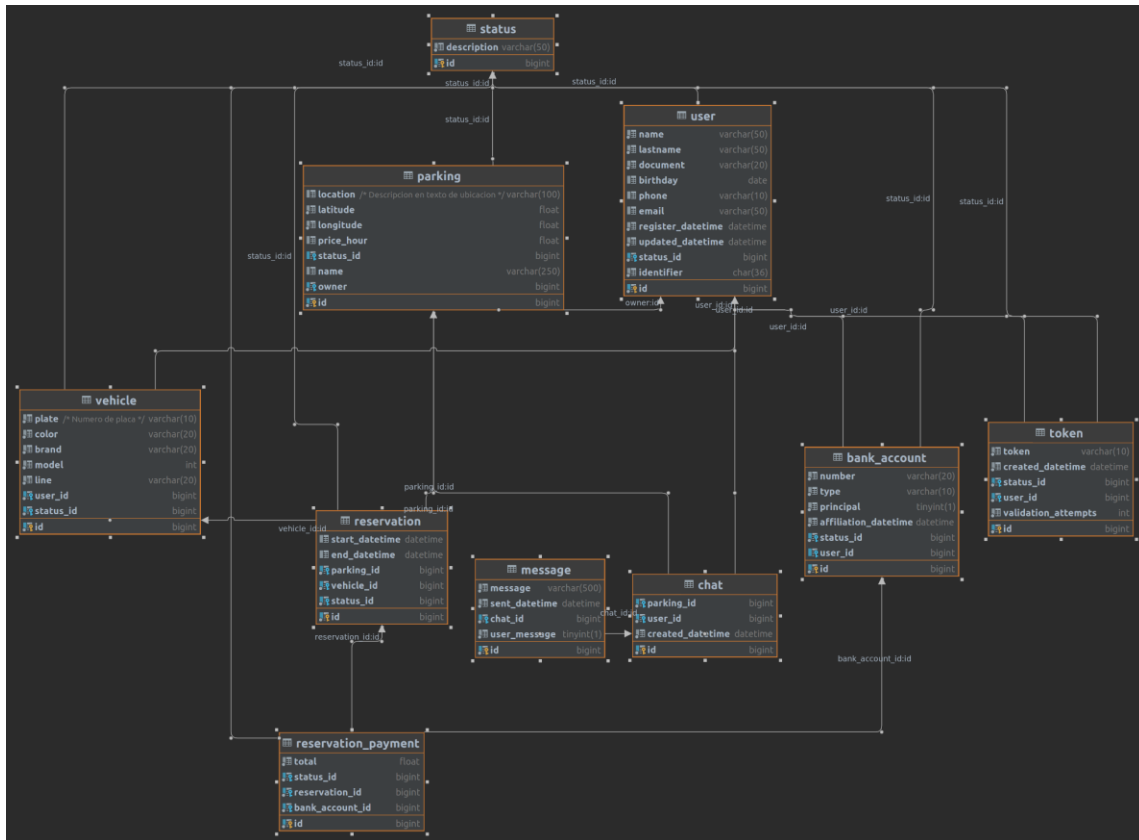
#### **11.4. Base de datos**

El sistema gestor de base de datos que se utilizó es mysql. Al momento de la creación de una nueva base de datos, no posee ningún esquema, ni tabla, por lo que es necesario realizar la creación de las mismas, el sistema utiliza diez tablas, que son para el manejo de cuentas bancarias, chats, mensajes, parqueos, reservaciones, pagos, códigos de confirmación vehículos y usuarios.

En la siguiente gráfica se hace la presentación del modelo entidad relación de nuestra base de datos, la cual representa cada tabla existente, datos, tipo de datos que utiliza y relación que existe entre las distintas tablas que existen.

**Figura 61.**

*Diagrama entidad relación de base de datos.*



Nota. Diagrama de base de datos. Elaboración propia, realizado con datagrip.

### 11.4.1. Script sql con esquema de la base de datos

Para poder crear y configurar la base de datos, es necesario ejecutar comandos para la creación de tablas y relacionar.



**Figura 62.**

*Script de base de datos parte 1*

```
create table status
(
  id          bigint auto_increment primary key,
  description varchar(50) not null
);

create table user
(
  id          bigint auto_increment primary key,
  name        varchar(50)                not null,
  lastname    varchar(50)                not null,
  document    varchar(20)                not null,
  birthday    date                       null,
  phone       varchar(10)                null,
  email       varchar(50)                null,
  register_datetime datetime default CURRENT_TIMESTAMP not null,
  updated_datetime datetime default CURRENT_TIMESTAMP not null
on update CURRENT_TIMESTAMP,
  status_id   bigint                     not null,
  identifier  char(36)                   not null,
  constraint user_status_fk foreign key (status_id) references status (id)
);

create table bank_account
(
  id          bigint auto_increment primary key,
  number      varchar(20)                not null,
  type        varchar(10)                not null,
  principal   tinyint(1) default 0       not null,
  affiliation_datetime datetime default CURRENT_TIMESTAMP not null,
  status_id   bigint                     not null,
  user_id     bigint                     null,
  constraint bank_account_status_fk foreign key (status_id) references status (id),
  constraint bank_account_user_fk foreign key (user_id) references user (id)
);

create table parking
(
  id          bigint auto_increment primary key,
  location    varchar(100)                null comment 'Descripcion en texto de ubicacion',
  latitude    float                       not null,
  longitude   float                       not null,
  price_hour  float                       null,
  status_id   bigint                       null,
  name        varchar(250) default ''    null,
  owner       bigint                       not null,
  constraint parking_status_fk foreign key (status_id) references status (id),
  constraint parking_user_id_fk foreign key (owner) references user (id)
);
```

*Nota.* Código de creación de base de datos. Elaboración propia, realizado con carbon.now.sh.

**Figura 63.**

*Script de base de datos parte 2*

```
create table chat
(
  id          bigint auto_increment          primary key,
  parking_id  bigint                        not null,
  user_id     bigint                        not null,
  created_datetime datetime default CURRENT_TIMESTAMP not null,
  constraint chat_parking_fk foreign key (parking_id) references parking (id),
  constraint chat_user_fk   foreign key (user_id) references user (id)
);

create table message
(
  id          bigint auto_increment          primary key,
  message     varchar(500)                  not null,
  sent_datetime datetime default CURRENT_TIMESTAMP not null,
  chat_id     bigint                        not null,
  user_message tinyint(1) default 1         not null,
  constraint message_chat_fk foreign key (chat_id) references chat (id)
);

create table token
(
  id          bigint auto_increment          primary key,
  token       varchar(10)                   not null,
  created_datetime datetime default CURRENT_TIMESTAMP not null,
  status_id   bigint                        not null,
  user_id     bigint                        not null,
  validation_attempts int default 0         not null,
  constraint token_status_fk foreign key (status_id) references status (id),
  constraint token_user_fk   foreign key (user_id) references user (id)
);
```

*Nota.* Código de creación de base de datos. Elaboración propia, realizado con carbon.now.sh.

**Figura 64.**

*Script de base de datos parte 3*

```
create table vehicle
(
  id          bigint auto_increment
            primary key,
  plate       varchar(10) not null comment 'Numero de placa',
  color       varchar(20) not null,
  brand       varchar(20) not null,
  model       int         not null,
  line        varchar(20) not null,
  user_id     bigint      not null,
  status_id   bigint      not null,
  constraint status_fk      foreign key (status_id) references status (id),
  constraint user_fk       foreign key (user_id) references user (id)
);

create table reservation
(
  id          bigint auto_increment primary key,
  start_datetime datetime null,
  end_datetime datetime null,
  parking_id  bigint not null,
  vehicle_id  bigint not null,
  status_id   bigint not null,
  constraint reservation_parking_fk foreign key (parking_id) references parking (id),
  constraint reservation_status_fk  foreign key (status_id) references status (id),
  constraint reservation_vehicule_fk foreign key (vehicle_id) references vehicle (id)
);

create table reservation_payment
(
  id          bigint auto_increment primary key,
  total       float not null,
  status_id   bigint not null,
  reservation_id bigint not null,
  bank_account_id bigint not null,
  constraint reservation_payment_bank_account_fk
            foreign key (bank_account_id) references bank_account (id),
  constraint reservation_payment_reservation_fk
            foreign key (reservation_id) references reservation (id),
  constraint reservation_payment_status_fk
            foreign key (status_id) references status (id)
);
```

*Nota.* Código de creación de base de datos. Elaboración propia, realizado con carbon.now.sh.

#### 11.4.2. Descripción de tablas

Se dará un detalle de cada una de las tablas utilizadas en base de datos y descripción de sus atributos.

**Tabla 4.**

*Descripción de tablas de base de datos*

<b>Tabla</b>	<b>Descripción</b>
bank_accoun t	Entidad en la cual se almacena la información de las cuentas bancarias que se utilizaran para realizar los pagos dentro del sistema.
chat	Entidad en la cual se almacena información de chats existentes entre distintos usuarios.
Message	Entidad en la cual se almacenarán los mensajes que los usuarios envíen.
Parking	Entidad en la cual se almacenará toda la información relacionada a los parqueos.
Reservatio n	Entidad en la cual se almacenará la información relacionada a las reservaciones, hora de entrada y hora de salida.
Reservatio n_payment	Entidad en la que se almacenará toda la información relacionada con los pagos de reservaciones.
Status	Entidad en la que se almacenarán los diferentes estados que pueden poseer todas las demás entidades que existan.
Token	Entidad en la cual se almacenará toda la información relacionada con los códigos de confirmación que se envíen al usuario para poder registrarse.
User	Entidad en la que se almacenarán todos los usuarios e información relacionada con estos.
Vehicle	Entidad en la que se almacenará toda la información relacionada con los vehículos.

*Nota.* Descripción de tablas en base de datos. Elaboración propia, realizado con Microsoft Word.

### 11.4.2.1. Entidad bank\_account

Entidad en la cual se realizará el almacenamiento de todas las cuentas bancarias que se registren dentro de la aplicación.

**Tabla 5.**

*Detalle de campos de tabla bank\_account*

Nombre del atributo	Tipo	LLave Primaria	Llave Foránea	Descripción
id	bigint	Si	No	Identificador único de la entidad cuenta bancaria.
nombre	varchar(20)	No	No	Nombre asociado a la cuenta bancaria.
type	varchar(10)	No	No	Tipo de cuenta bancaria, puede ser ahorro o monetaria.
principal	boolean	No	No	Si el usuario posee múltiples cuentas, solo una cuenta podrá ser marcada como principal.
affiliation_date	datetime	No	No	Fecha de afiliación la cuenta bancaria en la aplicación por primera vez.
status_id	bigint	No	Si	Valor numérico que hace referencia a la entidad Estado la cuenta bancaria.
user_id	bigint	No	Si	Valor numérico que hace referencia al usuario al que pertenece la cuenta

Continuación de la tabla 5

Nombre del atributo	Tipo	LLave Primaria	Llave Foránea	Descripción
				bancaria.

*Nota.* Descripción de tabla de cuentas. Elaboración propia, realizado con Microsoft Word.

#### 10.4.2.2. Entidad chat

Entidad en la cual se realizará el almacenamiento de información de todos los chats que puedan existir en el sistema.

**Tabla 6.**

*Detalle de campos de tabla chat*

Nombre del atributo	Tipo	Llave primaria	Llave foránea	Descripción
Id	bigint	Si	No	Identificador único de la entidad Chat.
Parking_id	bigint	No	Si	Valor de tipo numérico el cual hace referencia a un parqueo con el cual se está realizando el chat.
User_id	bigint	No	Si	Valor de tipo numérico el cual hace referencia a un usuario con el cual se está realizando el chat.
Created_datetime	datetime	No	No	Fecha en la que se inició el chat entre el usuario y el parqueo.

*Nota.* Descripción de tabla de chats. Elaboración propia, realizado con Microsoft Word.

### 10.4.2.3. Entidad message

Entidad en la cual se realizará el almacenamiento de todos los mensajes que se enviarán por medio de chats en la aplicación.

**Tabla 7.**

*Detalle de campos de tabla message*

Nombre del atributo	Tipo	Llave primaria	Llave foránea	Descripción
Id	bigint	Si	No	Identificador único de la entidad message.
Message	varchar(500)	No	No	Valor en el que se almacena el mensaje enviado por el usuario.
Sent_datetime	datetime	No	No	Fecha en la que fue enviado el mensaje.
User_message	boolean	No	No	Bandera que indica si el mensaje fue enviado por el usuario, o por el parqueo.
Chat_id	bigint	No	Si	Valor de tipo numérico el cual hace referencia a que chat pertenece el mensaje que se está almacenando.

*Nota.* Descripción de tabla de mensajes. Elaboración propia, realizado con Microsoft Word.

#### 10.4.2.4. Entidad parking

Entidad en la cual se realizará el almacenamiento de información relacionada a los parqueos que existan dentro de la aplicación.

**Tabla 8.**

*Detalle de campos de tabla parking*

Nombre del atributo	Tipo	Llave primaria	Llave foránea	Descripción
Id	bigint	Si	No	Identificador único de la entidad parking
location	varchar(100)	No	No	Valor que almacena la ubicación del parqueo descrita con palabras.
latitudo	float	No	No	La latitud en la que se ubica el parqueo.
longitud	float	No	No	Longitud en la que se ubica el parqueo.
price_hora	float	No	No	Precio que cobra el parqueo por hora.
status_id	bigint	No	Si	Valor de tipo numérico que hace referencia a la entidad status.
name	varchar(250)	No	No	Nombre del parqueo
owner	bigint	No	Si	Entidad de tipo numérico que referencia al usuario al que pertenece el parqueo.

*Nota.* Descripción de tabla de parqueos. Elaboración propia, realizado con Microsoft Word.



#### 10.4.2.5. Entidad reservation

Entidad en la cual se realizará el almacenamiento de información relacionada a las reservaciones que existan dentro de la aplicación.

**Tabla 9.**

*Detalle de campos de tabla reservation*

Nombre del atributo	Tipo	Llave primaria	Llave foránea	Descripción
Id	bigint	Si	No	Identificador único de la tabla reservation
start_datetime	datetime	No	No	Fecha y hora de inicio de la reservación.
end_datetime	datetime	No	No	Fecha y hora de fin de la reservación.
parking_id	bigint	No	Si	Valor de tipo numérico que hace referencia al parqueo en el cual se desea hacer la reservación.
vehicle_id	bigint	No	Si	Valor de tipo numérico que hace referencia al vehículo para el que se está realizando la reservación.
status_id	bigint	No	Si	Valor de tipo numérico que hace referencia al estado en el que se encuentra la reservación.

*Nota.* Descripción de tabla de reservaciones. Elaboración propia, realizado con Microsoft Word.

#### 10.4.2.6. Entidad user

Entidad en la cual se realizará el almacenamiento de información relacionada a los usuarios que existan dentro de la aplicación.

**Tabla 10.**

*Detalle de campos de tabla user*

Nombre del atributo	Tipo	Llave primaria	Llave foránea	Descripción
Id	bigint	Si	No	Identificador unico de la entidad user.
name	varchar(50)	No	No	Nombre del usuario.
lastname	varchar(50)	No	No	Apellido del usuario.
document	varchar(20)	No	No	Número de CUI del usuario.
birthday	date	No	No	Fecha de nacimiento del usuario.
phone	varchar(10)	No	No	Número de teléfono del usuario.
email	varchar(50)	No	No	Email del usuario.
register_datetime	datetime	No	No	Fecha de registro del usuario.
updated_datetime	datetime	No	No	Fecha de última actualización de la data del usuario.
status_id	bigint	No	Si	Estado en el que se encuentra el usuario.
identifier	identifier	No	No	Código único para identificar el usuario.

*Nota.* Descripción de tabla de usuarios. Elaboración propia, realizado con Microsoft Word.

#### 10.4.2.7. Entidad vehicle

Entidad en la cual se realizará el almacenamiento de información relacionada a los vehículos que posean los usuarios dentro de la aplicación.

**Tabla 11.**

*Detalle de campos de tabla vehicle*

Nombre del atributo	Tipo	Llave primaria	Llave foránea	Descripción
id	bigint	Si	No	Identificador único de la entidad vehicle.
plate	varchar(10)	No	No	Placa del vehículo.
color	varchar(20)	No	No	Color del vehículo.
brand	varchar(20)	No	No	Marca del vehículo.
model	int	No	No	Año del vehículo.
line	varchar(20)	No	No	Tipo del vehículo
user_id	bigint	No	Si	Valor tipo numérico que hace referencia al usuario al que pertenece el vehículo.
status_id	bigint	No	Si	Valor de tipo numérico que hace referencia al estado en el que se encuentra el vehículo.

*Nota.* Descripción de tabla de vehiculos. Elaboración propia, realizado con Microsoft Word.

## 11.5. Configuración de ambiente de desarrollo para frontend

Para la preparación y ejecución correcta del proyecto, es necesario realizar la instalación de múltiples herramientas, tanto para desarrollo como para compilación de la aplicación.

### 11.5.1. Modo desarrollador

Para poder ejecutar correctamente la aplicación en un dispositivo Android físico, es necesario habilitar el modo desarrollador en dicho dispositivo, por lo que se hace de la siguiente manera:

**Tabla 12.**

*Pasos para modo desarrollador*

<b>Paso</b>	<b>Descripción</b>
<b>#1</b>	Acceder al menú de configuraciones del dispositivo
<b>#2</b>	Seleccionar "Acerca del teléfono"
<b>#3</b>	Presionar "Información de software"
<b>#4</b>	Presionar varias veces la opción "Número de compilación", hasta que se muestre un mensaje que indique que el modo desarrollador se ha activado correctamente.
<b>#5</b>	Regresar al menú anterior y presionar "Opciones del desarrollador"
<b>#6</b>	Buscar y habilitar la opción de "Depuración por USB"

*Nota.* Pasos para configuración modo desarrollador. Elaboración propia, realizado con Microsoft Word.

## 11.5.2. Código fuente y dependencias

Ahora se debe obtener el proyecto, ubicado en un repositorio de GitHub de la siguiente manera, luego se instalan las dependencias y en ambos casos se hace utilizando una terminal de comandos:

**Tabla 13.**

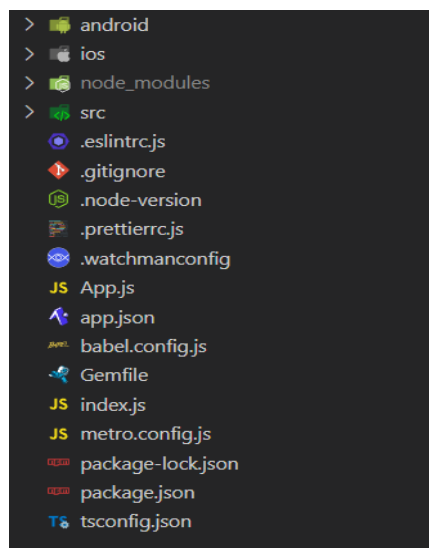
*Comandos para clonar repositorio e instalación de librerías*

Comando	Descripción
<b>git clone https://github.com/breedel/tesis_mobile_app.git</b>	Clona el repositorio del código fuente en el directorio, es importante realizarlo en una carpeta nueva o vacía.
<b>npm install --save --force</b>	Instala cada una de las librerías de terceros especificadas dentro del archivo package.json

*Nota.* códigos para configuración de librerías. Elaboración propia, realizado con Microsoft Word.

**Figura 65.**

*Estructura general de código de aplicación móvil*



*Nota.* Directorio de estructura de código fuente de la aplicación móvil. Elaboración propia, realizado con herramienta de recortes.

**Tabla 14.***Descripción general de carpetas y archivos del proyecto*

<b>Nombre de archivo/carpetas</b>	<b>Descripción</b>
<b>android</b>	Carpeta que contiene todos los archivos necesarios para la compilación en el sistema operativo Android, aquí se incluyen imágenes de presentación (Splash Screen en inglés), AndroidManifest.xml (acá se agregan los permisos a utilizar en aplicación), almacén de claves.
<b>ios</b>	Carpeta con los archivos necesarios para compilación en iOS, acá se incluye el archivo Info.plist (acá se agregan los permisos a utilizar en aplicación), entre otros.
<b>src</b>	Aquí se incluye todo el código del proyecto de aplicación móvil, incluidos componentes, archivo de configuración de ambiente, imágenes, llamadas al API, archivos para el cambio de idioma.
<b>.eslintrc.js</b>	Archivo para configurar y personalizar reglas que son utilizadas por ESLint.
<b>.gitignore</b>	Archivo en el que se definen cada uno de los directorios y archivos que no se le dará seguimiento al realizar algún cambio en código.
<b>.node-version</b>	Archivo utilizado por Node.js, en el que se especifica la versión de Node.js que utilizará el proyecto.
<b>.prettierrc.js</b>	Archivo utilizado para definir reglas para mantener un formato legible en el proyecto, no modificado en el proyecto
<b>.watchmanconfig</b>	Permite monitorear las carpetas o archivos modificados del proyecto, creado por defecto por React Native, no utilizado en el proyecto.
<b>App.js</b>	Archivo que representa la pantalla principal, en él se desarrolla todo el funcionamiento del proyecto.

Continuación de la tabla 14

<b>app.json</b>	Ayuda a centralizar acciones en común entre plataformas como configuración de permisos, Splash Screen, nombre de aplicación, creado por defecto por React Native, no modificado en el proyecto.
<b>babel.config.js</b>	Permite añadir reglas de Babel para utilizar código de JavaScript moderno, creado por defecto por React Native, no modificado en el proyecto.
<b>Gemfile</b>	Acá se definen las dependencias utilizadas por Ruby y Cocoapods para la compilación en iOS, no utilizado en el proyecto.
<b>index.js</b>	Archivo principal del proyecto, es la puerta de entrada a la aplicación móvil
<b>metro.config.js</b>	Archivo de configuración del Metro Bundler, no modificado en el proyecto.
<b>package-lock.json</b>	Archivo que contiene importante información sobre las dependencias del proyecto que se encuentran ubicadas dentro de node_modules.
<b>package.json</b>	Archivo que especifica el nombre y versión de las dependencias instaladas en el proyecto.
<b>tsconfig.json</b>	Archivo de configuración en proyectos de React Native en el que se utiliza TypeScript, no utilizado en el proyecto.

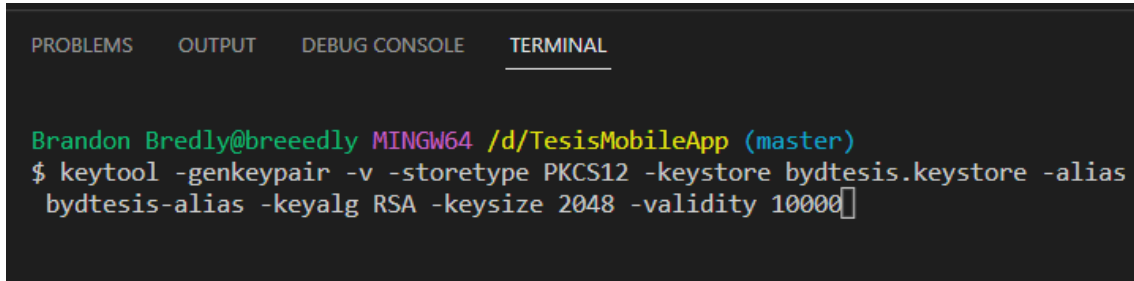
*Nota.* Descripción de carpetas de aplicación. Elaboración propia, realizado con Microsoft Word.

### 11.5.3. Almacén de claves

Se debe generar un archivo de almacén de llaves, que es necesario para la publicación en la tienda de Google Play Store, así como la generación de archivo de ejecución .apk, para utilizar en cualquier dispositivo Android físico.

## Figura 66.

### Generación de almacén de claves



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Brandon Bredly@breedly MINGW64 /d/TesisMobileApp (master)
$ keytool -genkeypair -v -storetype PKCS12 -keystore bydtesis.keystore -alias
  bydtesis-alias -keyalg RSA -keysize 2048 -validity 10000
```

*Nota.* Comando para generar archivo bydtesis.keystore. Elaboración propia, realizado con herramienta recortes.

## Tabla 15.

### Parámetros al ejecutar comando de generación de almacén de claves

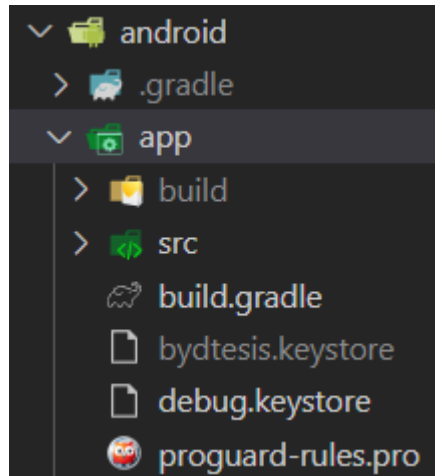
Parámetro	Valor
Contraseña del almacén	bydt3s1s
Confirmar contraseña	bydt3s1s
Nombre y/o apellido	Brandon
Nombre de organización	USAC
Ciudad o localidad	Guatemala
Estado o provincia	Guatemala
Código del país de dos letras	GT
Es correcto	Si

*Nota.* Descripción de parámetros para generación de almacén de claves de aplicación. Elaboración propia, realizado con Microsoft Word.



**Figura 67.**

*Ubicación de archivo de almacén de claves*



*Nota.* Ubicación de archivo bydtesis.keystore. Elaboración propia, realizado con herramienta recortes.

#### 11.5.4. Configuración para AWS

Se debe crear un archivo llamado aws-exports.js el cuál contiene la información de la cuenta de AWS para consumir los servicios de Cognito y debe ser incluido en la carpeta llamada src.

**Figura 68.**

*Archivo de conexión con AWS*

```
const awsmobile = {
  'aws_project_region': 'us-east-2',
  'aws_cognito_identity_pool_id': 'arn:aws:cognito-idp:us-east-2:129340444638:userpool/us-east-2_x36iLZPWv',
  'aws_cognito_region': 'us-east-2',
  'aws_user_pools_id': 'us-east-2_x36iLZPWv',
  'aws_user_pools_web_client_id': 'm710d3n2kbfjj3s227mka8s5s',
  'authenticationFlowType': 'USER_PASSWORD_AUTH',
  'oauth': {}
}

export default awsmobile;
```

*Nota.* Contenido del archivo aws-exports.js. Elaboración propia, realizado con Recortes.

**Tabla 16.**

*Descripción de contenido de archivo aws-exports.js*

<b>Nombre atributo</b>	<b>Descripción</b>
<b>aws_project_region</b>	Atributo utilizado para especificar la región a la que se conectará la aplicación.
<b>aws_cognito_identity_pool_id</b>	Especifica el ID del grupo de identidades de Cognito para gestionar la autorización y autenticación de usuarios.
<b>aws_cognito_region</b>	Define la región en la que se encuentra Cognito, configurada previamente en AWS.
<b>aws_user_pools_id</b>	Especifica el ID del grupo de usuarios para gestionar la autenticación.
<b>aws_user_pools_web_client_id</b>	Es el ID de cliente web que la aplicación móvil utiliza para comunicarse al hacer la autenticación con Cognito.
<b>authenticationFlowType</b>	Se define el tipo de autenticación utilizado para comunicarse con Cognito
<b>oauth</b>	Se utiliza para configurar la autenticación basada en OAuth, el atributo no tiene ningún valor asignado, pero debe ir especificado en el archivo.

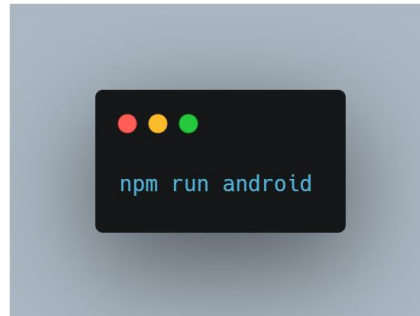
Nota. Descripción de archivo de configuración. Elaboración propia, realizado con Microsoft Word.

### **11.5.5. Ejecutar proyecto**

Se debe conectar por medio de USB el dispositivo Android físico a la computadora, cambiar la configuración a transferencia de archivos y abrir una terminal en la carpeta raíz del proyecto y ejecutar el siguiente comando:

**Figura 69.**

*Comando para ejecutar proyecto*



*Nota.* Instalará aplicación en teléfono en modo de pruebas. Elaboración propia, realizado con herramienta de recortes.



## **12. PRESENTACIÓN DE RESULTADOS**

### **12.1. Beneficio de uso de plataforma**

Al tener en la aplicación móvil una funcionalidad de selección de idioma entre español e inglés, permite que pueda ser utilizada con facilidad por extranjeros como por personas nacionales, y por su configuración a nivel de código, permite que la implementación de nuevos idiomas como francés o italiano se pueda realizar en un corto período de tiempo.

Al asegurar un espacio tras hacer una reservación en un establecimiento para uno o varios vehículos, permitirá a los propietarios de estos establecimientos poder incrementar sus precios al poseer un servicio nuevo disponible a sus clientes y con esto poder tener un historial de ingresos, saber cuántas personas realizaron una reservación en un día en específico.

Para los usuarios que utilicen la aplicación móvil pueden verificar cerca de la ubicación que se encuentran, a la que se viajarán al día siguiente o en los próximos días y saber del precio, lugar en específico de donde guardarán su vehículo de forma segura y poderse comunicar con el dueño del parqueo en tiempo real, ante cualquier duda que pueda surgir.

### **12.2. Oportunidades**

Al ser una aplicación dirigida a utilizar con vehículos y establecimientos públicos, puede ser integrada en cualquier lugar que posea varios espacios disponibles para acomodar de forma eficiente a vehículos como motos. Se puede

integrar en establecimientos nuevos, como en lugares que ya se encuentran en funcionamiento como en centros comerciales y no es necesario la construcción de una estructura física para que puedan utilizar esta aplicación.

Dado que se utiliza AWS, para futuras actualizaciones con nuevas funcionalidades, se podrían utilizar servicios como S3 para almacenar fotografías de los parqueos, y que estas puedan ser vistas desde la aplicación móvil, generando así, mayor confianza para quienes la utilizan, cuando son establecimientos pequeños y los usuarios desean saber cómo es la infraestructura del lugar previo a ir al lugar o reservar.

Si se desea disminuir el tiempo de desarrollo en nuevas funcionalidades para la aplicación móvil, se le debe dar continuidad al proyecto a nivel de código utilizando un computadora Mac, ya que en estas se pueden instalar emuladores de iPhone, lo que permite ver el comportamiento de la aplicación en un dispositivo iOS, aunque no se cuente con uno, debido a que actualmente la aplicación móvil no posee una funcionalidad que no pueda ejecutarse en un emulador de iPhone, como el caso de las notificaciones emergentes.

### **12.3. Consideraciones**

Si se desea tener un mayor alcance con usuarios que tengan dispositivos con sistema operativo iOS y publicar la aplicación en la App Store, es necesario tener una computadora Mac y configurarla para ejecutar el proyecto correctamente, realizar configuraciones a nivel de código, pagar anualmente \$99.00 USD para habilitar el modo desarrollador y mantener las actualizaciones en la App Store.

Inicialmente se propone el uso de cuentas bancarias para realizar los pagos de las reservaciones, pero existen librerías de terceros que pueden ser integradas en la aplicación móvil, así como del lado del servidor, si se desean agregar pagos con tarjetas de crédito. Esto permitiría poder tener un mayor alcance en cuánto a publicar la aplicación en distintos países.

Para tener un mayor control en las reservaciones, es necesario identificar cada uno de los espacios en el parqueo, para que en la aplicación móvil se pueda escoger un espacio en específico, garantizando que la persona que haga la reservación pueda contar con el lugar disponible al llegar al lugar. Si bien garantiza la disponibilidad del espacio, hay que considerar factores externos ajenos a los parqueos los cuales pueden ser el tráfico o el clima, por lo que habría que evitar hacer reservaciones para el mismo día o con un par de horas o minutos de diferencia con la hora actual.





## CONCLUSIONES

1. Se facilitó la gestión de los espacios, tiempo y costos utilizados para parqueos de vehículos como de motos, mediante una aplicación móvil que permite realizar reservaciones a personas con un margen de tiempo prudencial, pudiendo incrementar los costos a los dueños de los parqueos para garantizar la disponibilidad de espacios a quienes utilicen los parqueos disponibles.
2. Se geolocalizó en un mapa en tiempo real en la aplicación móvil, mediante la utilización de un API de mapas de Google, y se muestra cada una de las ubicaciones disponibles de los parqueos cercanos a la ubicación obtenida del dispositivo y, al seleccionar una en específico, se muestra la información detallada del parqueo.
3. Se automatizó el tiempo de salidas y entradas en cada uno de los parqueos que se registren en la aplicación móvil, al poder determinar el tiempo que se tendrá de utilización y la cantidad de vehículos que llegan al parqueo.



## RECOMENDACIONES

1. Asignar un código de identificación en los espacios de los parqueos, para que en posibles nuevas funcionalidades de la aplicación móvil, se pueda seleccionar el espacio que se desea reservar, mejorando la experiencia del usuario.
2. Integrar un API para que se puedan agregar métodos de pagos con tarjetas de crédito lo que permitirá obtener una comisión extra, tanto para quienes administran el parqueo como para los desarrolladores de la aplicación manteniendo siempre la aplicación gratuita.
3. Adquirir un equipo Mac y configurarlo para publicar la aplicación en la tienda App Store para tener un mayor alcance de usuarios.



## REFERENCIAS

- Angular. (s/f). Angular.lat. Recuperado el 7 de octubre de 2023, de <https://docs.angular.lat/tutorial/toh-pt5>
- Angular material. (s/f). Angular Material. Recuperado el 7 de octubre de 2023, de <https://material.angular.io/guide/getting-started>
- Clear cache & cookies. (s/f). Google.com. Recuperado el 7 de octubre de 2023, de <https://console.cloud.google.com/welcome?project=amazing-ripple-311705>
- Extensions, L. M. A. (s/f). Visual Studio Code - code editing. Redefined. Visualstudio.com. Recuperado el 7 de octubre de 2023, de <https://code.visualstudio.com/>
- Font awesome. (s/f). Recuperado el 7 de octubre de 2023, de <https://fontawesome.com>
- Getting started. (s/f). DataGrip Help. Recuperado el 7 de octubre de 2023, de <https://www.jetbrains.com/help/datagrip/getting-started.html>
- Google Play Console. (s/f). Google play console. Google.com. Recuperado el 7 de octubre de 2023, de <https://play.google.com/console/u/0/developers>
- IntelliJ IDEA: el IDE líder para Java y Kotlin. (s/f). JetBrains. Recuperado el 7 de octubre de 2023, de <https://www.jetbrains.com/es-es/idea/>
- Introducción a Android Studio. (s/f). Android Developers. Recuperado el 7 de octubre de 2023, de <https://developer.android.com/studio/intro?hl=es-419>
- Melgar, R. (2017, mayo 25). Visitantes y vecinos de Antigua se quejan por la falta de espacios para estacionarse. <https://www.prensalibre.com/ciudades/sacatepequez/visitantes-y-vecinos-de-antigua-se-quejan-por-la-falta-de-espacios-para-estacionarse/>
- Pocasangre, H. (2016, diciembre 20). Comuna emite nuevos costos para parqueos municipales.

<https://www.prensalibre.com/guatemala/comunitario/comuna-emite-nuevo-costo-para-parqueos-municipales/>

Referencia. (s/f). Git-scm.com. Recuperado el 7 de octubre de 2023, de <https://git-scm.com/docs>

Rodas, S. M. (2023, marzo 1). Verificamos por usted: ¿Es ilegal reservar espacios en la vía pública para cobrar por el estacionamiento? Prensa Libre. <https://www.prensalibre.com/guatemala/comunitario/verificamos-por-usted-es-ilegal-reservar-espacios-en-la-via-publica-para-cobrar-por-el-estacionamiento/>

Setting up the development environment. (s/f). Reactnative.dev. Recuperado el 7 de octubre de 2023, de <https://reactnative.dev/docs/environment-setup>

Transito, M. T. (2021, julio 20). Sigue incrementándose el parque vehicular en Guatemala. Departamento de Tránsito de la Policía Nacional Civil. <https://transito.gob.gt/sigue-incrementandose-el-parque-vehicular-en-guatemala/>

(S/f-a). Amazon.com. Recuperado el 7 de octubre de 2023, de <https://aws.amazon.com/es>

(S/f-b). Recuperado el 7 de octubre de 2023, de <https://docs.amplify.aws/start/q/integration/react-native/>

(S/f-c). Muniguate.com. Recuperado el 7 de octubre de 2023, de [https://www.muniguate.com/concejo-municipal/wp-content/uploads/sites/14/2022/03/24\\_Tasa\\_Municipal\\_Estacionamiento\\_Vehiculos\\_Estacionamientos\\_Publicos\\_Municipales.pdf](https://www.muniguate.com/concejo-municipal/wp-content/uploads/sites/14/2022/03/24_Tasa_Municipal_Estacionamiento_Vehiculos_Estacionamientos_Publicos_Municipales.pdf)

(S/f-d). Icon.kitchen. Recuperado el 7 de octubre de 2023, de <https://icon.kitchen/i/H4sIAAAAAAAAAA6tWKkvMKU0tVrKqVkpJLMoOyUjNTVWYskvMKU6t1VHKzU8pzQHJRisI5qUU5WemKOkoZeYXA8ny1CSi2FoApT8%2BHkAAAAA%3D>