



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

**PROTOTIPO DE SISTEMA DE INFORMACIÓN PARA LA OPTIMIZACIÓN DE LA GESTIÓN  
DE MARBETES CON CASO DE ESTUDIO EN LA ANTIGUA GUATEMALA**

**Bruno Marco José Coronado Morales**  
Asesorado por el Ing. Josué David Pérez

Guatemala, octubre de 2023



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**PROTOTIPO DE SISTEMA DE INFORMACIÓN PARA LA OPTIMIZACIÓN DE LA GESTIÓN  
DE MARBETES CON CASO DE ESTUDIO EN LA ANTIGUA GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**BRUNO MARCO JOSÉ CORONADO MORALES**  
ASESORADO POR EL ING. JOSUÉ DAVID PÉREZ

AL CONFERÍRSELE 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 Martinez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Ing. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Carlos Gustavo Alonzo
EXAMINADOR	Ing. Manuel Haroldo Castillo Reyna
EXAMINADOR	Ing. Byron Rodolfo Zepeda Arévalo
SECRETARIO	Ing. Hugo Humberto Rivera Pérez



## **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:

### **PROTOTIPO DE SISTEMA DE INFORMACIÓN PARA LA OPTIMIZACIÓN DE LA GESTIÓN DE MARBETES CON CASO DE ESTUDIO EN LA ANTIGUA GUATEMALA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 24 de febrero del 2022.



**Bruno Marco José Coronado Morales**

Guatemala, 05 de septiembre de 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 **BRUNO MARCO JOSÉ CORONADO MORALES** con carné 201709362 y CUI 3058 01864 0301 titulado "PROTOTIPO DE SISTEMA DE INFORMACIÓN PARA LA OPTIMIZACIÓN DE LA GESTIÓN DE MARBETES CON CASO DE ESTUDIO EN LA ANTIGUA GUATEMALA", luego de corroborar que el mismo se encuentra finalizado, lo he revisado y doy fé 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. José David Pérez  
Colegiado No. 17098

*José David Pérez*  
INGENIERO EN SISTEMAS DE INFORMACION  
Y CIENCIAS DE LA COMPUTACION  
No. COLEGIADO 17098



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

Guatemala 19 de septiembre de 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 del estudiante **BRUNO MARCO JOSÉ CORONADO MORALES** con carné **201709362** y CUI **3058 01864 0301** titulado “**PROTOTIPO DE SISTEMA DE INFORMACIÓN PARA LA OPTIMIZACIÓN DE LA GESTIÓN DE MARBETES CON CASO DE ESTUDIO EN LA ANTIGUA GUATEMALA**”, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

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

Atentamente,



**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.8.EICCSS.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: **Prototipo de Sistema de Información para la Optimización de la Gestión de Marbetes con Caso de Estudio en la Antigua Guatemala**, presentado por: **Bruno Marco José Coronado Morales**, 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



**USAC**  
TRICENTENARIA  
Universidad de San Carlos de Guatemala

Decanato  
Facultad e Ingeniería

24189101- 24189102

LNG.DECANATO.OIE.87.2023

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: **PROTOTIPO DE SISTEMA DE INFORMACIÓN PARA LA OPTIMIZACIÓN DE LA GESTIÓN DE MARBETES CON CASO DE ESTUDIO EN LA ANTIGUA GUATEMALA**, presentado por: **Bruno Marco José Coronado Morales** 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: 10/10/2023 17:43:54  
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: 87 CUI: 3058018640301

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 permitirme cumplir una meta más en la vida.

**Mis padres**

Beatriz Morales y José Luis Coronado. Por su amor y apoyo incondicional en todas las etapas de mi vida.



## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES .....	VII
LISTA DE SÍMBOLOS .....	XI
GLOSARIO .....	XIII
RESUMEN .....	XV
OBJETIVOS .....	XVII
INTRODUCCIÓN .....	XIX
1. IDENTIFICACIÓN DE LA PROBLEMÁTICA PRESENTADA.....	1
1.1. Análisis histórico.....	2
1.2. Antecedentes .....	5
1.3. Ubicaciones geográficas con una problemática similar .....	6
2. TECNOLOGÍAS PARA EL DESARROLLO DE APLICACIONES WEB, MÓVILES Y SIMULACIÓN .....	7
2.1. Tecnologías para el desarrollo de software.....	7
2.1.1. Tecnologías para el desarrollo de plataformas web .....	8
2.1.2. Tecnologías para el desarrollo de aplicaciones móviles .....	10
2.1.3. Tecnologías para la simulación de sistemas..	15
2.1.4. Tecnologías para la implementación de base de datos.....	16
2.2. Comunicación entre plataformas web y móviles .....	18
2.2.1. Transferencia de estado representacional .....	19
2.2.2. Protocolo simple de acceso a objetos .....	20

2.3.	Estándares de seguridad en el desarrollo de plataformas web y móviles.....	21
2.4.	Herramientas para la implementación de pruebas unitarias .....	24
3.	DISEÑO Y DOCUMENTACIÓN DE LA SOLUCIÓN.....	25
3.1.	Diseño de la solución.....	25
3.1.1.	Entradas, procesos y salidas del sistema .....	25
3.1.2.	Flujo de información en los procesos.....	29
3.1.3.	Arquitectura propuesta para la solución.....	30
3.1.4.	Modelo lógico de base de datos .....	33
3.2.	Definición de recursos en servicios backend principales ...	36
3.2.1.	Servicio de autenticación .....	36
3.2.1.1.	Inicio de sesión .....	36
3.2.1.2.	Registro de visitante .....	37
3.2.1.3.	Obtención de usuarios vendedores.....	38
3.2.2.	Servicio de venta .....	40
3.2.2.1.	Registro de vehículo del visitante .....	40
3.2.2.2.	Obtención de vehículos del visitante .....	41
3.2.2.3.	Registrar marbete sobre un vehículo .....	42
3.2.2.4.	Obtención de marbetes sobre un vehículo específico....	43
3.2.3.	Servicio de validación .....	44
3.2.3.1.	Registrar verificación de marbete .....	44

	3.2.3.2.	Obtención de verificaciones realizadas por un usuario .....	45
	3.2.4.	Servicio de reportería .....	46
	3.2.4.1.	Reporte de rendimiento de vendedores .....	46
	3.2.4.2.	Reporte de ventas de vendedores .....	47
	3.2.4.3.	Reporte de verificaciones de vendedores .....	48
3.3.		Interfaz gráfica de la plataforma web .....	49
	3.3.1.	Inicio de sesión.....	49
	3.3.2.	Registro.....	50
	3.3.3.	Gestión de vehículos del visitante .....	50
	3.3.4.	Historial de transacciones del usuario.....	51
	3.3.5.	Control de vendedores .....	52
	3.3.6.	Verificación en tiempo real .....	52
3.4.		Interfaz gráfica de la aplicación móvil.....	53
	3.4.1.	Inicio de sesión.....	53
	3.4.2.	Listado de marbetes vendidos .....	54
	3.4.3.	Venta de marbete.....	55
	3.4.4.	Detalle de marbete .....	56
	3.4.5.	Listado de verificaciones .....	57
	3.4.6.	Verificación de marbete.....	58
	3.4.7.	Detalle de verificación de marbete .....	59
	3.4.8.	Histórico de venta de marbetes.....	60
	3.4.9.	Histórico de verificación de marbetes.....	61
4.		SIMULACIÓN DE LOS PROCESOS INVOLUCRADOS.....	63
	4.1.	Características del sistema .....	63

4.2.	Vistas del modelo de simulación.....	65
4.2.1.	Vista aérea.....	65
4.2.2.	Entrada del sistema .....	66
4.2.3.	Vendedores.....	66
4.2.1.	Salida del sistema.....	67
4.3.	Configuración de los componentes del modelo de simulación.....	68
4.4.	Resultados del modelo de simulación del proceso actual	69
4.5.	Resultados del modelo de simulación con la solución propuesta.....	69
4.6.	Análisis comparativo entre los resultados de la simulación.....	70
4.7.	Recomendación de mejora sobre los modelos basado en la teoría de colas.....	71
4.7.1.	Modelo del proceso actual .....	72
4.7.2.	Modelo de la solución propuesta .....	74
4.7.3.	Recomendación de vendedores para el modelo actual.....	75
4.7.4.	Recomendación de vendedores para el modelo de la solución propuesta .....	76
4.8.	Resultados del modelo de simulación del proceso actual según recomendación de vendedores .....	76
4.9.	Resultados del modelo de simulación de la solución propuesta con la recomendación de vendedores .....	78
4.10.	Análisis comparativo entre los resultados iniciales y según recomendación de vendedores .....	79

4.10.1.	Modelo del proceso actual contra modelo del proceso actual con recomendación de vendedores .....	79
4.10.2.	Modelo de la solución propuesta contra modelo de la solución propuesta con recomendación de vendedores .....	80
4.10.3.	Modelo de proceso actual contra modelo de la solución propuesta con recomendación de vendedores .....	81
CONCLUSIONES .....		83
RECOMENDACIONES .....		85
REFERENCIAS .....		87



# ÍNDICE DE ILUSTRACIONES

## FIGURAS

<b>Figura 1.</b>	Parque vehicular a nivel nacional hasta mayo 2021 .....	1
<b>Figura 2.</b>	Tráfico en el ingreso a la antigua guatemala .....	5
<b>Figura 3.</b>	Características principales de flutter .....	12
<b>Figura 4.</b>	Compilación en dart .....	14
<b>Figura 5.</b>	Industrias a las que es aplicable simio .....	15
<b>Figura 6.</b>	Flujo abstracto para la implementación de oauth 2.0 .....	23
<b>Figura 7.</b>	Proceso de venta en sitio .....	26
<b>Figura 8.</b>	Proceso de venta en línea .....	27
<b>Figura 9.</b>	Proceso de validación de marbete digital .....	28
<b>Figura 10.</b>	Flujo de información entre componentes .....	30
<b>Figura 11.</b>	Implementación de arquitectura soa .....	31
<b>Figura 12.</b>	Modelo lógico de base de datos.....	33
<b>Figura 13.</b>	Modelo relacional de base de datos.....	35
<b>Figura 14.</b>	Definición del recurso de inicio de sesión .....	37
<b>Figura 15.</b>	Definición del recurso de registro de visitante.....	38
<b>Figura 16.</b>	Definición del recurso de obtención de vendedores.....	39
<b>Figura 17.</b>	Definición del recurso de registro de vehículo del visitante .....	40
<b>Figura 18.</b>	Definición del recurso de obtención de vehículos .....	41
<b>Figura 19.</b>	Definición del recurso de registro de marbete.....	42
<b>Figura 20.</b>	Definición de recurso de obtención de marbetes .....	43
<b>Figura 21.</b>	Definición del recurso de registro de verificación .....	44
<b>Figura 22.</b>	Definición del recurso para la obtención de verificaciones.....	45
<b>Figura 23.</b>	Definición del recurso de rendimiento de vendedores.....	46

<b>Figura 24.</b>	Definición del recurso del reporte de ventas .....	47
<b>Figura 25.</b>	Definición del recurso del reporte de verificaciones .....	48
<b>Figura 26.</b>	Página de inicio de sesión .....	49
<b>Figura 27.</b>	Página de registro de usuario .....	50
<b>Figura 28.</b>	Página de gestión de vehículos del visitante .....	51
<b>Figura 29.</b>	Página del historial de transacciones del usuario .....	51
<b>Figura 30.</b>	Página de control de vendedores .....	52
<b>Figura 31.</b>	Página de verificación en tiempo real .....	53
<b>Figura 32.</b>	Pantalla de inicio de sesión.....	54
<b>Figura 33.</b>	Listado de venta de marbetes .....	55
<b>Figura 34.</b>	Pantalla de venta de marbete .....	56
<b>Figura 35.</b>	Pantalla de detalle de marbete .....	57
<b>Figura 36.</b>	Pantalla de listado de verificaciones .....	58
<b>Figura 37.</b>	Pantalla de verificación de marbete .....	59
<b>Figura 38.</b>	Pantalla de detalle de verificación de marbete.....	60
<b>Figura 39.</b>	Pantalla con el histórico de venta de marbetes.....	61
<b>Figura 40.</b>	Pantalla con el histórico de verificación de marbetes .....	62
<b>Figura 41.</b>	Vista aérea del modelo .....	65
<b>Figura 42.</b>	Vista de la entrada al sistema .....	66
<b>Figura 43.</b>	Vendedores atendiendo visitantes .....	67
<b>Figura 44.</b>	Vista de la salida del modelo .....	68

## TABLAS

<b>Tabla 1.</b>	Definición de características principales del sistema .....	63
<b>Tabla 2.</b>	Resultados del modelo de simulación del proceso actual.....	69
<b>Tabla 3.</b>	Resultados del modelo de simulación de la solución propuesta .....	70
<b>Tabla 4.</b>	Comparación resultados proceso actual y solución propuesta .....	71
<b>Tabla 5.</b>	Resultados del proceso actual según recomendación .....	77
<b>Tabla 6.</b>	Resultados de la solución propuesta según recomendación .....	78
<b>Tabla 7.</b>	Comparación resultados proceso actual y según recomendación ...	79
<b>Tabla 8.</b>	Resultados solución propuesta y según recomendación .....	80
<b>Tabla 9.</b>	Resultados proceso actual y propuesta según recomendación .....	81



## LISTA DE SÍMBOLOS

Símbolo	Significado
$\rho$	Factor de utilización en la teoría de colas.
<b>M/M/K</b>	Indica un modelo bajo una tasa de llegadas independiente, tasa de servicio independiente y una cantidad de servidores positiva mayor a uno.
$\lambda$	Tasa de llegadas en la teoría de colas.
$\mu$	Tasa de servicio en la teoría de colas.



## GLOSARIO

<b>API</b>	Por sus siglas en inglés al término Application programming interface, es el componente de software que intercomunica otros componentes de software.
<b><i>Backend</i></b>	Componente de software que contiene la lógica del negocio, que se ejecuta del lado del servidor y es transparente al usuario final.
<b>CLI</b>	Por sus siglas en inglés al término Command line interface, es la interfaz que permite al usuario ejecutar comandos por medio de una línea de texto.
<b>CSS</b>	Por sus siglas en inglés al término Cascading style sheets, es un lenguaje que permite manejar el diseño y presentación de una página web.
<b><i>Frontend</i></b>	Componente de software que contiene la interfaz gráfica que utiliza el usuario final.
<b>HTML</b>	Por sus siglas en inglés al término Hypertext markup language, es el lenguaje basado en etiquetas que estructura una página web.

<b>HTTP</b>	Por sus siglas en ingles al termino Hypertext transfer protocol, es el protocolo que permite la transferencia de hipertexto a través de la red.
<b>IDE</b>	Se refiere por sus siglas en inglés al término Integrated development environmet. Es un programa informático por medio del cual se pueden crear aplicaciones de software.
<b>JavaScript</b>	Lenguaje de programación que puede ser utilizado tanto a nivel de servidor como de cliente.
<b>OpenSource</b>	Modelo de desarrollo en el cual cualquier usuario tiene acceso al código fuente.
<b>WebApp</b>	Software que se ejecuta a través de un navegador web adaptado a cualquier dispositivo.
<b>Framework</b>	Es un conjunto de herramientas para realizar un software bajo un concepto o estructura definida.
<b>TypeScript</b>	Lenguaje de programación de tipado fuerte el cual fue construido tomando como base JavaScript.

## **RESUMEN**

Con el paso del tiempo más y más personas adquieren un vehículo propio, lo que incrementa considerablemente el tiempo en tráfico de muchas personas en Guatemala y uno de los lugares en los que se ve reflejado este fenómeno es en la Antigua Guatemala en sus días festivos y fines de semana.

Actualmente la venta de marbetes contribuye a incrementar el tiempo que los visitantes deben pasar en el tráfico, por este motivo surge la necesidad de automatizar el proceso de venta a través de las tecnologías de la información con el fin de reducir el tiempo de espera para el ingreso a la ciudad.

Los visitantes serán capaces de adquirir un marbete digital con antelación y evitar el proceso de venta al ingreso de la ciudad y de igual manera los vendedores, a través de dispositivos móviles, por medio del escaneo de la placa de un vehículo pueden realizar la venta o verificación de un marbete, información que es enviada directamente al ente rector para el monitoreo y seguimiento de la venta de marbetes.



# OBJETIVOS

## General

Desarrollar una plataforma para la gestión de marbetes mediante una aplicación web y móvil para la optimización de los procesos involucrados.

## Específicos

1. Proporcionar a los visitantes una plataforma web que les permita adquirir marbetes en línea.
2. Proveer al personal a cargo de la venta y verificación de marbetes un medio para poder agilizar los procesos mediante una aplicación móvil.
3. Comparar los resultados de la simulación del proceso actual contra el proceso digitalizado con la plataforma desarrollada.



## INTRODUCCIÓN

Con el paso del tiempo los lugares turísticos se vuelven más conocidos y visitados tanto por turistas nacionales como extranjeros. Comúnmente los visitantes llegan por medio de vehículos, tanto privados como públicos, y que a lo largo del tiempo han ido incrementando y que, debido a la falta de infraestructura necesaria para soportar la alta carga vehicular, se tiende a caer en largos tiempos de espera para visitar los lugares turísticos.

La implementación de estrategias que permitan gestionar la alta carga vehicular en los distintos lugares turísticos juega un papel importante en la experiencia de los visitantes, por lo tanto, es importante pensar en soluciones que permitan brindar una mejor experiencia a los visitantes.

Uno de los lugares más visitados en Guatemala es la Antigua Guatemala, ubicado en el departamento de Sacatepéquez, que debido a sus características ha implementado el cobro del marbete para el estacionamiento de vehículos en espacios públicos. Esto, adicional al incremento de visitantes, ocasiona un incremento en los tiempos de espera en tráfico por parte de los visitantes.

El modelo de venta de marbetes que actualmente es implementado en la Antigua Guatemala es un modelo que fácilmente puede ser replicado en distintos destinos turísticos con características similares de Guatemala y que, de igual forma, incrementen los tiempos de espera en tráfico a los visitantes.





Según la gráfica anterior, un 43.06 % de los vehículos registrados se encuentran en el departamento de Guatemala, seguido de Quetzaltenango con un 6.88 % Y entre los tipos de vehículos que más transitan hay un 43.09 % que son motocicletas, seguido de un 19.40 % de automóviles particulares,

### **1.1. Análisis histórico**

Una de las principales actividades económicas dentro del territorio guatemalteco es el turismo. Existen muchos lugares que son visitados en fines de semana y días festivos por turistas nacionales o extranjeros, los cuales, dependiendo del sitio al que se dirijan deben de realizar un viaje, que muchas veces es en un vehículo propio y que al llegar al lugar los visitantes se enfrentan a la tarea de buscar un lugar para estacionar dicho vehículo, lo que dependiendo del lugar visitado puede variar en cuanto a la disponibilidad y costo.

La Antigua Guatemala es una ciudad colonial que recibe frecuentemente la visita de turistas nacionales y extranjeros no solo en fines de semana o días festivos, sino también en cualquier época del año. Es común encontrar turistas visitando las distintas ruinas, restaurantes típicos e iglesias que reflejan la historia colonial que caracteriza la ciudad.

Según el Reglamento para la regulación del establecimiento de vehículos en la vía pública dentro del perímetro urbano colonial, y áreas circundantes del Municipio de La Antigua Guatemala publicado en el diario oficial el 12 de noviembre del 2008 con una adición el 16 de marzo del 2020, publicado también en el diario oficial, define según los artículos 8 y 15.

ARTÍCULO 8.- Tasa: Para tener derecho al estacionamiento en el municipio de La Antigua Guatemala. Los usuarios deberán cancelar en la Oficina de Control, con los agentes la PMT o en los puestos de cobro, la tasa autorizada que se clasifica así:

- Vehículos particulares hasta 7 pasajeros. Diez quetzales (Q. 10.00) por día calendario.
- Vehículos de 7 a 15 pasajeros: Veinte quetzales (Q20.00) por día calendario.
- Autobuses de 15 a 30 pasajeros: cuarenta quetzales (Q40.00) por día calendario
- Autobuses de más de 30 pasajeros, Cincuenta quetzales Q50.00 por día calendario.

ARTÍCULO 15.- Excepciones: Los vecinos del Municipio de La Antigua Guatemala, quedan exentos del pago de estacionamiento, previo registro en la Oficina de Control, en la cual se les entregará una calcomanía de registro gratuita, después de haber llenado los requisitos siguientes:

- Fotocopia de cédula de vecindad de este municipio.
- Fotocopia de la tarjeta de circulación del o los vehículos a nombre del vecino.
- Fotocopia del boleto de Ornato de este municipio.
- Dirección y teléfono.

Todos aquellos vecinos de otros municipios, que por cualquier razón, utilicen las Áreas de Estacionamiento, en el municipio de La Antigua Guatemala de manera frecuente, podrán optar por adquirir la calcomanía de Estacionamiento, en la Oficina de Control, la cual tendrá un costo de Doscientos Quetzales Anuales (Q. 200.00), por vehículo, de hasta siete (7) pasajeros por año calendario, debiendo llenar los requisitos de registro que determine la Oficina de Control.

Todos los vehículos de dos ruedas, están exentos del pago por estacionamiento.

Este pago, descrito en el artículo 8, es realizado al ingreso de la ciudad de forma manual, lo que ocasiona, mayormente en fines de semana y días festivos, una larga cola de vehículos y grandes tiempos de espera para los visitantes que desean ingresar a la ciudad, lo que con el paso del tiempo se ha vuelto en un problema, no solo para los visitantes, sino para toda aquella persona que se dirija a la ciudad.

## Figura 2.

### *Tráfico en el ingreso a la Antigua Guatemala*



*Nota.* Vehículos ingresando a la Antigua Guatemala. Obtenido de Publinews (2017). *Parqueo en Antigua Guatemala* (<https://static.publinews.gt/2017/01/17/parques-en-antigua-guatemala-para-viernes-santo-3.jpg>), Consultado el 7 de marzo de 2022. De dominio público.

## 1.2. Antecedentes

En muchos lugares en donde se tiene que pagar por el parqueo, ya sea al ingreso o egreso ya se ha implementado el uso de la tecnología para poder agilizar los procesos y facilitar la interacción con los usuarios, entre los beneficios que estos ofrecen son el pago en línea o pago por medio de tarjetas de crédito/débito, ubicación del vehículo parqueado y ubicación de espacios disponibles.

### **1.3. Ubicaciones geográficas con una problemática similar**

Guatemala cuenta con muchos lugares turísticos, que al igual que la Antigua Guatemala, es visitada por turistas locales y extranjeros, por lo que existen diversos destinos los cuales puedan enfrentar la problemática de espacios de parqueo y adoptar una temática similar a la venta de marbetes. Algunos de estos destinos turísticos pueden ser:

- Quetzaltenango, Quetzaltenango
- Flores, Petén
- Chichicastenango, Quiché
- Ciudad de Guatemala, Guatemala
- Río Dulce, Izabal

## **2. TECNOLOGIAS PARA EL DESARROLLO DE APLICACIONES WEB, MOVILES Y SIMULACIÓN**

En la actualidad se está viviendo una revolución tecnológica que está involucrando a la sociedad y en general la vida diaria de las personas, esto marca una época de progreso, desarrollo e innovación en diferentes aspectos de la sociedad y que ha derivado en el surgimiento de nuevas ramas tecnológicas como la ingeniería informática, ciberseguridad, Big Data, Robótica, entre otras.

### **2.1. Tecnologías para el desarrollo de software**

El desarrollo de software pertenece a la rama de ingeniería informática, siendo esta una de las que inició con la revolución tecnológica y que ha sufrido cambios a lo largo del tiempo. En la actualidad en muchas organizaciones se busca poder agilizar los procesos internos y externos por medio de sistemas de información, que permitan generar mejores resultados aprovechando de manera óptima los recursos con los que cuentan y para lograr esto han surgido una amplia gama de tecnologías en diferentes áreas del desarrollo de software que permitan a las organizaciones cumplir con este objetivo.

### **2.1.1. Tecnologías para el desarrollo de plataformas web**

Las tecnologías y modelos para el desarrollo de aplicaciones web se encuentran en constante evolución, existen muchas alternativas para frameworks, aplicaciones, librerías, arquitecturas y sistemas de publicación que constantemente son actualizados y mejorados con el tiempo. Esta evolución ha dado paso a tener un amplio abanico de posibilidades a la hora de tomar una elección por lo que es importante tomar en cuenta los aspectos más importantes de cada tecnología y orientar la elección en base a las necesidades de la plataforma.

La arquitectura de las plataformas web está compuesta, generalmente por los siguientes elementos, los cuales pueden variar dependiendo de las necesidades:

- Navegador, permite al usuario descargar los recursos estáticos que conforman la plataforma web al acceder por medio de una URL.
- Servidor, es el encargado de brindar el contenido estático y datos solicitados por el cliente según la dirección URL.
- Protocolo HTTP, protocolo basado en TCP/IP mediante el cual el navegador realiza peticiones al servidor.
- HTML, lenguaje utilizado para la estructuración del contenido de una plataforma web.
- CSS, lenguaje utilizado para definir la presentación del contenido de una plataforma web.

Los componentes mencionados anteriormente conforman los elementos básicos de una plataforma web. En la actualidad es común encontrar una subdivisión en las tecnologías que conforman el servidor separándolas en tecnologías que se ejecutan del lado del cliente y tecnologías que se ejecutan del lado del servidor, algunas de estas tecnologías pueden ser:

- Cliente, estas tecnologías permiten crear la interfaz gráfica con la que el usuario interactúa y controlar toda la comunicación con el servidor, estas están basadas en las tecnologías HTML, CSS y Javascript. Existen frameworks de trabajo que permiten definir la estructura, lógica de los componentes y agilizar el desarrollo de la interfaz gráfica, una de estas plataformas es Angular. Angular es un framework de trabajo para desarrollar SPA en typescript y que es mantenido por mantenido por Google, entre sus principales características entramos:
  - Velocidad y Rendimiento, A partir de plantillas basadas en HTML, CSS y Javascript, angular genera código optimizado para las máquinas virtuales de Javascript, puede ser ejecutado sobre cualquier tipo de servidor de contenido estático además de permitir la carga rápida de los componentes y dividir automáticamente el contenido según las solicitudes.
  - Productividad, Ofrece una Interfaz de Línea de Comandos (CLI) desde la cual se pueden generar nuevos componentes, ejecutar pruebas y visualizar el contenido de la aplicación, es compatible con una amplia gama de editores de código e IDEs y las plantillas están basadas en una sintaxis simple, pero a la vez potente.

- Ambiente Completo de Desarrollo, Utiliza Karma para poder realizar pruebas unitarias sobre el código fuente, Protractor para realizar pruebas punto-a-punto y un API propia e intuitiva para la implementación de animaciones complejas con poco código.
- Servidor, estas tecnologías se encargan de poder procesar las solicitudes del cliente a través de distintos tipos de petición. Se encargan de realizar toda la lógica del negocio, procesar los datos de entrada y/o salida e interconectarse con servicios externos como pueden ser una base de datos. Entre las tecnologías que podemos encontrar en esta subdivisión se encuentra Node.js. Node.js es un entorno de ejecución multiplataforma desarrollado en C++ y Javascript y basado en el motor V8 de Google, está enfocado en la creación de programas de red altamente escalables como los servidores web.

### **2.1.2. Tecnologías para el desarrollo de aplicaciones móviles**

Con los avances de la transformación digital, el desarrollo de aplicaciones móviles se ha visto en una creciente evolución y expansión en la tecnología. Muchas plataformas optan por desarrollar estas aplicaciones debido a las distintas ventajas que estas brindan y de la mano las tecnologías que son utilizadas también se han visto envueltas en esta transformación.

Los tipos de aplicaciones móviles que se desarrollan actualmente se pueden catalogar en tres tipos:

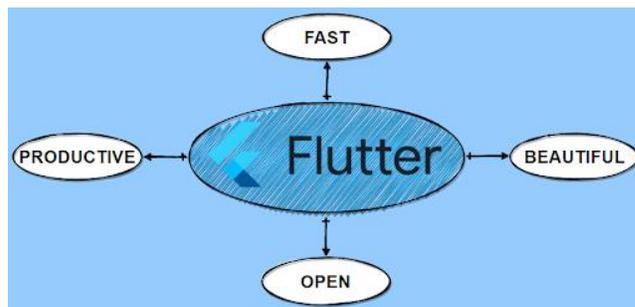
- Aplicaciones Nativas, son desarrolladas bajo un lenguaje de programación específico según el sistema operativo, por lo que ofrecen más seguridad y menos fallos por la compatibilidad entre el lenguaje y el entorno donde se ejecuta. Además, al estar desarrolladas con un lenguaje específico del sistema operativo permite tener un mayor control sobre las características y componentes internos del dispositivo.
- WebApps, este tipo de aplicaciones están desarrolladas bajo las mismas tecnologías que utilizan las plataformas web, esto hace que sean adaptables a cualquier sistema operativo o dispositivo móvil. En esencia este tipo de aplicaciones son páginas web adaptadas para tener una apariencia de aplicación nativa.
- Aplicaciones Híbridas, este tipo de aplicaciones comparte características tanto de las nativas como las WebApps, sigue el desarrollo de una aplicación web y su capacidad para poder ejecutarse sobre cualquier dispositivo o sistema operativo, pero tiene más acceso a las características y componentes del dispositivo como las aplicaciones nativas.

Existen frameworks que permiten desarrollar aplicaciones móviles de forma rápida, ágil y eficaz. Estos nos proveen de esquemas conceptuales que engloban toda una estructura lógica para una aplicación y no iniciar desde cero con el desarrollo. Cuentan con herramientas que automatizan las tareas más comunes para evitar escribir código repetitivo y enfocarse en desarrollar características más avanzadas. Uno de estos frameworks es Flutter.

Flutter es un framework para desarrollo de aplicaciones móviles nativas multiplataforma, desarrollado por Google, que mediante el desarrollo de un único proyecto genera una aplicación con rendimiento nativo para distintos sistemas operativos utilizando el lenguaje de programación Dart, también desarrollado por Google.

### Figura 3.

*Características principales de Flutter*



*Nota.* Características principales de Flutter. Obtenido de D. Bernal (2021). *Características de Flutter* (<https://profile.es/wp-content/media/Caracteristicas-de-Flutter.png>), Consultado el 8 de marzo de 2022. De dominio público.

En la imagen se puede observar las cuatro características principales que caracterizan a flutter frente a otras opciones, estas se describen a continuación:

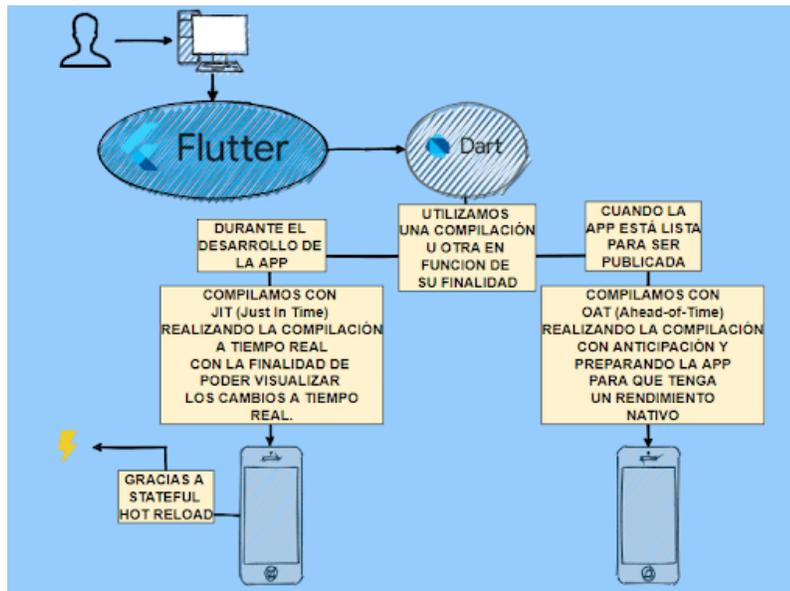
- Rapidez, Flutter ofrece un set de herramientas que permiten tener mayor rapidez tanto en el desarrollo como rendimiento de las aplicaciones.
- OpenSource, Tanto Flutter (Framework) como Dart (Lenguaje) son de código abierto y mantenidos por Google, por lo que la documentación y comunidad asociada a esta tecnología es grande y con una popularidad en crecimiento.

- Aplicaciones Vistas, Mediante las herramientas que ofrece flutter se pueden crear interfaces que van desde lo más sencillo, hasta interfaces muy vistosas y expresivas.
- Productividad, Incrementa la productividad al permitir generar aplicaciones multiplataforma a partir de código escrito en Flutter.

Flutter ofrece un conjunto de herramientas que permiten agilizar el desarrollo y mejorar el rendimiento de la aplicación, estas herramientas están constituidas por:

- Recarga en Caliente con Estado: esta herramienta agiliza el desarrollo de aplicaciones ya que permite realizar cambios en el código y que estos se vean reflejados instantáneamente en la vista previa de la aplicación.
- Compilación JIT vs. OAT con Dart: inicialmente cuando desarrollamos una aplicación Flutter realiza una compilación Just In Time (JIT) lo que implica que los cambios en código son compilados en tiempo de ejecución y se puedan visualizar en tiempo real. Mientras que una compilación AOT es realizada antes de la ejecución de la aplicación, por lo que no se pueden realizar o ver cambios del código una vez ésta ya ha sido ejecutada.
- Skia, motor de gráficos 2D: esta es una biblioteca de renderizado de gráficos 2D escrita en C++ que únicamente necesita un Canvas, elemento HTML para generar gráficos dinámicos en base a código. De esta manera flutter se encarga de renderizar todos los componentes gráficos de la aplicación y no la plataforma, esto permite que las aplicaciones consuman menos recursos y tengan una mejor velocidad.

**Figura 4.**  
*Compilación en Dart*



*Nota.* Compilación en Dart. Obtenido de D. Bernal (2021). *Compilación en Dart* (<https://profile.es/wp-content/media/Compilacion-de-Dart-en-Flutter.png>), Consultado el 8 de marzo de 2022. De dominio público.

En la imagen anterior se describen las distintas formas de compilación (JIT o AOT), si es un contexto de desarrollo se compila con JIT, mientras que cuando la aplicación está lista para ser publicada se realiza una compilación AOT para obtener un rendimiento nativo.

### 2.1.3. Tecnologías para la simulación de sistemas

Las tecnologías para simular sistemas y procesos son una herramienta idónea para ensayar, conocer el funcionamiento y anticiparse a posibles problemas que puedan surgir en una implementación completa. Al implementar estas tecnologías podemos comprobar cómo es que en realidad funciona un sistema o proceso, analizar las variables de entrada, tiempos de ejecución y variables externas que estén involucradas y una herramienta ideal para poder realizar todo este proceso es SIMIO.

SIMIO es un software para Windows de modelización, simulación y animación 3D de flujos de procesos por eventos discretos, basado en un enfoque mixto que combina objetos y procedimientos para producir modelos de comportamiento de sistemas logísticos, industriales y de servicio.

#### Figura 5.

*Industrias a las que es aplicable SIMIO*



*Nota.* Industrias en las que se puede implementar SIMIO. Obtenido de SIMIO (2022). *Industrias* (<https://www.simio.com/>), Consultado el 15 de marzo de 2022. De dominio público.

En la imagen anterior podemos observar las industrias a las que es posible aplicar los modelos de simulación creados con SIMIO. Las aplicaciones son diversas y pueden ser aplicadas para simular cualquier sistema donde existan flujos de materiales, personas o información. El uso de SIMIO nos permite realizar las siguientes tareas a la hora de simular un proceso:

- Modelar: SIMIO ofrece una interfaz capaz de capturar y describir el funcionamiento de un proceso y cómo se comporta al ser ejecutado.
- Simular: a partir del modelo generado, podemos estudiar los resultados y ver las respuestas del proceso a partir de situaciones hipotéticas o futuras.
- Visualizar: SIMIO tiene la capacidad de presentar los resultados de los modelos mediante animaciones 3D gráficamente atractivas e informes numéricos de todo tipo.

#### **2.1.4. Tecnologías para la implementación de base de datos**

La información que obtienen las empresas por el uso de sus servicios es un insumo fundamental para la toma de decisiones y otros procesos internos, por lo tanto, la persistencia de estos datos debe ser realizada por profesionales y sistemas capaces de procesar toda esta información y asegurar que se almacene de forma segura. Para lograr esto y con el surgimiento de nuevas necesidades a partir de esta información, se han desarrollado distintos tipos de bases de datos orientadas a necesidades más generales o específicas, según sea el caso.

Entre los tipos de bases de datos existentes, se pueden mencionar tres, que son comúnmente encontradas dentro de los procesos de las organizaciones y cumplen un propósito específico para el cual se implementa, estas son: relacionales, NoSQL y OLTP.

Las bases de datos relacionales se hicieron predominantes en la década de 1980. Los elementos que la conforman se organizan en un conjunto de tablas con filas y columnas. Este tipo de base de datos proporciona una forma más eficiente y flexible de acceder a la información estructurada y para las empresas en la actualidad poseer una base de datos de este tipo es una pieza clave de su funcionamiento debido a que mantienen la integridad sobre los datos en entornos de acceso múltiple, lo que garantiza altos niveles de efectividad.

Entre las distintas implementaciones existentes para las bases de datos relacionales existe PostgreSQL, la cual es una implementación de código abierto que utiliza un sistema de consultas relacionales basado en SQL. SQL es el lenguaje estándar utilizado para la comunicación con una base de datos relacional según el instituto nacional estadounidense de estándares ANSI.

PostgreSQL es uno de los sistemas de base de datos más avanzados y esto es gracias a que cuenta con una serie de características que se describen a continuación:

- Código Abierto, una de las principales razones de la popularidad de PostgreSQL es la comunidad que ha generado la cual está integrada por una gran cantidad de desarrolladores que aportan y respaldan el proyecto.

- Multiplataforma, la implementación de PostgreSQL puede ser realizada bajo distintos entornos y sistemas operativos además de ser compatible con distintos servidores web como Apache, Nginx o LiteSpeed.
- Soporte para Grandes Volúmenes de Datos, debido a las altas transferencias y transacciones que se dan hoy día en las plataformas web, es necesario que la base de datos tenga la capacidad de procesar todas las solicitudes. Mediante el método de control MVCC implementado en PostgreSQL es posible realizar operaciones de lectura y escritura de forma simultánea.
- ACID, PostgreSQL cuenta con un soporte total de ACID, son las siglas en inglés de atomicidad, consistencia, aislamiento y durabilidad, lo que asegura que si se produce una falla en una transacción los datos no se perderán ni se guardarán a medias.
- Soporte NoSQL, adicionalmente a las características intrínsecas de PostgreSQL como una base de datos relacional, cuenta también con una característica híbrida que le permite realizar consultas de forma no relacional al utilizar el formato JSON en lugar de SQL.

Al juntar las características de PostgreSQL encontramos una base de datos completa y avanzada que cumple con las necesidades que puedan surgir durante el desarrollo de cualquier tipo de aplicaciones web.

## **2.2. Comunicación entre plataformas web y móviles**

A medida que nuevos sistemas y servicios van surgiendo a través de internet, la comunicación debe seguir un estándar para poder tener una interacción entre todos. En la actualidad existen dos enfoques distintos para la transmisión de datos en línea: transferencia de estado representacional (REST) y el protocolo simple de acceso a objetos (SOAP).

### **2.2.1. Transferencia de estado representacional**

Comúnmente conocido como REST, es un conjunto de principios arquitectónicos que se ajusta a las necesidades de las aplicaciones móviles y los servicios web. Al tratarse de un conjunto de pautas la implementación de estas queda a criterio del equipo de desarrollo.

Cuando un cliente solicita datos a un servidor web configurado como REST, comúnmente conocido como API o servicios web RESTful, este servicio devuelve los datos en distintos formatos: HTML, XML, texto sin formato o JSON, siendo el más utilizado el formato JSON, el cual es una notación de objetos JavaScript que puede ser interpretado por cualquier lenguaje de programación y comprendido tanto por personas o máquinas.

Las pautas arquitectónicas para considerar un servicio web como RESTful son las siguientes:

- Se debe seguir una arquitectura cliente-servidor compuesta por clientes, servidores y recursos.
- La comunicación entre el cliente y el servidor debe ser sin estado. Esto significa que el contenido de los clientes no se almacena en el servidor entre las solicitudes, sino que la información sobre el estado de la sesión queda en el cliente.
- Los datos de acceso frecuente pueden almacenarse en caché para evitar la interacción entre algunas operaciones del cliente-servidor.
- Una interfaz uniforme entre elementos para que la información se transfiera de forma estandarizada, en lugar de ser específica para las necesidades de cierta aplicación.

- Una restricción del sistema en capas, en la que las interacciones cliente-servidor pueden estar mediadas por capas jerárquicas.
- Código según se solicite, lo que permite que los servidores amplíen las funciones de un cliente al transferir el código ejecutable.

### **2.2.2. Protocolo simple de acceso a objetos**

SOAP es un protocolo estándar que se creó para posibilitar la comunicación entre las aplicaciones que se diseñan con diferentes lenguajes y en distintas plataformas. Al ser un protocolo impone reglas integradas que aumentan la complejidad y la sobrecarga, lo que puede retrasar el tiempo que tardan las páginas en cargarse. Sin embargo, estos estándares también ofrecen normas integradas que pueden ser ideales para el sector empresarial.

Las especificaciones comunes de los servicios web implementados siguiendo el protocolo SOAP incluyen las siguientes:

- Seguridad de los servicios web (WS-Security): estandariza la forma de proteger y transferir los mensajes usando identificadores únicos llamados tokens.
- Mensajería segura de los servicios web (WS-ReliableMessaging): estandariza el control de errores entre mensajes que se transfieren con fuentes poco confiables.
- Abordaje de los servicios web (WS-Addressing): paquetes que enrutan la información como metadatos dentro de los encabezados SOAP, en lugar de mantener la información en un lugar más profundo de la red.
- Lenguaje de descripción de los servicios web (WSDL): describe que hace un servicio web, su inicio y fin.

El soporte de envío de datos por medio de este protocolo es soportado dentro de distintos protocolos de la capa de aplicación: HTTP para los navegadores web, SMTP para el correo electrónico, TCP, entre otros. Las solicitudes a servicios bajo este protocolo no son almacenadas en caché por lo que no es posible acceder a ellas después de enviadas.

### **2.3. Estándares de seguridad en el desarrollo de plataformas web y móviles**

La seguridad en la transferencia de información para las plataformas web y móviles es un aspecto fundamental y que dependiendo del tipo de aplicación puede ser flexible o estricta. Para poder asegurar que la información se encuentre protegida se desarrolló OAuth 2.0 como un estándar abierto para la autorización de API 's que permite compartir información entre sitios sin tener que compartir la identidad y que es utilizado por grandes compañías como Google, Facebook, Microsoft, entre otras.

Mediante la implementación de OAuth 2.0 el usuario delega la capacidad de realizar ciertas acciones sobre las cuales da su consentimiento y permite hacerlas en su nombre. De esta forma permite desarrollar aplicaciones que no tengan que almacenar el usuario y contraseña de este. Este estándar está compuesto por tres componentes: roles, tipos de clientes y tipos de concesión.

Dentro del estándar podemos encontrar la definición de los siguientes roles:

- Dueño del recurso (Owner): es el usuario que da autorización para realizar determinadas acciones en su nombre.

- Cliente: es la aplicación que realiza las acciones en el nombre del usuario.
- Servidor de Autorización (Authorization Server): es la plataforma encargada de gestionar las peticiones de autorización y verifica la identidad de los usuarios para luego emitir tokens de acceso.
- Servidor de recursos protegidos (Resource Server): es el servicio que aloja los recursos a los cuales se quiere acceder.

Dentro de la definición de clientes dentro del estándar podemos encontrar a los clientes confidenciales y públicos. Los clientes confidenciales son aquellos que son capaces de guardar una contraseña sin que sea expuesta. Un ejemplo de este tipo es una aplicación nativa compilada. Y los clientes públicos son aquellos que no son capaces de mantener la contraseña a salvo, como por ejemplo aplicaciones web.

Y el último componente dentro del estándar son los tipos de concesión, estos pueden variar dependiendo de la implementación, algunos de los tipos más conocidos son:

- Authorization code, basada en un código de autorización.
- Implicit, implícita.
- Resource owner password credentials, contraseña del propietario de recursos.
- Device code flow, basada en el dispositivo.
- Client credentials flow, basada en las credenciales del cliente.

**Figura 6.**

*Flujo abstracto para la implementación de OAuth 2.0*



*Nota.* Flujo abstracto de OAuth 2.0. Obtenido de L. López (2020). *Flujo Abstracto* (<https://openwebinars.net/blog/que-es-oauth2/>), Consultado el 18 de marzo de 2022. De dominio público.

En la imagen anterior, podemos observar lo que el flujo genérico de autenticación a seguir al implementar este estándar, se observa la solicitud de autorización del cliente hacia el dueño de los recursos para luego interactuar con los servidores de autorización y recursos.

## **2.4. Herramientas para la implementación de pruebas unitarias**

Para asegurar la calidad de un producto y como una herramienta útil durante el proceso de desarrollo, la implementación de las pruebas unitarias forma parte importante para hacer valer estos aspectos. Dependiendo de las tecnologías que se utilizan para el desarrollo de un software las herramientas de pruebas pueden variar.

En el caso del desarrollo de plataformas web con JavaScript, específicamente en el framework Angular, la herramienta para la realización de estas pruebas es Karma. Karma es un módulo que permite automatizar algunas de las tareas de las suites de testing y fue desarrollada por el mismo equipo que Angular.

Las pruebas implementadas con karma permiten probar elementos individuales de forma automatizada, lo que permite realizar el proceso de pruebas de forma acumulativa a la hora de trabajar. Mientras desarrollamos nuevas funcionalidades podemos generar pruebas unitarias que validen el resultado que esperamos obtener, para luego desarrollar la función que haga que la prueba unitaria sea satisfactoria según lo dicta el desarrollo dirigido por pruebas o TDD.

### **3. DISEÑO Y DOCUMENTACIÓN DE LA SOLUCIÓN**

Para la implementación de un software que busque automatizar cualquier tipo de proceso es necesario iniciar desde el diseño y la documentación de las necesidades y procesos involucrados. El diseño y documentación servirá de guía para poder establecer el alcance de la implementación y asegurar que el producto final cumpla con el objetivo inicial.

#### **3.1. Diseño de la solución**

El diseño de la solución plasma todos los aspectos importantes en la planificación, desarrollo e implementación de un software. Busca plasmar el flujo de los procesos que serán digitalizados, a partir de estos modelar la lógica del negocio y diseñar una infraestructura que esté por detrás con todas las implementaciones tecnológicas necesarias para suplir las necesidades.

##### **3.1.1. Entradas, procesos y salidas del sistema**

La implementación de herramientas digitales para el control de marbetes busca automatizar los procesos principales que se llevan a cabo, estos son:

Venta de Marbetes, este proceso abarca las dos modalidades propuestas para la adquisición de marbetes por parte de las personas, una en el sitio de ingreso a la ciudad y otra totalmente en línea.

## Figura 7.

### Proceso de venta en sitio



*Nota.* Diagrama del proceso de venta de marbetes en sitio. Elaboración propia, realizado con Canva.

En la imagen anterior se observa la interacción del usuario con la aplicación móvil para asociar un marbete a un vehículo. Inicialmente el vendedor escanea la placa del vehículo por medio de la aplicación móvil para asociar el marbete de forma digital y recibe el pago en forma física por parte del visitante.

## Figura 8.

### Proceso de venta en línea



*Nota.* Diagrama del proceso de venta de marbetes en línea. Elaboración propia, realizado con Canva.

En la imagen anterior se observa la interacción del visitante con la plataforma web para adquirir el marbete. El visitante accede a la plataforma con sus credenciales, obtenidas por medio del registro de nuevos usuarios, y realiza la compra de un marbete para alguno de los distintos vehículos que tiene registrados.

Validación de Vehículos, este proceso abarca la verificación por medio de la aplicación móvil al escanear la placa del vehículo para verificar la adquisición del marbete.

## Figura 9.

### *Proceso de validación de marbete digital*



*Nota.* Diagrama del proceso de validación de marbete digital. Elaboración propia, realizado con Canva.

En la imagen anterior se observa la interacción del usuario con la aplicación móvil para escanear la placa del vehículo y determinar si esta cuenta o no con la adquisición del marbete mediante las notificaciones de la aplicación móvil.

La principal entrada en los procesos mencionados anteriormente es el ingreso de datos por medio del escaneo de la placa por la plataforma móvil y la captura de información del usuario y su interacción con la plataforma web para la adquisición de su marbete digital.

Las salidas resultantes de los procesos se dividen en dos categorías, la primera son las notificaciones informativas según el proceso que informan al usuario acerca del resultado de las verificaciones y el segundo es la recopilación de la información capturada por medio de la aplicación móvil y plataforma web para la consolidación de reportería que abarca dos módulos:

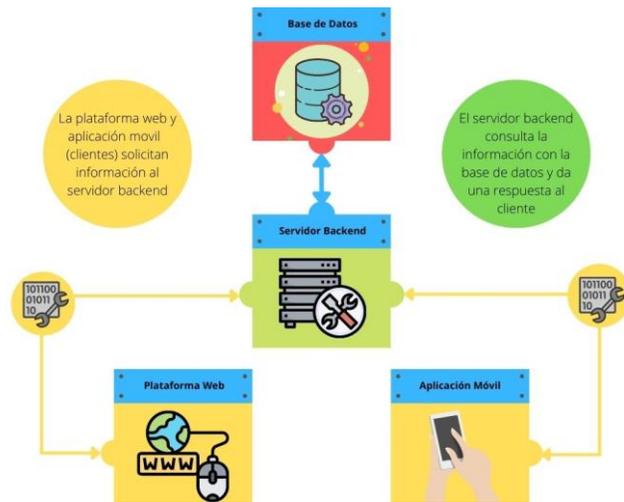
- Control de verificación de vehículos, Este módulo consolidará la información recopilada en tiempo real sobre los vehículos con y sin marbetes registrados para permitir la toma de decisiones sobre los procesos administrativos.
- Control de vendedores, Este módulo consolidará la reportería acerca de los indicadores de los vendedores.

### **3.1.2. Flujo de información en los procesos**

A lo largo de los procesos mencionados anteriormente, la información se captura desde algún punto de entrada el cual puede ser la aplicación móvil o la plataforma web (clientes) y es enviada a un servidor backend que se encarga de procesar los datos y consultar la información con una base de datos para poder dar una respuesta según la solicitud.

**Figura 10.**

*Flujo de información entre componentes*



*Nota.* Diagrama del flujo de información entre componentes de la plataforma. Elaboración propia, realizado con Canva.

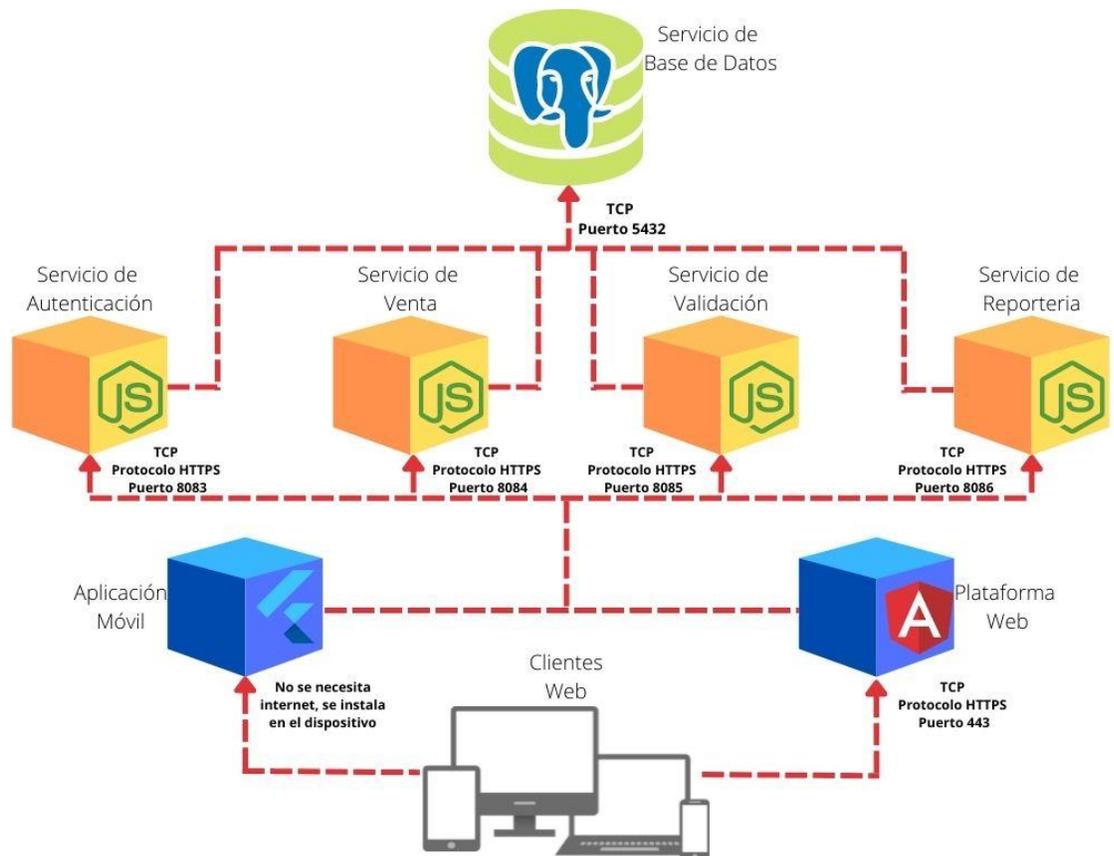
En la imagen anterior podemos observar todos los componentes de manera general que engloba la solución propuesta, inicialmente la plataforma web como la aplicación móvil, denominados clientes, se comunican con el servidor de backend solicitando alguna información según la solicitud, al recibir la solicitud el servidor de backend se comunica con una base de datos con la cual consulta la información según la solicitud y a partir de esta información da una respuesta al cliente.

### **3.1.3. Arquitectura propuesta para la solución**

Para la implementación de la solución a la problemática se propone implementar una arquitectura orientada a servicios dada las ventajas que esta ofrece: flexibilidad, escalabilidad, disponibilidad y desarrollo ágil.

Este tipo de arquitectura incrementa la confiabilidad sobre la plataforma al modularizar toda la lógica de los procesos en componentes más generales encargados de funciones específicas lo que reduce los fallos, ya que, por ejemplo, si un servicio presenta fallos solo una funcionalidad en específico es dada de baja y no el sistema en general.

**Figura 11.**  
*Implementación de arquitectura SOA*



*Nota.* Diagrama de implementación de arquitectura SOA de la plataforma. Elaboración propia, realizado con Canva.

En la imagen anterior podemos observar la implementación de la arquitectura SOA. Inicialmente se cuenta con un servicio de base de datos el cual se encarga de persistir los datos y mantener la integridad sobre los mismos, este servicio se expone a través del puerto por defecto de PostgreSQL el cual es el 5432 por medio del protocolo de red TCP.

Luego se cuenta con la modularización de microservicios en el cual cada uno contiene parte de la lógica del negocio y se comunican con el servicio de base de datos. Estos servicios son:

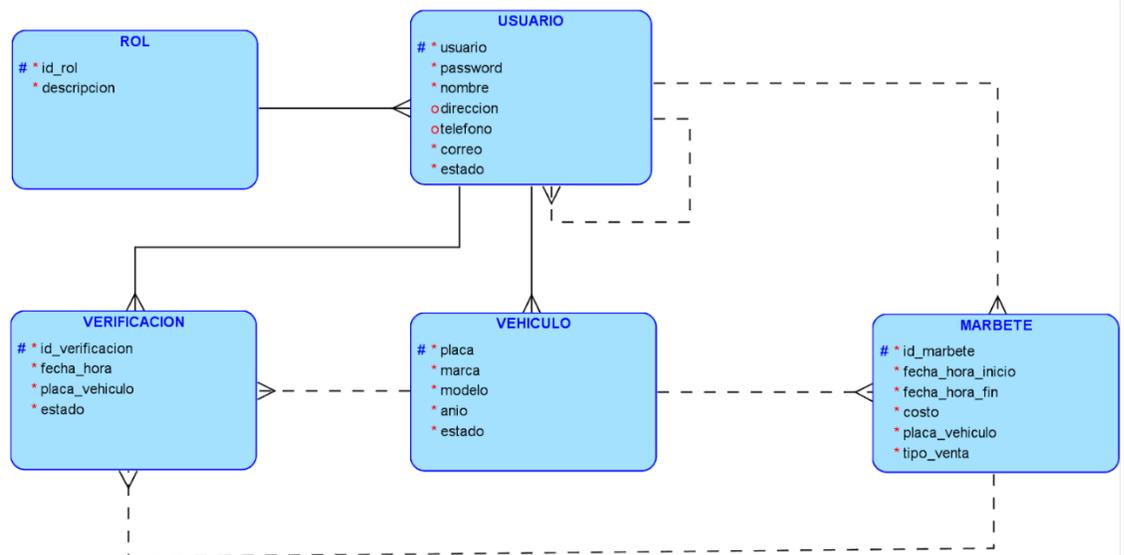
- Servicio de Autenticación, Este servicio engloba todas las operaciones de autenticación del usuario las cuales son:
  - Registro
  - Inicio de sesión
  - Recuperación de contraseña
  - Actualización de contraseña
  - Obtención de datos del usuario
  
- Servicio de Venta, Este servicio se encarga de manejar la lógica en las modalidades de venta mencionadas anteriormente: en sitio y en línea.
- Servicio de Validación, Este servicio incluye las operaciones relacionadas a la validación de la adquisición del marbete digital.
- Servicio de Reportería, Este servicio se encarga de consolidar la reportería útil para el seguimiento de las ventas y verificación de marbetes digitales.

### 3.1.4. Modelo lógico de base de datos

A continuación, se presenta el modelo lógico de base de datos el cual presenta la lógica de negocio de forma general y que no se encuentra ligado a una tecnología en específico.

**Figura 12.**

*Modelo lógico de base de datos*



*Nota.* Diagrama del modelo de lógico de base de datos. Elaboración propia, realizado con SQL Developer Data Modeler.

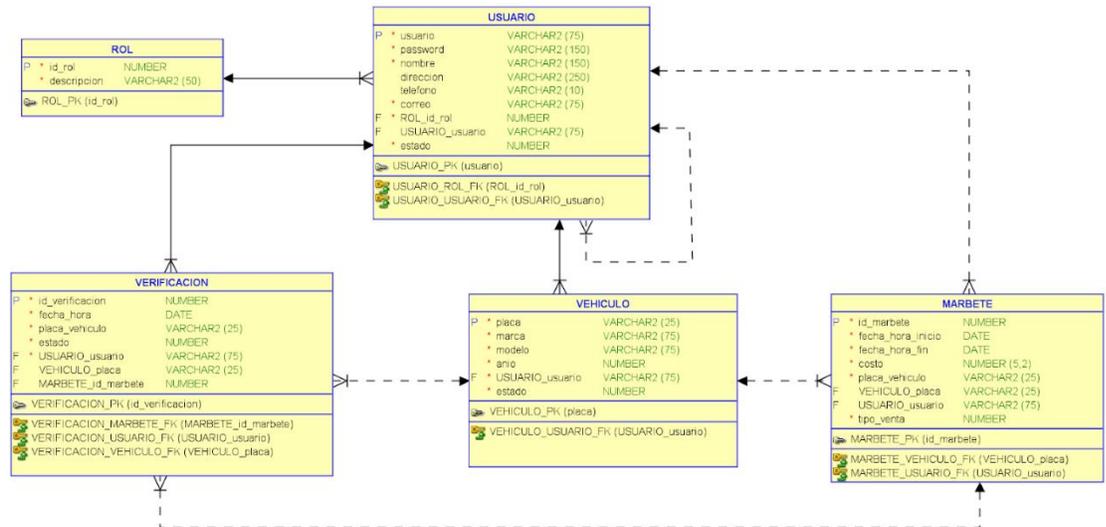
En la imagen anterior podemos observar el diagrama lógico que representa el modelado de la lógica del negocio a un nivel abstracto, sin definir aún el tipo de base de datos a implementar, por lo tanto, la definición de tipos de datos y restricciones tampoco se encuentra, únicamente las relaciones. Se plasman cinco entidades que se describen a continuación:

- Rol, esta entidad es de la categoría catálogo ya que la información en esta es únicamente de consulta y no de carácter operativo. Tiene una relación de carácter obligatorio con la entidad usuario en una multiplicidad de uno a muchos.
- Usuario, esta entidad tiene un carácter operativo sobre la cual se van a realizar operaciones de ingreso, consulta, actualización y eliminación. Tiene una relación con la entidad rol, definida anteriormente, otra de carácter obligatorio con la entidad vehículo en una multiplicidad de uno a muchos, otra relación con la entidad marbete de carácter opcional que soportará las ventas en sitio por parte de un vendedor en una multiplicidad de uno a muchos, otra relación con la entidad verificación de carácter opcional en una multiplicidad de uno a muchos y por último una relación a sí misma de carácter opcional para el soporte de la relación supervisor-vendedor con multiplicidad de uno a muchos .
- Vehículo, esta entidad tiene un carácter operativo, tiene una relación con la entidad usuario, definida anteriormente, otra relación de carácter opcional con la entidad verificación en multiplicidad de uno a muchos y otra de carácter opcional con la entidad marbete en una multiplicidad de uno a muchos. Esta entidad será utilizada para almacenar todos los posibles vehículos que un usuario pueda tener al hacer uso de la plataforma web.
- Marbete, esta entidad tiene un carácter operativo, tiene una relación con la entidad vehículo, definida anteriormente, otra con la entidad usuario de carácter opcional y multiplicidad de uno a muchos y una última relación con la entidad verificación de carácter opcional de multiplicidad uno a muchos. Esta entidad será utilizada para almacenar la venta de marbetes en las dos modalidades, por lo mismo se observa que a pesar de tener la relación con la entidad vehículo cuenta con un atributo placa para los que provengan de la modalidad de venta en sitio.

- Verificación, esta entidad tiene un carácter operativo, tiene relaciones con las entidades usuario, vehículo y marbete definidas anteriormente. Esta entidad será utilizada para llevar el historial de las verificaciones realizadas a los marbetes de los vehículos por medio de las placas.

**Figura 13.**

*Modelo relacional de base de datos*



*Nota.* Diagrama del modelo relacional de base de datos. Elaboración propia, realizado con SQL Developer Data Modeler.

En la imagen anterior, podemos observar la transformación de nuestro modelo lógico a un modelo relacional aplicable a una base de datos PostgreSQL, por ejemplo. En este diagrama podemos observar que ya se encuentran las definiciones de los tipos de datos según los atributos, la definición de llaves primarias y foráneas y restricciones existentes según lo definido en el modelo lógico.

## **3.2. Definición de recursos en servicios backend principales**

Todos los servicios expuestos en el Backend cuentan con una estructura de metadatos y parámetros de entrada que son utilizados para validar y procesar las solicitudes de los distintos clientes, a continuación, se detallan estas definiciones para cada uno de los servicios a implementar.

### **3.2.1. Servicio de autenticación**

El servicio de autenticación es el encargado de toda la gestión de los distintos tipo de usuarios, la obtención de credenciales de autenticación y validación de sesiones.

#### **3.2.1.1. Inicio de sesión**

Este recurso es el encargado de generar los tokens para el acceso a los recursos privados de otros servicios mediante las credenciales del usuario.

**Figura 14.**

*Definición del recurso de inicio de sesión*

<b>Ruta del recurso</b>	/public/auth/login		
<b>Operación HTTP</b>	POST		
<b>Tipo de autenticación</b>	público		
<b>Tipo de contenido</b>	JSON		
<b>Estado de respuesta exitosa</b>	200 OK		
<b>Estado de respuesta con error</b>	400 <i>solicitud incorrecta</i> 500 <i>error interno del servidor</i>		
<b>Parámetros del cuerpo de la solicitud</b>	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	usuario	<i>obligatorio</i>	Nombre de usuario para acceder al sistema
	password	<i>obligatorio</i>	Contraseña del usuario para acceder al sistema
<b>Parámetros de respuesta de la solicitud</b>	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	status	<i>obligatorio</i>	Mensaje con el resultado de la operación
	token	<i>opcional</i>	Cadena encriptada con metadata de la sesión del usuario

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.2.1.2. Registro de visitante

Este recurso es el encargado de registrar un nuevo usuario con rol visitante provenientes de la plataforma web.

**Figura 15.**

*Definición del recurso de registro de visitante*

<b>Ruta del recurso</b>	/public/auth/registrar		
<b>Operación HTTP</b>	POST		
<b>Tipo de autenticación</b>	público		
<b>Tipo de contenido</b>	JSON		
<b>Estado de respuesta exitosa</b>	200 OK		
<b>Estado de respuesta con error</b>	400 <i>solicitud incorrecta</i> 500 <i>error interno del servidor</i>		
<b>Parámetros del cuerpo de la solicitud</b>	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	usuario	obligatorio	Nombre de usuario para acceder al sistema
	password	obligatorio	Contraseña del usuario para acceder al sistema
	nombre	obligatorio	Nombre de la persona que crea el usuario
	direccion	obligatorio	Dirección de la persona que crea el usuario
	telefono	obligatorio	Teléfono de la persona que crea el usuario
	correo	obligatorio	Correo Electrónico de la persona que crea el usuario
	<b>Parámetros de respuesta de la solicitud</b>	<b>Parámetro</b>	<b>Tipo</b>
status		obligatorio	Mensaje con el resultado de la operación

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.2.1.3. Obtención de usuarios vendedores

Este recurso se encarga de retornar toda la información de los usuarios con rol vendedor.

**Figura 16.**

*Definición del recurso de obtención de vendedores*

<b>Ruta del recurso</b>	/private/usuario/vendedores		
<b>Operación HTTP</b>	GET		
<b>Tipo de autenticación</b>	Bearer Token en cabecera de autorización		
<b>Tipo de contenido</b>	JSON		
<b>Estado de respuesta exitosa</b>	<b>200</b> OK <b>204</b> sin contenido		
<b>Estado de respuesta con error</b>	<b>401</b> no autorizado <b>500</b> error interno del servidor		
<b>Parámetros del cuerpo de la solicitud</b>	No solicita parámetros en el cuerpo de la solicitud		
<b>Parámetros de respuesta de la solicitud</b>	<i>*Pueden venir 0 o más estructuras con los parámetros definidos</i>		
	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	<b>usuario</b>	<i>obligatorio</i>	Nombre del usuario
	<b>nombre</b>	<i>obligatorio</i>	Nombre de la persona
	<b>direccion</b>	<i>obligatorio</i>	Direccion de la persona
	<b>telefono</b>	<i>obligatorio</i>	Teléfono de la persona
	<b>correo</b>	<i>obligatorio</i>	Correo electrónico de la persona
	<b>id_rol</b>	<i>obligatorio</i>	Identificador único del rol del usuario
	<b>usuario_padre</b>	<i>opcional</i>	Identificador único del usuario que creó el usuario

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.2.2. Servicio de venta

El servicio de venta es el encargado de toda la administración de marbetes digitales y gestión de vehículos para los usuarios de la plataforma web.

#### 3.2.2.1. Registro de vehículo del visitante

Este recurso se encarga de registrar un nuevo vehículo de un usuario con rol visitante desde la plataforma web.

#### Figura 17.

*Definición del recurso de registro de vehículo del visitante*

<b>Ruta del recurso</b>	/private/vehiculo/regar		
<b>Operación HTTP</b>	POST		
<b>Tipo de autenticación</b>	Bearer Token en cabecera de autorización		
<b>Tipo de contenido</b>	JSON		
<b>Estado de respuesta exitosa</b>	200 OK		
<b>Estado de respuesta con error</b>	400 <i>solicitud incorrecta</i> 401 <i>no autorizado</i> 500 <i>error interno del servidor</i>		
<b>Parámetros del cuerpo de la solicitud</b>	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	placa	<i>obligatorio</i>	Identificador único del vehículo
	marca	<i>obligatorio</i>	Marca del vehículo
	modelo	<i>obligatorio</i>	Modelo del vehículo
	anio	<i>obligatorio</i>	Año del vehículo
	usuario	<i>obligatorio</i>	Usuario que crea el vehículo
<b>Parámetros de respuesta de la solicitud</b>	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	status	<i>obligatorio</i>	Mensaje con el resultado de la operación

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.2.2.2. Obtención de vehículos del visitante

Este recurso se encarga de obtener todos los vehículos registrados de un usuario con rol vendedor.

**Figura 18.**

*Definición del recurso de obtención de vehículos*

<b>Ruta del recurso</b>	/private/vehiculo/usuario?usuario=<nombre del usuario>		
<b>Operación HTTP</b>	GET		
<b>Tipo de autenticación</b>	Bearer Token en cabecera de autorización		
<b>Tipo de contenido</b>	JSON		
<b>Estado de respuesta exitosa</b>	200 OK 204 sin contenido		
<b>Estado de respuesta con error</b>	400 solicitud incorrecta 401 no autorizado 500 error interno del servidor		
<b>Parámetros del cuerpo de la solicitud</b>	No solicita parámetros en el cuerpo de la solicitud		
<b>Parámetros de respuesta de la solicitud</b>	<i>*Pueden venir 0 o más estructuras con los parámetros definidos</i>		
	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	placa	obligatorio	Identificador único del vehículo
	marca	obligatorio	Marca del vehículo
	modelo	obligatorio	Modelo del vehículo
	anio	obligatorio	Año del vehículo
	usuario	obligatorio	Usuario que creó el vehículo
	estado	obligatorio	Valor número entre 0 y 1 que define el estado del vehículo
	marbete	obligatorio	Valor número entre 0 y 1 que define si el vehículo cuenta con marbete activo

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.2.2.3. Registrar marbete sobre un vehículo

Este recurso se encarga de registrar un marbete digital, válido por un día y con un costo definido, sobre un vehículo previamente registrado por un usuario del tipo visitante a través de la plataforma web.

**Figura 19.**

*Definición del recurso de registro de marbete*

<b>Ruta del recurso</b>	/private/marbete/registrar		
<b>Operación HTTP</b>	POST		
<b>Tipo de autenticación</b>	Bearer Token en cabecera de autorización		
<b>Tipo de contenido</b>	JSON		
<b>Estado de respuesta exitosa</b>	200 OK		
<b>Estado de respuesta con error</b>	400 <i>solicitud incorrecta</i> 401 <i>no autorizado</i> 500 <i>error interno del servidor</i>		
<b>Parámetros del cuerpo de la solicitud</b>	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	costo	obligatorio	Costo del marbete
	placa	obligatorio	Identificador único del vehículo sobre el cual se adquiere el marbete
	usuario	obligatorio	Usuario que registra la compra del marbete
	tipo_venta	obligatorio	Identificador único que define el tipo de venta
<b>Parámetros de respuesta de la solicitud</b>	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	status	obligatorio	Mensaje con el resultado de la operación

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.2.2.4. Obtención de marbetes sobre un vehículo específico

Este recurso se encarga de obtener un histórico de todos los marbetes que han sido adquiridos por un vehículo en específico.

**Figura 20.**

*Definición de recurso de obtención de marbetes*

<b>Ruta del recurso</b>	/private/marbete/vehiculo?placa=<placa del vehículo>		
<b>Operación HTTP</b>	GET		
<b>Tipo de autenticación</b>	Bearer Token en cabecera de autorización		
<b>Tipo de contenido</b>	JSON		
<b>Estado de respuesta exitosa</b>	200 OK 204 sin contenido		
<b>Estado de respuesta con error</b>	400 solicitud incorrecta 401 no autorizado 500 error interno del servidor		
<b>Parámetros del cuerpo de la solicitud</b>	No solicita parámetros en el cuerpo de la solicitud		
<b>Parámetros de respuesta de la solicitud</b>	<i>*Pueden venir 0 o más estructuras con los parámetros definidos</i>		
	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	id_marbete	obligatorio	Identificador único del marbete
	fecha_hora_inicio	obligatorio	Fecha y hora que se adquirió el marbete
	fecha_hora_fin	obligatorio	Fecha y hora en la que vence el marbete
	costo	obligatorio	Costo del marbete
	placa	obligatorio	Identificador único del vehículo que adquiere el marbete
	ref_placa	opcional	Identificador único del vehículo que adquiere el marbete y se encuentra previamente registrado
	usuario	obligatorio	Usuario que adquiere el marbete
	tipo_venta	obligatorio	Valor numérico que define el tipo de la venta del marbete

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.2.3. Servicio de validación

El servicio de validación es el encargado de toda la gestión de validación de marbetes vendidos, con el fin de llevar un control sobre todos los vehículos parqueados dentro de la ciudad.

#### 3.2.3.1. Registrar verificación de marbete

Este recurso se encarga de registrar una verificación de marbete a partir del número de placa del vehículo y retorna si este cuenta con un marbete valido o no.

#### Figura 21.

*Definición del recurso de registro de verificación*

<b>Ruta del recurso</b>	/private/verificacion/registrar		
<b>Operación HTTP</b>	POST		
<b>Tipo de autenticación</b>	Bearer Token en cabecera de autorización		
<b>Tipo de contenido</b>	JSON		
<b>Estado de respuesta exitosa</b>	200 OK		
<b>Estado de respuesta con error</b>	400 <i>solicitud incorrecta</i> 401 <i>no autorizado</i> 500 <i>error interno del servidor</i>		
<b>Parámetros del cuerpo de la solicitud</b>	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	placa	obligatorio	Identificador único del vehículo sobre el cual se realiza la verificación
	usuario	obligatorio	Usuario que realiza la verificación
<b>Parámetros de respuesta de la solicitud</b>	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	status	obligatorio	Mensaje con el resultado de la operación

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.2.3.2. Obtención de verificaciones realizadas por un usuario

Este recurso se encarga de obtener un histórico de todas las verificaciones realizadas por un usuario con rol vendedor.

**Figura 22.**

*Definición del recurso para la obtención de verificaciones*

<b>Ruta del recurso</b>	/private/verificacion/usuario/<nombre del usuario>		
<b>Operación HTTP</b>	GET		
<b>Tipo de autenticación</b>	Bearer Token en cabecera de autorización		
<b>Tipo de contenido</b>	JSON		
<b>Estado de respuesta exitosa</b>	200 OK 204 sin contenido		
<b>Estado de respuesta con error</b>	400 solicitud incorrecta 401 no autorizado 500 error interno del servidor		
<b>Parámetros del cuerpo de la solicitud</b>	No solicita parámetros en el cuerpo de la solicitud		
<b>Parámetros de respuesta de la solicitud</b>	<i>*Pueden venir 0 o más estructuras con los parámetros definidos</i>		
	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	id_verificacion	obligatorio	Identificador único de la verificación
	fecha_hora	obligatorio	Fecha y hora que se realizó la verificación
	placa	obligatorio	Identificador único del vehículo sobre el cual se realizó la verificación
	estado	obligatorio	Valor numérico entre 0 y 1 que define el resultado de la verificación
	usuario	obligatorio	Usuario que realiza la verificación
	ref_placa	opcional	Referencia a un vehículo registrado dentro de la plataforma
	id_marbete	obligatorio	Referencia a el marbete si la verificación encontró coincidencia válida

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.2.4. Servicio de reportería

El servicio de reportaría es el encargado de recopilar y exponer toda la información necesaria para generar reportes en la plataforma web.

#### 3.2.4.1. Reporte de rendimiento de vendedores

Este recurso es el encargado de exponer la cantidad de marbetes vendidos y verificados de los usuarios con rol vendedor.

#### Figura 23.

*Definición del recurso de rendimiento de vendedores*

<b>Ruta del recurso</b>	/private/reporteria/rendimiento												
<b>Operación HTTP</b>	GET												
<b>Tipo de autenticación</b>	Bearer Token en cabecera de autorización												
<b>Tipo de contenido</b>	JSON												
<b>Estado de respuesta exitosa</b>	200 OK 204 sin contenido												
<b>Estado de respuesta con error</b>	400 solicitud incorrecta 401 no autorizado 500 error interno del servidor												
<b>Parámetros del cuerpo de la solicitud</b>	No solicita parámetros en el cuerpo de la solicitud												
<b>Parámetros de respuesta de la solicitud</b>	<i>*Pueden venir 0 o más estructuras con los parámetros definidos</i> <table border="1"><thead><tr><th>Parámetro</th><th>Tipo</th><th>Descripción</th></tr></thead><tbody><tr><td>usuario</td><td>obligatorio</td><td>Nombre del usuario vendedor</td></tr><tr><td>marbetes</td><td>obligatorio</td><td>Valor numérico que define la cantidad de marbetes vendidos por el usuario</td></tr><tr><td>verificaciones</td><td>obligatorio</td><td>Valor numérico que define la cantidad de verificaciones realizadas por el usuario</td></tr></tbody></table>	Parámetro	Tipo	Descripción	usuario	obligatorio	Nombre del usuario vendedor	marbetes	obligatorio	Valor numérico que define la cantidad de marbetes vendidos por el usuario	verificaciones	obligatorio	Valor numérico que define la cantidad de verificaciones realizadas por el usuario
Parámetro	Tipo	Descripción											
usuario	obligatorio	Nombre del usuario vendedor											
marbetes	obligatorio	Valor numérico que define la cantidad de marbetes vendidos por el usuario											
verificaciones	obligatorio	Valor numérico que define la cantidad de verificaciones realizadas por el usuario											

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.2.4.2. Reporte de ventas de vendedores

Este recurso es el encargado de exponer el histórico de las últimas ventas realizadas con el fin de llevar un monitoreo en tiempo real de las transacciones que se realizan a través de la plataforma.

**Figura 24.**

*Definición del recurso del reporte de ventas*

<b>Ruta del recurso</b>	/private/reporteria/usuario/ventas		
<b>Operación HTTP</b>	GET		
<b>Tipo de autenticación</b>	Bearer Token en cabecera de autorización		
<b>Tipo de contenido</b>	JSON		
<b>Estado de respuesta exitosa</b>	200 OK 204 sin contenido		
<b>Estado de respuesta con error</b>	400 solicitud incorrecta 401 no autorizado 500 error interno del servidor		
<b>Parámetros del cuerpo de la solicitud</b>	No solicita parámetros en el cuerpo de la solicitud		
<b>Parámetros de respuesta de la solicitud</b>	<i>*Pueden venir 0 o más estructuras con los parámetros definidos</i>		
	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	placa	obligatorio	Identificador único del vehículo que adquiere el marbete
	tipo_venta	obligatorio	Valor numérico que identifica el tipo de venta que se realizó
	usuario	obligatorio	Nombre del usuario que realizó la venta del marbete
fecha_hora_inicio	obligatorio	Fecha y hora en la que se realizó la venta del marbete	

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.2.4.3. Reporte de verificaciones de vendedores

Este recurso es el encargado de exponer el histórico de las últimas verificaciones realizadas con el fin de llevar un monitoreo en tiempo real de las transacciones que se realizan a través de la plataforma.

**Figura 25.**

*Definición del recurso del reporte de verificaciones*

<b>Ruta del recurso</b>	/private/reporteria/usuario/verificaciones		
<b>Operación HTTP</b>	GET		
<b>Tipo de autenticación</b>	Bearer Token en cabecera de autorización		
<b>Tipo de contenido</b>	JSON		
<b>Estado de respuesta exitosa</b>	200 OK 204 sin contenido		
<b>Estado de respuesta con error</b>	400 solicitud incorrecta 401 no autorizado 500 error interno del servidor		
<b>Parámetros del cuerpo de la solicitud</b>	No solicita parámetros en el cuerpo de la solicitud		
<b>Parámetros de respuesta de la solicitud</b>	<i>*Pueden venir 0 o más estructuras con los parámetros definidos</i>		
	<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
	placa	obligatorio	Identificador único del vehículo sobre el cual se realizó la verificación
	usuario	obligatorio	Nombre del usuario que realizó la verificación
	estado	obligatorio	Valor numérico entre 0 y 1 que define el resultado de la verificación
fecha_hora	obligatorio	Fecha y hora en la que se realizó la verificación	

*Nota.* Ficha técnica del recurso del API REST. Elaboración propia, realizado con Canva.

### 3.3. Interfaz gráfica de la plataforma web

A continuación se presentan las pantallas principales que se proponen implementar para la plataforma web que será utilizada tanto por los usuarios visitantes para la gestión de sus vehículos y adquisición de marbetes como también por los usuarios administradores para la administración de vendedores y el seguimiento en tiempo real de las transacciones a través de la reportería.

#### 3.3.1. Inicio de sesión

Esta pantalla es el punto inicial de la plataforma web desde la cual, a partir de las credenciales del usuario podrá ingresar a las distintas funcionalidades según su rol.

#### Figura 26.

*Página de inicio de sesión*



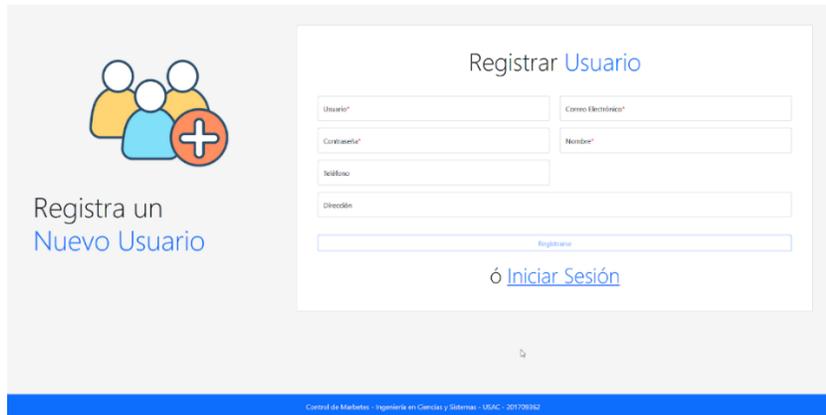
*Nota.* Maquetado de la interfaz de la plataforma web. Elaboración propia, realizado con Angular.

### 3.3.2. Registro

Pantalla desde la cual cualquier persona que no cuente con un usuario puede registrarse y generar un nuevo usuario con rol visitante.

#### Figura 27.

*Página de registro de usuario*



Registra un Nuevo Usuario

Registrar Usuario

Usuario\*      Correo Electrónico\*

Contraseña\*      Nombre\*

Teléfono

Dirección

Registrar

ó [Iniciar Sesión](#)

Control de Marbetes - Ingeniería en Ciencias y Sistemas - USAC - 2017/2022

*Nota.* Maquetado de la interfaz de la plataforma web. Elaboración propia, realizado con Angular.

### 3.3.3. Gestión de vehículos del visitante

Pantalla principal para los usuarios de rol visitante desde la cual se puede realizar la gestión de vehículos y adquisición de marbetes.

**Figura 28.**

*Página de gestión de vehículos del visitante*



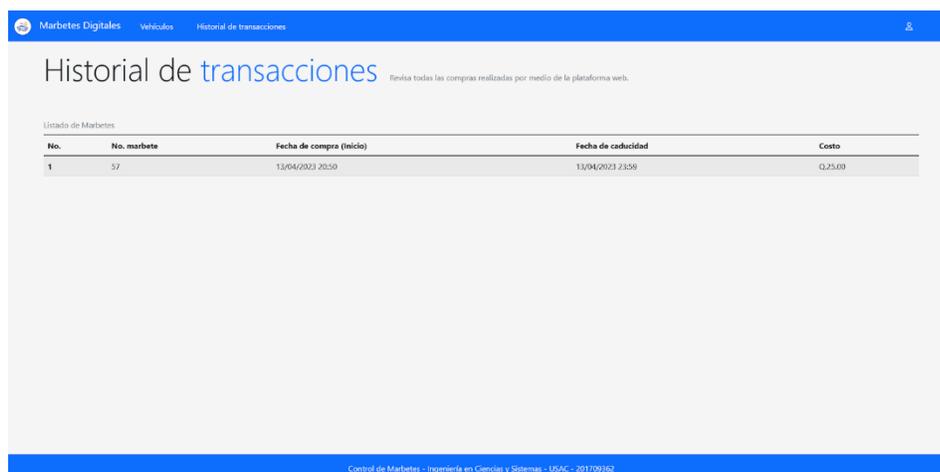
*Nota.* Maquetado de la interfaz de la plataforma web. Elaboración propia, realizado con Angular.

### 3.3.4. Historial de transacciones del usuario

Pantalla con el histórico de todas las transacciones realizadas por un usuario dentro de la plataforma.

**Figura 29.**

*Página del historial de transacciones del usuario*



*Nota.* Maquetado de la interfaz de la plataforma web. Elaboración propia, realizado con Angular.

### 3.3.5. Control de vendedores

Pantalla principal de los usuarios con rol administrador, desde la cual se gestiona todo lo relacionado a los usuarios con rol vendedor.

#### Figura 30.

*Página de control de vendedores*

No.	Nombre	Correo	Usuario
1	vendedorv2	vendedorv2@correx.com	vendedor
2	ventas	ventas@mail.com	ventas1

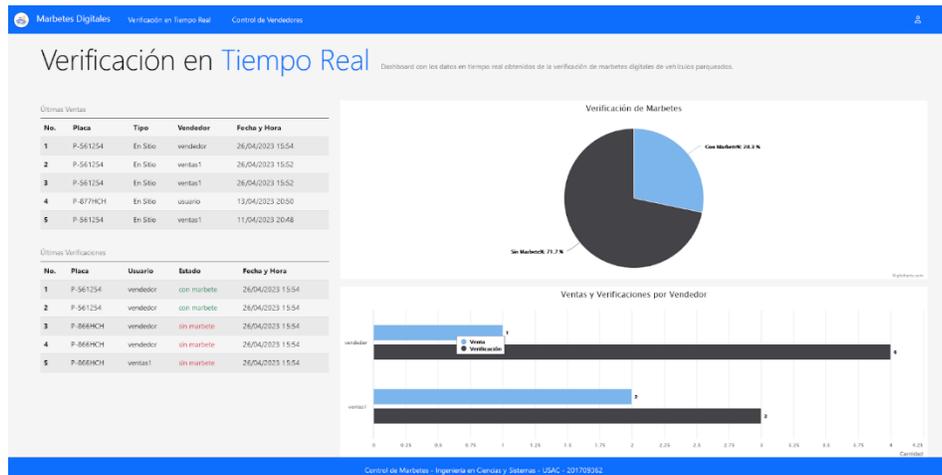
*Nota.* Maquetado de la interfaz de la plataforma web. Elaboración propia, realizado con Angular.

### 3.3.6. Verificación en tiempo real

Pantalla con la reportería principal para el seguimiento de la venta, verificación y rendimiento de los vendedores en tiempo real.

**Figura 31.**

*Página de verificación en tiempo real*



*Nota.* Maquetado de la interfaz de la plataforma web. Elaboración propia, realizado con Angular.

### 3.4. Interfaz gráfica de la aplicación móvil

A continuación se presentan las pantallas principales que se proponen implementar para la aplicación móvil que tiene como función principal ser utilizada por usuarios con rol vendedor y desde la cual puedan gestionar la venta y verificación de marbetes.

#### 3.4.1. Inicio de sesión

Pantalla de inicio de la aplicación móvil, desde la cual los usuarios deben ingresar sus credenciales para poder acceder a otras funcionalidades.

## Figura 32.

*Pantalla de inicio de sesión*



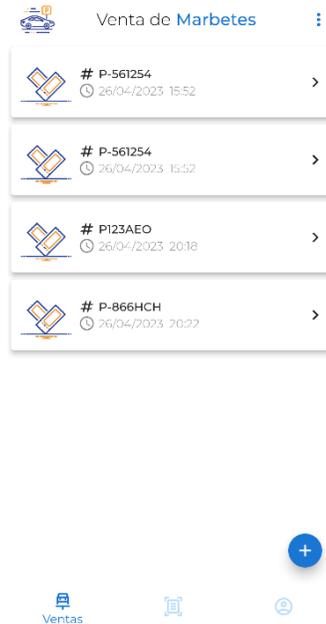
*Nota.* Maquetado de la interfaz de la aplicación móvil. Elaboración propia, realizado con Flutter.

### 3.4.2. Listado de marbetes vendidos

Pantalla con el detalle de todos los marbetes vendidos por un usuario en el día actual y desde la cual se puede acceder a ver el detalle de cada uno.

**Figura 33.**

*Listado de venta de marbetes*



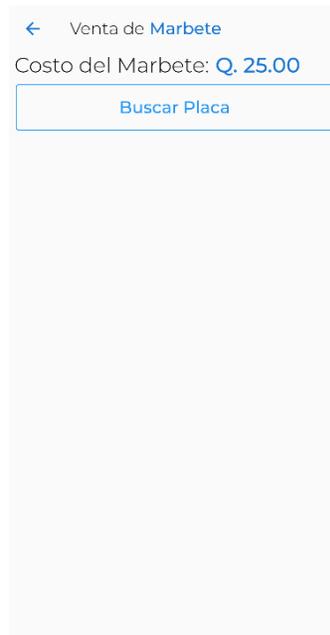
*Nota.* Maquetado de la interfaz de la aplicación móvil. Elaboración propia, realizado con Flutter.

### 3.4.3. Venta de marbete

Pantalla desde la cual un usuario puede realizar la venta de un marbete ya sea escaneando la placa o ingresándola manualmente.

## Figura 34.

### *Pantalla de venta de marbete*



*Nota.* Maquetado de la interfaz de la aplicación móvil. Elaboración propia, realizado con Flutter.

#### **3.4.4. Detalle de marbete**

Pantalla desde la cual se describe la información de la venta del marbete y del vehículo, en caso existiera un vehículo registrado por la plataforma web con la placa ingresada.

## Figura 35.

### *Pantalla de detalle de marbete*



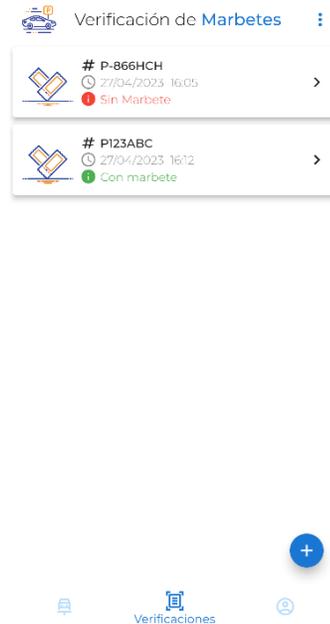
*Nota.* Maquetado de la interfaz de la aplicación móvil. Elaboración propia, realizado con Flutter.

### 3.4.5. Listado de verificaciones

Pantalla con el detalle de todas las verificaciones realizadas por un usuario en el día actual y desde la cual se puede acceder a ver el detalle de cada uno.

**Figura 36.**

*Pantalla de listado de verificaciones*



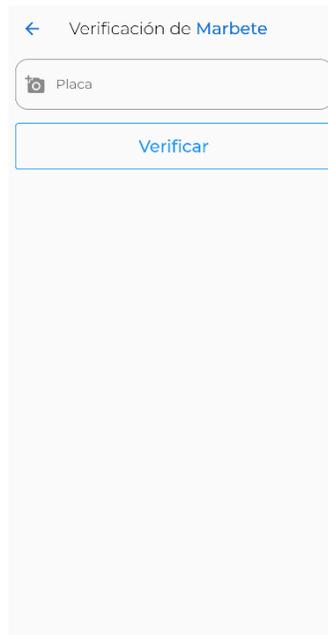
*Nota.* Maquetado de la interfaz de la aplicación móvil. Elaboración propia, realizado con Flutter.

### **3.4.6. Verificación de marbete**

Pantalla desde la cual un usuario puede registrar la verificación de un marbete ya sea escaneando la placa o ingresándola manualmente.

### **Figura 37.**

#### *Pantalla de verificación de marbete*



*Nota.* Maquetado de la interfaz de la aplicación móvil. Elaboración propia, realizado con Flutter.

#### **3.4.7. Detalle de verificación de marbete**

Pantalla desde la cual se describe la información de la verificación del marbete y del vehículo, en caso existiera un vehículo registrado por la plataforma web con la placa ingresada.

### Figura 38.

*Pantalla de detalle de verificación de marbete*



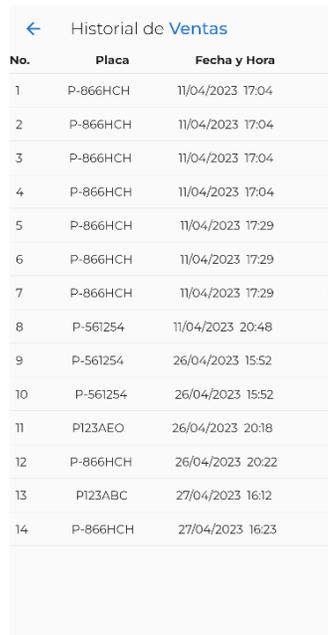
*Nota.* Maquetado de la interfaz de la aplicación móvil. Elaboración propia, realizado con Flutter.

#### 3.4.8. Histórico de venta de marbetes

Pantalla con el histórico de todas las ventas realizadas por un usuario desde la creación del usuario.

### Figura 39.

*Pantalla con el histórico de venta de marbetes*



The screenshot shows a mobile application interface with a title bar containing a back arrow and the text 'Historial de Ventas'. Below the title bar is a table with three columns: 'No.', 'Placa', and 'Fecha y Hora'. The table contains 14 rows of data, each with a right-pointing chevron icon. The data is as follows:

No.	Placa	Fecha y Hora
1	P-866HCH	11/04/2023 17:04
2	P-866HCH	11/04/2023 17:04
3	P-866HCH	11/04/2023 17:04
4	P-866HCH	11/04/2023 17:04
5	P-866HCH	11/04/2023 17:29
6	P-866HCH	11/04/2023 17:29
7	P-866HCH	11/04/2023 17:29
8	P-561254	11/04/2023 20:48
9	P-561254	26/04/2023 15:52
10	P-561254	26/04/2023 15:52
11	P123AEO	26/04/2023 20:18
12	P-866HCH	26/04/2023 20:22
13	P123ABC	27/04/2023 16:12
14	P-866HCH	27/04/2023 16:23

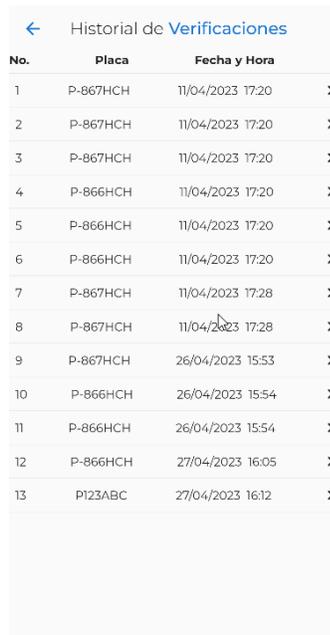
*Nota.* Maquetado de la interfaz de la aplicación móvil. Elaboración propia, realizado con Flutter.

#### 3.4.9. Histórico de verificación de marbetes

Pantalla con el histórico de todas las verificaciones realizadas por un usuario desde la creación del usuario.

**Figura 40.**

*Pantalla con el histórico de verificación de marbetes*



No.	Placa	Fecha y Hora	>
1	P-867HCH	11/04/2023 17:20	>
2	P-867HCH	11/04/2023 17:20	>
3	P-867HCH	11/04/2023 17:20	>
4	P-866HCH	11/04/2023 17:20	>
5	P-866HCH	11/04/2023 17:20	>
6	P-866HCH	11/04/2023 17:20	>
7	P-867HCH	11/04/2023 17:28	>
8	P-867HCH	11/04/2023 17:28	>
9	P-867HCH	26/04/2023 15:53	>
10	P-866HCH	26/04/2023 15:54	>
11	P-866HCH	26/04/2023 15:54	>
12	P-866HCH	27/04/2023 16:05	>
13	P123ABC	27/04/2023 16:12	>

*Nota.* Maquetado de la interfaz de la aplicación móvil. Elaboración propia, realizado con Flutter.

## 4. SIMULACIÓN DE LOS PROCESOS INVOLUCRADOS

### 4.1. Características del sistema

Actualmente el cobro de marbete dentro de la Antigua Guatemala se lleva a cabo en puntos específicos de la ciudad, a partir de la consulta al personal a cargo se definió que actualmente existen 4 vendedores que rotan en los siguientes puntos:

- 3ra. calle, Casa Santo Domingo, Antigua Guatemala, Sacatepéquez.
- 3ra. calle y 3ra avenida, Liceo Antiguero, Antigua Guatemala, Sacatepéquez.
- 5ta. calle, esquina de la Catedral, Antigua Guatemala, Sacatepéquez.
- Parque Tanque de la Unión, Antigua Guatemala, Sacatepéquez.
- Calle del Arco, Antigua Guatemala, Sacatepéquez.
- Cerro de la Cruz, Antigua Guatemala, Sacatepéquez.

A partir de esta consulta también se obtuvo información respecto a las características principales para poder realizar modelos de simulación que permitan estudiar los procesos que se llevan actualmente y realizar propuestas o recomendaciones. Estos parámetros se definen en la siguiente tabla:

**Tabla 1.**

*Definición de características principales del sistema*

<b>Característica</b>	<b>Valor</b>
Vehículos que ingresan a la ciudad por hora	88
Porcentaje de vehículos que adquieren marbete	45 %

Continuación de la tabla 1

<b>Característica</b>	<b>Valor</b>
Porcentaje de vehículos que no adquieren marbete	55 %
Tiempo de venta de un marbete máximo (proceso actual)	25 minutos
Tiempo de venta de un marbete mínimo (proceso actual)	7.5 minutos
Tiempo de venta de un marbete máximo (solución propuesta)	15 minutos
Tiempo de venta de un marbete mínimo (solución propuesta)	5 minutos
Hora de inicio	07:00 AM
Hora fin	07:00 PM

*Nota.* Detalle de las características principales que conforman el sistema. Elaboración propia, realizado con Microsoft Word.

Al analizar los datos, se propone la siguiente composición para los modelos de simulación en ambos escenarios (actual y propuesta):

- Una entrada de vehículos con una tasa de llegadas definida por una distribución exponencial con una media de 1.46 vehículos por minuto (88 vehículos por hora).
- Cuatro servicios que simulan a los vendedores con una tasa de servicio definida por una distribución uniforme con un mínimo de 5 y máximo de 16.25 minutos para el modelo del proceso actual y una distribución uniforme con un mínimo de 5 y máximo de 15 minutos para el modelo con la solución propuesta.
- Para cada vendedor existe una probabilidad del 45 % de vender un marbete, el resto sale del sistema.
- Para cada vendedor existe una cola única de vehículos, la cual representa el tráfico de la ciudad.
- Dos salidas del sistema.
- Un horario de trabajo de 12 horas, de 7AM a 7PM, el cual cubre los turnos definidos en la información obtenida.

## 4.2. Vistas del modelo de simulación

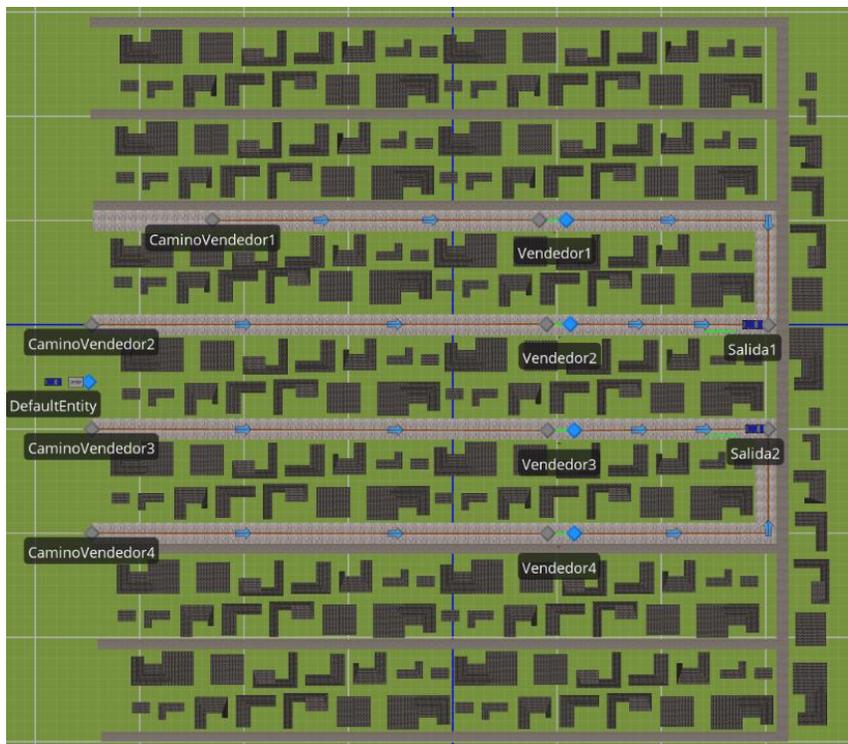
A continuación, se presenta el modelo de simulación propuesto de manera grafica en el cual se detallan los componentes existentes en el modelo actual.

### 4.2.1. Vista aérea

Vista del modelo desde una perspectiva aérea en el cual se visualizan todos los componentes que interactúan en el modelo además de elementos decorativos que agregan más contexto al modelo en general.

#### Figura 41.

*Vista aérea del modelo*



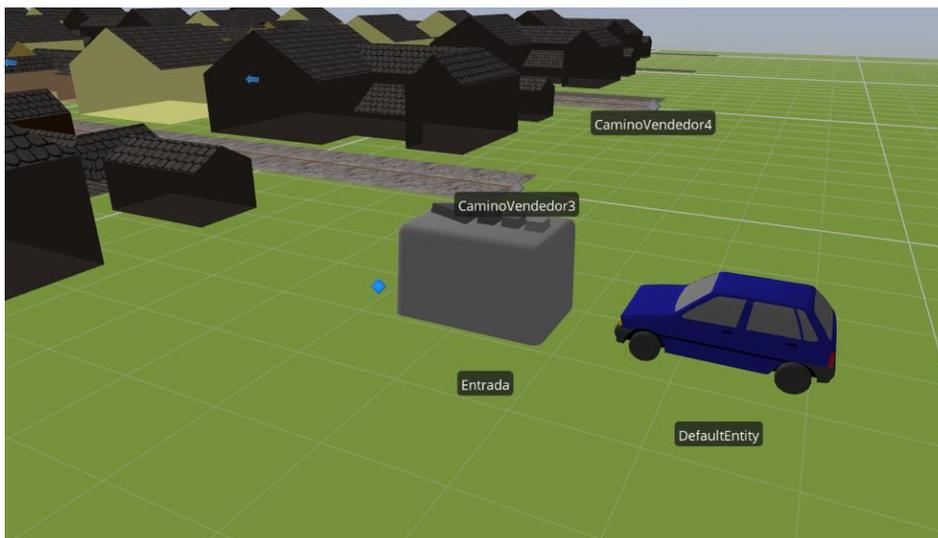
*Nota.* Vista del maquetado del modelo de simulación. Elaboración propia, realizado con Simio.

### 4.2.2. Entrada del sistema

Vista con los componentes de entrada al sistema desde los cuales se generan las entidades que van a interactuar con todos los demás componentes, en este caso se ven representados como un vehículo.

**Figura 42.**

*Vista de la entrada al sistema*



*Nota.* Vista del maquetado del modelo de simulación. Elaboración propia, realizado con Simio.

### 4.2.3. Vendedores

Vista de las áreas de servicio, representadas por medio de vendedores que atienden a los visitantes en vehículo.

### Figura 43.

*Vendedores atendiendo visitantes*



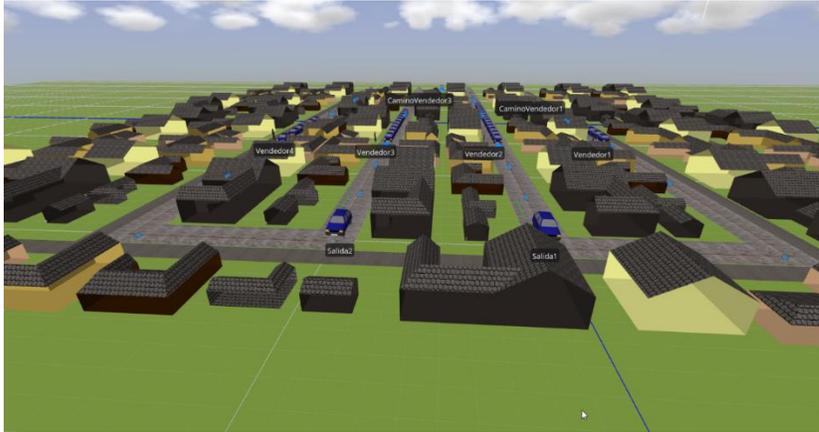
*Nota.* Vista del maquetado del modelo de simulación. Elaboración propia, realizado con Simio.

#### 4.2.1. Salida del sistema

Vista de los componentes que finalizan el proceso de simulación para las entidades, se representan por medio de calles que destruyen las entidades.

## Figura 44.

*Vista de la salida del modelo*



*Nota.* Vista del maquetado del modelo de simulación. Elaboración propia, realizado con Simio.

### 4.3. Configuración de los componentes del modelo de simulación

A continuación, se detalla las configuraciones utilizadas dentro del modelo de simulación para representar tanto el flujo actual como el flujo propuesto.

- Entrada: generación de vehículos según una distribución exponencial con media de 1.46 vehículos por minuto.
- Caminos: paso entre vehículos desactivado.
- Vendedores: atención de un vehículo a la vez, con una tasa de servicio uniforme con mínimo de 5 y máximo de 15 minutos para el modelo con la solución propuesta y para el modelo del proceso actual mínimo de 7.5 y máximo de 25 minutos. Adicional una probabilidad del 55 % de que el vehículo no compre un marbete.
- Salidas: Configuraciones por defecto.

#### 4.4. Resultados del modelo de simulación del proceso actual

A partir de las variables previamente definidas, se ejecutó el modelo de simulación el cual dio los resultados definidos en la siguiente tabla.

**Tabla 2.**

*Resultados del modelo de simulación del proceso actual*

<b>Resultado</b>	<b>Valor</b>
Cantidad de vehículos que ingresan a la ciudad	526
Tiempo promedio de un vehículo en el tráfico en horas	1.9 (114 minutos)
Cantidad de marbetes vendidos por el vendedor 1	41
Tiempo promedio de venta de marbete del vendedor 1 (horas)	0.28 (16.8 minutos)
Cantidad de marbetes vendidos por el vendedor 2	45
Tiempo promedio de venta de marbete del vendedor 2 (horas)	0.26 (15.6 minutos)
Cantidad de marbetes vendidos por el vendedor 3	45
Tiempo promedio de venta de marbete del vendedor 3 (horas)	0.25 (15 minutos)
Cantidad de marbetes vendidos por el vendedor 4	43
Tiempo promedio de venta de marbete del vendedor 4 (horas)	0.275 (16.5 minutos)
Total de marbetes vendidos	174
Tiempo promedio de venta (horas)	0.27 (15.96 minutos)

*Nota.* Detalle de resultados de modelo de simulación del proceso actual. Elaboración propia, realizado con Microsoft Word.

#### 4.5. Resultados del modelo de simulación con la solución propuesta

A partir de las variables previamente definidas, se ejecutó el modelo de simulación el cual dio los resultados definidos en la siguiente tabla.

**Tabla 3.**

*Resultados del modelo de simulación de la solución propuesta*

<b>Resultado</b>	<b>Valor</b>
Cantidad de vehículos que ingresan a la ciudad	488
Tiempo promedio de un vehículo en el tráfico en horas	0.24 (14.4 minutos)
Cantidad de marbetes vendidos por el vendedor 1	54
Tiempo promedio de venta de marbete del vendedor 1 (horas)	0.156 (9.36 minutos)
Cantidad de marbetes vendidos por el vendedor 2	63
Tiempo promedio de venta de marbete del vendedor 2 (horas)	0.166 (9.96 minutos)
Cantidad de marbetes vendidos por el vendedor 3	42
Tiempo promedio de venta de marbete del vendedor 3 (horas)	0.17 (10.02 minutos)
Cantidad de marbetes vendidos por el vendedor 4	55
Tiempo promedio de venta de marbete del vendedor 4 (horas)	0.177 (10.6 minutos)
Total de marbetes vendidos	214
Tiempo promedio de venta (horas)	0.17 (10.02 minutos)

*Nota.* Detalle de resultados de modelo de simulación de la solución propuesta. Elaboración propia, realizado con Microsoft Word.

#### **4.6. Análisis comparativo entre los resultados de la simulación**

A partir de los datos obtenidos en los modelos de simulación, se realizó un cuadro comparativo entre los resultados, el cual se define en la siguiente tabla.

**Tabla 4.**

*Comparación resultados proceso actual y solución propuesta*

<b>Resultado</b>	<b>Modelo Original</b>	<b>Modelo Propuesto</b>	<b>Mejora</b>
Tiempo promedio de un vehículo en el tráfico en minutos	114	14.4	-87.4 %
Tiempo promedio de venta en minutos	15.96	10.02	-37.2 %

*Nota.* Comparación entre los resultados de los modelos de simulación. Elaboración propia, realizado con Microsoft Word.

A partir de este cuadro comparativo, podemos definir que:

- El tiempo que un vehículo pasa en el tráfico debido a la venta de un marbete se reduce de 114 a 14.4 minutos, lo que se traduce en una mejora de un 87.4 %.
- El tiempo promedio que un vendedor invierte para finalizar la venta de un marbete se reduce de 15.96 a 10.02 minutos, lo que representa una mejora en el tiempo de venta del 37.2 %.

#### **4.7. Recomendación de mejora sobre los modelos basado en la teoría de colas**

Actualmente, los resultados de los modelos de simulación tienen una saturación de vehículos en el sistema, que se traduce en altos tiempos de espera en el tráfico por cada vehículo.

A partir de los resultados obtenidos y según la teoría de colas para un modelo M/M/K, podemos analizar y verificar los resultados obtenidos de los modelos de simulación y de esta manera realizar ajustes que permitan brindar un mejor servicio con tiempos de espera menores. El modelo de simulación planteado se cataloga bajo la teoría de colas en un modelo M/M/K debido a que:

- Los tiempos de llegada siguiendo una distribución exponencial
- Los tiempo de servicio siguen una distribución exponencial
- El número de servidores es un entero positivo mayor a 1

Para evidenciar que los modelos se encuentren o no saturados, se tomará como base el cálculo de la métrica del factor de utilización, el cual si es menor a 1 indica que el sistema es capaz de atender la carga según la tasa de llegada.

#### **4.7.1. Modelo del proceso actual**

A continuación, se presenta el procedimiento realizado para el cálculo del factor de utilización para el modelo del proceso actual y las variables de entrada utilizadas.

La fórmula para calcular el factor de utilización ( $\rho$ ) en un sistema de colas con tasa de llegada  $\lambda$  y tasa de servicio  $\mu$  es:

$$\rho = \frac{\lambda}{(\mu * C)}$$

Donde:

- $\lambda$  es la tasa de llegada de vehículos al sistema (vehículos por hora).
- $\mu$  es la tasa media de servicio de los vendedores (vehículos por hora).

- C es el número de vendedores (en este caso, 4 vendedores).

Inicialmente el 55 % de los vehículos que ingresan al sistema no compran un marbete, por lo que es necesario ajustar la tasa de llegadas:

$$\lambda \text{ ajustada} = 88 \text{ vehículos por hora} * 0.45 \approx 39.6 \text{ vehículos por hora}$$

Dado que la tasa de servicio sigue una distribución uniforme con valores máximos y mínimos, para calcular el factor de utilización se calculará la tasa media de servicio:

$$\mu = \frac{1}{E[T]} \text{ Donde } E[T] \text{ es el tiempo medio de servicio por vehículo}$$

$$E[T] = \frac{(7.5 + 25)}{2} = 16.25 \text{ minutos por vehículo}$$

$$E[T] = \frac{16.25 \text{ minutos}}{60 \text{ minutos (1 hora)}} \approx 0.2708 \text{ horas por vehículo}$$

$$\mu = \frac{1}{E[T]} \approx \frac{1}{0.2708} \approx 3.687 \text{ vehículos por hora}$$

A partir de calcular la tasa de servicio podemos realizar el cálculo para el factor de utilización:

$$\lambda = 39.6 \text{ vehículos por hora}$$

$$\mu = 3.687 \text{ vehículos por hora}$$

$$C = 4 \text{ vendedores}$$

$$\rho = \frac{\lambda}{(\mu * C)}$$

$$\rho = \frac{39.6}{(3.687 * 4)} \approx 2.138$$

El factor de utilización es aproximadamente 2.138, al ser mayor a 1 representa que el sistema se encuentra saturado, esto corrobora los resultados obtenidos por el modelo de simulación.

#### 4.7.2. Modelo de la solución propuesta

A partir de los cálculos realizados para el modelo actual, únicamente realizamos el análisis de la tasa de servicio, la cual cambia para este modelo:

$$\mu = \frac{1}{E[T]} \text{ Donde } E[T] \text{ es el tiempo medio de servicio por vehículo}$$

$$E[T] = \frac{(5 + 55)}{2} = 10 \text{ minutos por vehículo}$$

$$E[T] = \frac{10 \text{ minutos}}{60 \text{ minutos (1 hora)}} \approx \frac{1}{6} \text{ horas por vehículo}$$

$$\mu = \frac{1}{E[T]} \approx \frac{1}{\frac{1}{6}} \approx 6 \text{ vehículos por hora}$$

A partir de calcular la tasa de servicio ya podemos realizar el cálculo para el factor de utilización:

$$\lambda = 39.6 \text{ vehículos por hora (calculado anteriormente)}$$

$$\mu = 6 \text{ vehículos por hora}$$

$$C = 4 \text{ vendedores}$$

$$\rho = \frac{\lambda}{(\mu * C)}$$

$$\rho = \frac{39.6}{(6 * 4)} \approx 1.65$$

El factor de utilización es aproximadamente 1.65, al ser mayor a 1 representa que el sistema se encuentra saturado, esto corrobora los resultados obtenidos por el modelo de simulación.

#### 4.7.3. Recomendación de vendedores para el modelo actual

Para que el modelo no se encuentre saturado, el factor de utilización debe ser 1 o menor, en este caso se va a calcular cuántos vendedores son necesarios para que el factor de utilización sea uno, basándonos en la fórmula utilizada en las secciones anteriores.

$$\rho = \frac{\lambda}{(\mu * C)}$$

Se realiza el despeje de C y se asume que el factor de utilización ( $\rho$ ) es 1:

$$C = \frac{\lambda}{\mu}$$

Se toman los valores calculados anteriormente para el modelo actual:

$$\lambda = 39.6 \text{ vehículos por hora}$$

$$\mu = 3.687 \text{ vehículos por hora}$$

$$C = \frac{39.6}{3.687} \approx 10.74 = 11 \text{ vendedores}$$

Dado que la cantidad de vendedores debe ser un número entero, 10.74 se aproxima a 11 vendedores con los cuales el sistema ya no se encontraría saturado y pueda manejar la tasa de llegada de vehículos adecuadamente, por lo que es necesario contratar 7 vendedores más.

#### **4.7.4. Recomendación de vendedores para el modelo de la solución propuesta**

A partir del despeje de la fórmula de la sección anterior y con los valores calculados para el modelo propuesto:

$$C = \frac{\lambda}{\mu}$$

$$\lambda = 39.6 \text{ vehículos por hora}$$

$$\mu = 6 \text{ vehículos por hora}$$

$$C = \frac{39.6}{6} \approx 6.6 = 7 \text{ vendedores}$$

Dado que la cantidad de vendedores debe ser un número entero, 6.6 se aproxima a 7 vendedores con los cuales el sistema ya no se encontraría saturado y pueda manejar la tasa de llegada de vehículos adecuadamente, por lo que es necesario contratar 3 vendedores más.

#### **4.8. Resultados del modelo de simulación del proceso actual según recomendación de vendedores**

A partir de las recomendaciones realizadas, se agregaron 7 vendedores más al modelo del proceso actual y se ejecutó dando los resultados definidos en la siguiente tabla.

**Tabla 5.***Resultados del proceso actual según recomendación*

<b>Resultado</b>	<b>Valor</b>
Cantidad de vehículos que ingresan a la ciudad	465
Tiempo promedio en tráfico en horas	0.20 (12 minutos)
Promedio de ventas vendedor 1	24
Tiempo de venta vendedor 1 (horas)	0.3 (18 minutos)
Promedio de ventas vendedor 2	15
Tiempo de venta vendedor 2 (horas)	0.3 (18 minutos)
Promedio de ventas vendedor 3	16
Tiempo de venta vendedor 3 (horas)	0.26 (15.6 minutos)
Promedio de ventas vendedor 4	26
Tiempo de venta vendedor 4 (horas)	0.26 (15.6 minutos)
Promedio de ventas vendedor 5	12
Tiempo de venta vendedor 5 (horas)	0.32 (19.2 minutos)
Promedio de ventas vendedor 6	21
Tiempo de venta vendedor 6 (horas)	0.28 (16.8 minutos)
Promedio de ventas vendedor 7	25
Tiempo de venta vendedor 7 (horas)	0.28 (16.8 minutos)
Promedio de ventas vendedor 8	13
Tiempo de venta vendedor 8 (horas)	0.28 (16.8 minutos)
Promedio de ventas vendedor 9	13
Tiempo de venta vendedor 9 (horas)	0.31 (18.6 minutos)
Promedio de ventas vendedor 10	24
Tiempo de venta vendedor 10 (horas)	0.26 (15.6 minutos)
Promedio de ventas vendedor 11	13
Tiempo de venta vendedor 11 (horas)	0.24 (14.4 minutos)
Total de marbetes vendidos	139
Tiempo promedio de venta (horas)	0.29 (17.1 minutos)

*Nota.* Detalle de resultados de modelo de simulación del proceso actual según recomendación.

Elaboración propia, realizado con Microsoft Word.

#### 4.9. Resultados del modelo de simulación de la solución propuesta con la recomendación de vendedores

A partir de las recomendaciones realizadas, se agregaron 3 vendedores más al modelo con la solución propuesta y se ejecutó dando los resultados definidos en la siguiente tabla.

**Tabla 6.**

*Resultados de la solución propuesta según recomendación*

<b>Resultado</b>	<b>Valor</b>
Cantidad de vehículos que ingresan a la ciudad	477
Tiempo promedio en tráfico en horas	0.11 (6.6 minutos)
Promedio de ventas vendedor 1	23
Tiempo de venta vendedor 1 (horas)	0.16 (9.6 minutos)
Promedio de ventas vendedor 2	25
Tiempo de venta vendedor 2 (horas)	0.16 (9.6 minutos)
Promedio de ventas vendedor 3	37
Tiempo de venta vendedor 3 (horas)	0.15 (9 minutos)
Promedio de ventas vendedor 4	40
Tiempo de venta vendedor 4 (horas)	0.16 (9.6 minutos)
Promedio de ventas vendedor 5	29
Tiempo de venta vendedor 5 (horas)	0.18 (10.8 minutos)
Promedio de ventas vendedor 6	35
Tiempo de venta vendedor 6 (horas)	0.16 (9.6 minutos)
Promedio de ventas vendedor 7	28
Tiempo de venta vendedor 7 (horas)	0.18 (10.8 minutos)
Total de marbetes vendidos	217
Tiempo promedio de venta (horas)	0.16 (9.86 minutos)

*Nota.* Detalle de resultados de modelo de simulación de la solución propuesta según recomendación. Elaboración propia, realizado con Microsoft Word.

#### **4.10. Análisis comparativo entre los resultados iniciales y según recomendación de vendedores**

Para poder tener un mejor panorama de las mejoras entre los modelos propuestos y sus resultados es necesario comparar los resultados de los modelos iniciales planteados y los modelos generados a partir de las recomendaciones basadas en la teoría de colas.

##### **4.10.1. Modelo del proceso actual contra modelo del proceso actual con recomendación de vendedores**

A partir de los datos obtenidos en los modelos de simulación, se realizó un cuadro comparativo entre los resultados, el cual se define en la siguiente tabla.

**Tabla 7.**

*Comparación resultados proceso actual y según recomendación*

<b>Resultado</b>	<b>Modelo Original</b>	<b>Modelo Propuesto</b>	<b>Mejora</b>
Tiempo promedio de un vehículo en el tráfico en minutos	114	12	-89.47 %
Tiempo promedio de venta en minutos	15.96	17.14	7.39 %

*Nota.* Comparación entre los resultados de los modelos de simulación del proceso actual y el proceso actual según recomendación. Elaboración propia, realizado con Microsoft Word.

A partir de este cuadro comparativo, podemos definir que:

- Luego de agregar 7 vendedores más el tiempo que los vehículos pasan en el tráfico se reduce significativamente, pasando de 114 minutos a 12 minutos, lo que equivale a una reducción del 89.47 %.

- El tiempo promedio de venta se incrementó de 15.96 a 17.14 minutos, lo que equivale a un 7.39 % más de tiempo que un vendedor se tarda en finalizar la venta de un marbete.

#### 4.10.2. Modelo de la solución propuesta contra modelo de la solución propuesta con recomendación de vendedores

A partir de los datos obtenidos en los modelos de simulación se realizó el siguiente cuadro comparativo entre los resultados:

**Tabla 8.**

*Resultados solución propuesta y según recomendación*

<b>Resultado</b>	<b>Modelo Original</b>	<b>Modelo Propuesto</b>	<b>Mejora</b>
Tiempo promedio de un vehículo en el tráfico en minutos	14.4	6.6	-54.17 %
Tiempo promedio de venta en minutos	10.02	9.86	-1.6 %

*Nota.* Comparación entre los resultados de los modelos de simulación de la solución propuesta y solución propuesta según recomendación. Elaboración propia, realizado con Microsoft Word.

A partir de este cuadro comparativo, podemos definir que:

- Luego de agregar 3 vendedores más el tiempo que los vehículos pasan en el tráfico se reduce de 14.4 a 6.6 minutos, lo que equivale a una reducción del 54.17 %.
- El tiempo promedio de venta se redujo de 10.02 a 9.86 minutos, lo que equivale a un 1.6 % menos de tiempo que un vendedor se tarda en finalizar la venta de un marbete.

#### 4.10.3. Modelo de proceso actual contra modelo de la solución propuesta con recomendación de vendedores

A partir de los datos obtenidos en los modelos de simulación se realizó el siguiente cuadro comparativo entre los resultados:

**Tabla 9.**

*Resultados proceso actual y propuesta según recomendación*

<b>Resultado</b>	<b>Modelo Original</b>	<b>Modelo Propuesto</b>	<b>Mejora</b>
Tiempo promedio de un vehículo en el tráfico en minutos	114	6.6	-94.21 %
Tiempo promedio de venta en minutos	15.96	9.86	-38.22 %

*Nota.* Comparación entre los resultados de los modelos de simulación del proceso actual y solución propuesta según recomendación. Elaboración propia, realizado con Microsoft Word.

A partir de este cuadro comparativo, podemos definir que:

- Al implementar la solución propuesta con más vendedores (7) se logra una reducción del 94.21 % en el tiempo que un vehículo pasa en el tráfico.
- Al implementar la solución propuesta para la venta del marbete respecto al proceso que se lleva a cabo actualmente, se reduce un 38.22 % el tiempo que un vendedor se tarda en finalizar con una venta.



## CONCLUSIONES

1. El principal uso de la plataforma web es agilizar la compra del marbete por parte de los visitantes y de esta forma ingresar al sitio turístico con mayor facilidad.
2. A través de la aplicación web los vendedores pueden vender y verificar un marbete de forma rápida, contribuyendo a tener un mayor control sobre la venta de marbetes.
3. A través de los modelos de simulación se evidencio una mejora del 94.21 % en el proceso de venta de marbetes y una disminución del 38.22 % en el tiempo de espera en el tráfico por parte de los visitantes.



## RECOMENDACIONES

1. El desarrollo de software para automatizar procesos permite reducir los recursos invertidos y contribuir con una mejor calidad, ya sea en un servicio o producto.
2. Es importante implementar modelos de simulación que representen el comportamiento del sistema en la realidad y ayuden a validar el estado actual de los procesos con el fin de realizar ajustes que permitan mejorar los procesos y en consecuencia el sistema en general.
3. El escenario ideal para la venta de marbetes es la implementación de la aplicación móvil con un total de 7 vendedores, para este caso de estudio, con esto el tiempo que un vehículo pasa en el tráfico se reduce un 94.21 % además de reducir un 38.22 % el tiempo que se toma una venta.



## REFERENCIAS

- González, D. (2021). *Flutter, el SDK de Google para desarrollar apps multiplataforma con rendimiento nativo*. <https://profile.es/blog/que-es-flutter-sdk/>
- Limón, R. (2018). *Las 10 ramas tecnológicas con más ofertas de trabajo*. [https://elpais.com/tecnologia/2018/11/06/actualidad/1541513949\\_902481.html](https://elpais.com/tecnologia/2018/11/06/actualidad/1541513949_902481.html)
- Lucas, J. (2023, 14 abril). *Qué es NodeJS y para qué sirve*. <https://openwebinars.net/blog/que-es-nodejs/>
- Pérez, J. (2022, 3 enero). *Reglamento de estacionamiento en Antigua Guatemala: Esto deberás pagar por el parqueo*. <https://www.guatemala.com/noticias/sociedad/reglamento-estacionamiento-antigua-guatemala-esto-deberas-pagar-parqueo.html>
- Rawat, A. (2021, 13 diciembre). *Text Recognition with ML-Kit | Flutter - FlutterDevs*. <https://medium.flutterdevs.com/text-recognition-with-ml-kit-flutter-c71f27089437>

Sicán, J. (2020, 5 marzo). *Ahora para entrar a Antigua Guatemala se deberá pagar el doble, según disposición municipal.*

<https://www.prensalibre.com/ciudades/sacatepequez/ahora-para-entrar-a-antigua-guatemala-se-debera-pagar-el-doble-segun-disposicion-municipal/>