

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**

**FACULTAD DE CIENCIAS E INGENIERÍA**



**DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB Y  
MÓVIL DESPLEGADA EN LA NUBE PARA LA OPTIMIZACIÓN DE  
PROCESOS ADMINISTRATIVOS EN UNA ESCUELA DE MANEJO**

**Tesis para obtener el título profesional de Ingeniero de las Telecomunicaciones**

**AUTOR:**

Alonso Dabor Rosales Antunez

**ASESOR:**

César Stuardo Lucho Romero


Lima, Septiembre, 2024

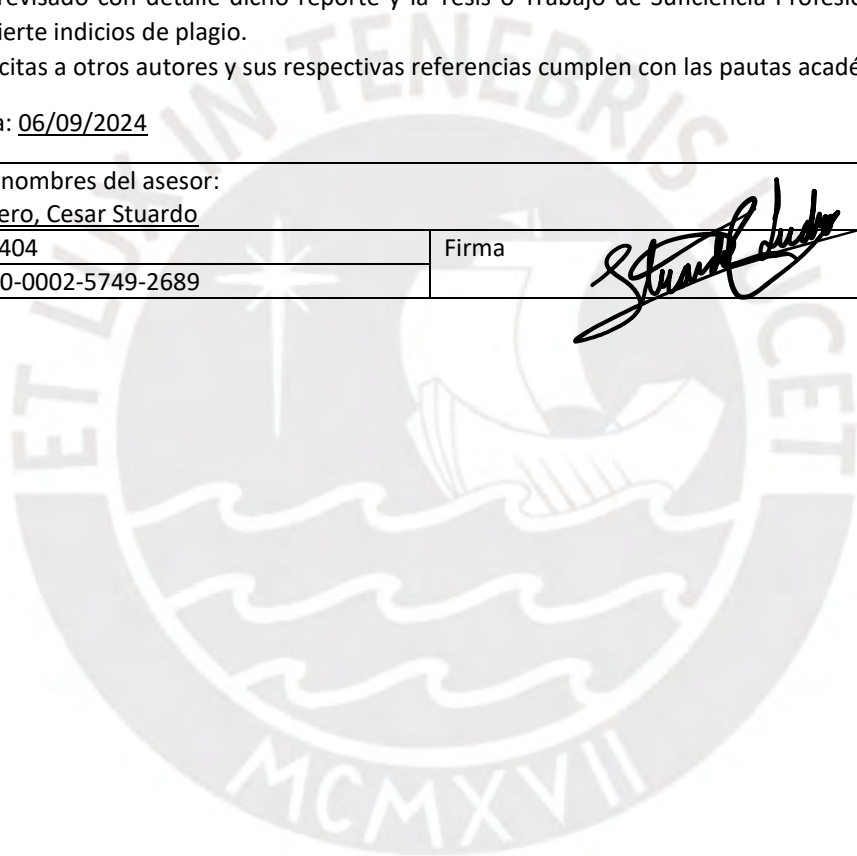
## Informe de Similitud

Yo, Cesar Stuardo Lucho Romero, docente de la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú, asesor de la tesis Diseño e implementación de una aplicación web y móvil desplegada en la nube para la optimización de procesos administrativos en una escuela de manejo, del autor Alonso Dabor Rosales Antunez, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 14%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 06/09/2024.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

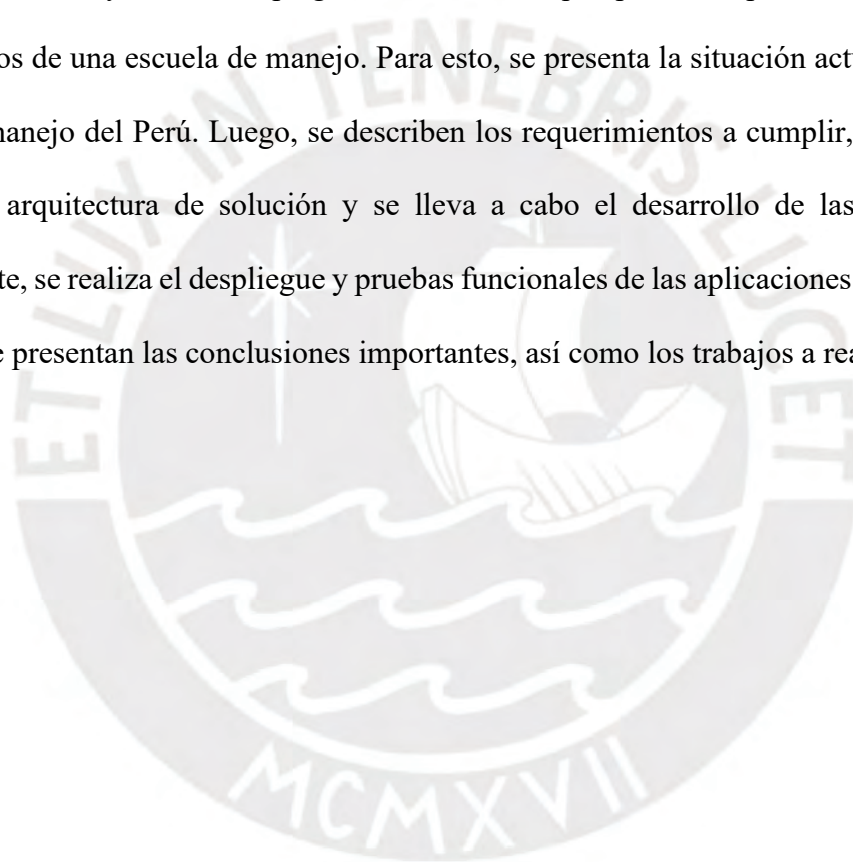
Lugar y fecha: 06/09/2024

Apellidos y nombres del asesor: <u>Lucho Romero, Cesar Stuardo</u>	
DNI: 70326404	Firma 
ORCID: 0000-0002-5749-2689	



## RESUMEN

El uso de métodos manuales por parte de las escuelas de manejo a nivel nacional en sus procesos administrativos genera inconvenientes como limitaciones de información, tiempo y errores humanos, afectando las operaciones de las escuelas. Por otro lado, la integración de sistemas tecnológicos, como aplicaciones web y móviles, permiten agilizar los procesos de negocio de una empresa. Es por eso que, el objetivo central de la tesis es el diseño e implementación de una aplicación web y móvil desplegada en la nube que permita optimizar los procesos administrativos de una escuela de manejo. Para esto, se presenta la situación actual de algunas escuelas de manejo del Perú. Luego, se describen los requerimientos a cumplir, se propone el diseño de la arquitectura de solución y se lleva a cabo el desarrollo de las aplicaciones. Posteriormente, se realiza el despliegue y pruebas funcionales de las aplicaciones desarrolladas. Por último, se presentan las conclusiones importantes, así como los trabajos a realizar a futuro.



## DEDICATORIA

A mis queridos padres, Ana y Amilcar, quienes siempre estuvieron brindándome su apoyo a lo largo de este trayecto académico.

A mi fiel compañero de todas las amanecidas, mi querida mascota Hachi, cuyo amor y travesuras fueron mi fuente de alegría en los momentos más desafiantes.

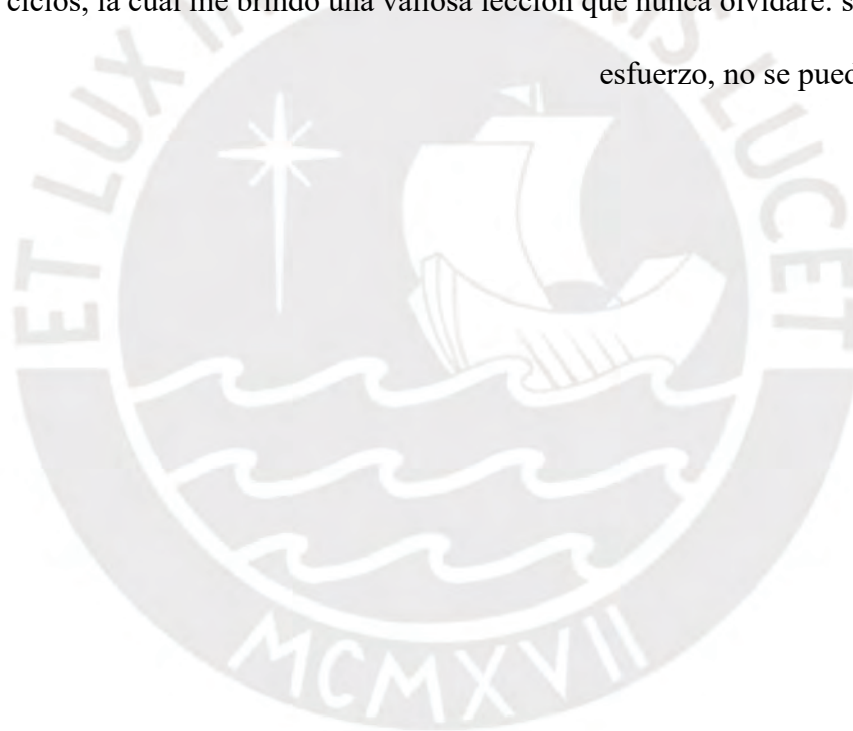


## AGRADECIMIENTOS

Agradezco a mi asesor, Stuardo Lucho, quien despertó mi interés en el área en la que está enfocada mi tesis, y por su apoyo constante desde el primer momento en que le propuse el tema de tesis.

Agradezco al profesor Pastor Chávez, cuya presentación en el simulacro de admisión del 2018, donde explicó en detalle la carrera de Telecomunicaciones, despertó mi interés y pasión por esta, impulsándome a estudiarla.

Agradezco al profesor Manuel Yarlequé por la exigencia solicitada en sus cursos en estos últimos ciclos, la cual me brindó una valiosa lección que nunca olvidaré: sin disciplina y esfuerzo, no se puede lograr nada.



## ÍNDICE GENERAL

RESUMEN.....	i
DEDICATORIA.....	ii
AGRADECIMIENTOS .....	iii
ÍNDICE GENERAL.....	iv
ÍNDICE DE TABLAS .....	vii
ÍNDICE DE FIGURAS.....	viii
INTRODUCCIÓN .....	1
Capítulo 1. Contextualización del problema y requerimientos de solución.....	3
1.1 Área de especialización en las telecomunicaciones .....	3
1.2 Situación Actual .....	4
1.3 Formulación del problema .....	8
1.4 Requerimientos de la propuesta de solución .....	9
1.5 Objetivos de la tesis.....	10
1.5.1 Objetivo general .....	10
1.5.2 Objetivos específicos.....	10
1.6 Motivación personal .....	11
1.7 Importancia del problema para las telecomunicaciones.....	12
Capítulo 2. Marco teórico y estado del arte de procesos administrativos en una escuela de manejo .....	14
2.1 Transformación Digital .....	14
2.2 Escuelas de Manejo .....	16
2.3 Aplicaciones Web .....	18
2.3.1 Modelos de Arquitecturas de Aplicaciones Web .....	18
2.3.2 Patrón de Diseño Modelo-Vista-Controlador (MVC).....	20

2.3.3 Herramientas Tecnológicas para el desarrollo de Aplicaciones Web .....	21
2.4 Aplicaciones Móviles.....	23
2.4.1 Tipos de Aplicaciones Móviles .....	23
2.4.2 Herramientas Tecnológicas para el desarrollo de Aplicaciones Móviles .....	24
2.5 Bases de Datos .....	25
2.5.1 Bases de Datos Relacionales .....	26
2.5.2 Bases de Datos No Relacionales .....	26
2.6 Estado del arte .....	27
2.6.1 Síntesis de las implementaciones descritas .....	39
Capítulo 3. Diseño e implementación de la propuesta de solución.....	30
3.1 Requerimientos de la solución .....	30
3.2 Arquitectura de la solución .....	32
3.2.1 Azure DNS .....	34
3.2.2 Azure Virtual Machines .....	35
3.2.3 Azure Blob Storage .....	37
3.2.4 Azure Database for MySQL.....	39
3.2.5 Stripe .....	43
3.2.6 Firebase Cloud Messaging .....	44
3.3 Implementación de la solución.....	46
3.4 Interfaces de las aplicaciones tecnológicas .....	55
Capítulo 4. Despliegue, pruebas y análisis económico .....	63
4.1 Arquitectura de Despliegue.....	63
4.2 Pruebas Iniciales.....	68
4.3 Optimización implementada .....	69
4.4 Arquitectura Final de Despliegue.....	70

4.5 Pruebas Funcionales de las Aplicaciones.....	72
4.6 Análisis Económico.....	77
4.7 Análisis Ambiental y de Sostenibilidad .....	84
4.8 Análisis relacionado a la Salud Pública y Seguridad .....	84
4.9 Análisis relacionado al Bienestar y el Orden Social .....	84
CONCLUSIONES .....	85
TRABAJO A FUTURO .....	86
BIBLIOGRAFÍA.....	87
ANEXOS.....	92





## ÍNDICE DE TABLAS

Tabla 3.1. Análisis de la información a ser almacenada .....	40
Tabla 3.2. Funcionalidades obtenidas del diagrama entidad-relación propuesto.....	42
Tabla 4.1. Especificaciones técnicas de las Azure Virtual Machines .....	65
Tabla 4.2. Especificaciones técnicas del Azure Database for MySQL .....	66
Tabla 4.3. Costos de Mano de Obra .....	78
Tabla 4.4. Cotización de Mano de Obra.....	78
Tabla 4.5. Estimación de GB almacenados en el servicio de Azure Blob Storage al mes.....	80
Tabla 4.6. Estimación de costo mensual del servicio Azure Blob Storage .....	80
Tabla 4.7. Estimación de datos transferidos a 1 usuario .....	81
Tabla 4.8. Estimación de costo mensual del servicio Azure CDN .....	81
Tabla 4.9. Resumen de costos de tipo OPEX.....	82
Tabla 4.10. Resumen de costos de tipo CAPEX.....	82

## ÍNDICE DE FIGURAS

Figura 1.1. Uso de Métodos manuales en la Programación de Clases de Manejo en Escuelas de Manejo .....	5
Figura 1.2. Distribución de Opiniones sobre la Programación de Clases de Manejo en Escuelas de Manejo .....	6
Figura 1.3. Uso de Métodos manuales en la Gestión de Alumnos en Escuelas de Manejo .....	6
Figura 1.4. Número de Instructores por Escuela de Manejo .....	7
Figura 1.5. Número de Vehículos por Escuela de Manejo .....	7
Figura 1.6. Número de Clases de Manejo al mes por Escuela de Manejo .....	8
Figura 1.7. Número de Alumnos por Escuela de Manejo .....	8
Figura 2.1. Esquema de una Arquitectura Monolítica .....	19
Figura 2.2. Esquema de una Arquitectura de Microservicios .....	20
Figura 2.3. Patrón de Diseño MVC .....	21
Figura 2.4. Logo de Angular .....	22
Figura 2.5. Logo de Node.js .....	22
Figura 2.6. Logo de Spring Framework .....	23
Figura 2.7. Logo de Flutter .....	25
Figura 2.8. Logo de React Native .....	25
Figura 3.1. Arquitectura de la solución propuesta .....	33
Figura 3.2. Dominio registrado en Azure DNS .....	34
Figura 3.3. Configuración para redirección de tráfico a la Azure Virtual Machine por nombre de dominio .....	35
Figura 3.4. Estructura de recursos definidos en Azure Blob Storage .....	38
Figura 3.5. Diseño del diagrama entidad-relación de la base de datos .....	42
Figura 3.6. Flujo de operación de envío de mensajes de notificación de FCM .....	45

Figura 3.7. Flujo de Control de Acceso en la aplicación web.....	46
Figura 3.8. Vista de información de producto en la plataforma de Stripe .....	47
Figura 3.9. Producto creado para su venta en la Pasarela de Pagos online .....	48
Figura 3.10. Diagrama de Flujo de Proceso del Pago Integrado con Stripe.....	49
Figura 3.11. Flujo de operación de Pasarela de Pago online para reserva de clase de manejo	50
Figura 3.12. Página web que contiene el enlace de pago generado por Stripe .....	51
Figura 3.13. Flujo de Control de Acceso en la aplicación móvil .....	52
Figura 3.14. Sección de Cuentas de Servicio de un proyecto en Firebase .....	53
Figura 3.15. Sección de Redes de la Azure Database for MySQL .....	54
Figura 3.16. Configuración de la Aplicación en Node.js para la comunicación con FCM y Azure Database for MySQL.....	54
Figura 3.17. Flujo de operación para el envío de mensajes de notificación .....	55
Figura 3.18. Interfaz de “Iniciar sesión” y “Registro” de la aplicación web .....	56
Figura 3.19. Interfaz de “Programar Clase” de la aplicación web .....	57
Figura 3.20. Opciones de “Programar Horario” y “Bloquear Horario” para un administrador en la interfaz de “Programar Clase” de la aplicación web .....	58
Figura 3.21. Interfaz de “Informes de Aprendizaje” de la aplicación web .....	59
Figura 3.22. Interfaz de “Balance” de la aplicación web .....	60
Figura 3.23. Interfaz de “Calendario de Clases” de la aplicación móvil.....	61
Figura 3.24. Interfaz de “Informes de Aprendizaje” de la aplicación móvil.....	62
Figura 4.1. Despliegue de la Arquitectura de Diseño .....	64
Figura 4.2. Configuración de red de contenedores de la Azure Virtual Machine 1 .....	65
Figura 4.3. Configuración de red de contenedores de la Azure Virtual Machine 2.....	66
Figura 4.4. Creación de Contenedores para el almacenamiento de archivos en Azure Blob Storage.....	67

Figura 4.5. Dominio adquirido en GoDaddy.....	67
Figura 4.6. Configuración de los Servidores DNS de Azure en GoDaddy .....	68
Figura 4.7. Tamaño y Tiempo de descarga de los archivos estáticos del Frontend.....	68
Figura 4.8. Reducción en el tiempo de descarga de los archivos estáticos del Frontend.....	70
Figura 4.9. Arquitectura Final de Despliegue .....	71
Figura 4.10. Operación exitosa en la Página de pago en línea.....	72
Figura 4.11. Visualización de clase reservada en el calendario del instructor elegido .....	73
Figura 4.12. Registro de recargas de combustible de un vehículo .....	73
Figura 4.13. Registro de mantenimientos realizados de un vehículo.....	74
Figura 4.14. Descarga de Balance de Operaciones en archivo .xlsx.....	74
Figura 4.15. Notificación recibida cuando la aplicación móvil está en primer plano.....	75
Figura 4.16. Notificación recibida cuando la aplicación móvil está cerrada o en segundo plano .....	75
Figura 4.17. Creación exitosa de un informe de aprendizaje de un alumno .....	76
Figura 4.18. Visualización del informe de aprendizaje creado desde la Aplicación Web.....	77
Figura 4.19. Flujo de Caja Mensual por Escuela de Manejo .....	83
Figura 4.20. Flujo de Caja Anual estimado.....	84

## INTRODUCCIÓN

Actualmente, la integración de tecnología por parte de las empresas está teniendo un impacto significativo en su desarrollo. La adopción de sistemas tecnológicos, como aplicaciones web y móviles, está permitiendo mejorar su productividad, eficiencia y la experiencia de sus usuarios. Por otra parte, en el sector de educación vial a nivel nacional, aún hay escuelas de manejo que continúan confiando en métodos manuales y tradicionales para llevar a cabo sus operaciones. Por lo que, la implementación de una aplicación web y móvil podría ser un cambio transformador, permitiendo a estas escuelas agilizar sus procesos administrativos, mejorando su gestión.

Por lo tanto, la tesis tiene como objetivo principal desarrollar e implementar una aplicación web y móvil desplegada en la nube que permita optimizar los procesos administrativos de una escuela de manejo.

El desarrollo de la tesis tiene como motivación principal el de mejorar la manera en cómo algunas escuelas de manejo llevan a cabo sus procesos administrativos, por lo que, al brindarles una solución tecnológica avanzada, podrán realizar una gestión más eficiente de estos y mejorar la calidad de entrega de sus servicios a sus alumnos.

La metodología que se empleó en la tesis es la metodología general del diseño de ingeniería, enfocada en abordar problemas en el área de las telecomunicaciones, la cual se desarrolla en cuatro capítulos.

En el primer capítulo se desarrolla la situación actual de algunas escuelas de manejo, la formulación de la problemática y los objetivos de la tesis.

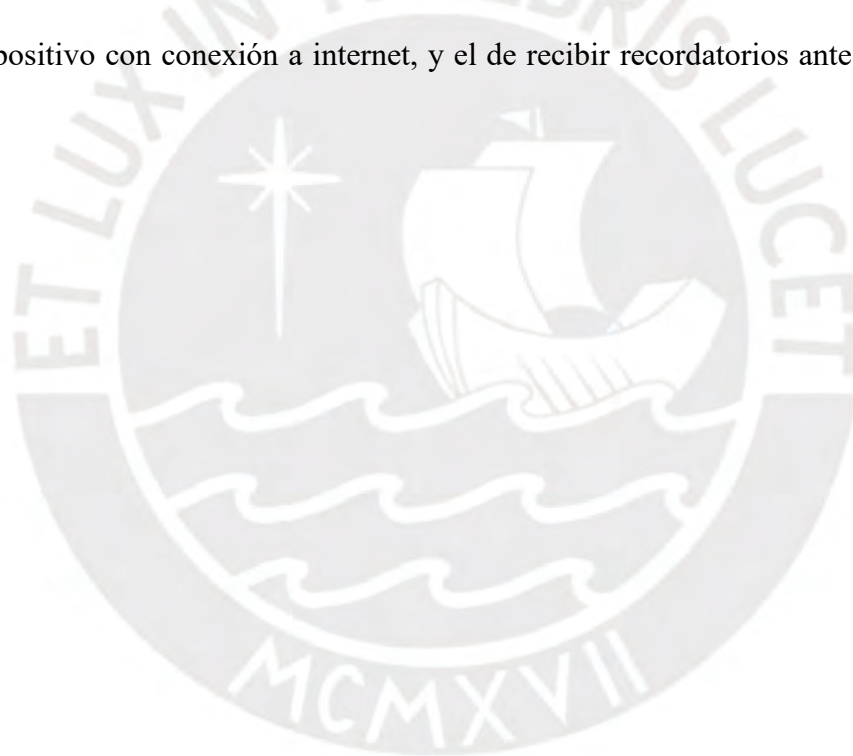
En el segundo capítulo se presenta el marco teórico exponiendo los conceptos que se utilizarán para el desarrollo de la aplicación web y móvil. Finalmente, se desarrolla el estudio del arte donde se describen las diversas implementaciones relacionadas al tema de tesis.

En el tercer capítulo se detalla los requerimientos que deben cumplir las aplicaciones a desarrollar, la arquitectura de diseño propuesta y la implementación de la propuesta de solución.

En el cuarto capítulo se presenta el despliegue de la arquitectura de diseño y las pruebas funcionales realizadas. Finalmente, se realiza un análisis económico, ambiental, de sostenibilidad, de salud pública y de seguridad, relacionado con las aplicaciones desarrolladas.

En adición, se presentan las conclusiones, trabajos a futuro, la bibliografía y los anexos de la tesis.

Finalmente, la aplicación web y móvil desarrollada tiene como logro principal facilitar la gestión de horarios de clase de manejo, permitiendo a los alumnos reservar sus clases desde cualquier dispositivo con conexión a internet, y el de recibir recordatorios antes del inicio de estas.



## **Capítulo 1. Contextualización del problema y requerimientos de solución**

El presente capítulo tiene como finalidad presentar el área de especialización en las telecomunicaciones en la que está enfocada el presente trabajo de tesis. También, se describe la situación actual y la formulación de la problemática, así como los requerimientos que implica la solución propuesta. Asimismo, se mencionan los objetivos de la tesis y la motivación personal del autor. Finalmente, se explica la importancia de la solución propuesta desde los puntos de vista social, técnico, educacional y económico para las telecomunicaciones.

### **1.1 Área de especialización en las telecomunicaciones**

El presente trabajo de tesis está enfocado en las áreas de servicios y aplicaciones para telecomunicaciones de la carrera de Ingeniería de las Telecomunicaciones. Los conocimientos aprendidos en esta área fueron adquiridos en los cursos de Ingeniería Web para Telecomunicaciones, Gestión de Servicios de TICs y Servicios y Aplicaciones para IOT. El primer curso sentó las bases del segundo para permitir el diseño y desarrollo de aplicaciones web con conexión a base de datos en un contexto de frameworks, así como su posterior despliegue en nube a través del uso de diversos servicios de los proveedores existentes. Por otro

lado, el tercer curso permitió el diseño y construcción de aplicaciones móviles bajo criterios de seguridad y eficiencia. A pesar que los cursos mencionados anteriormente están enfocados en el desarrollo de distintas plataformas tecnológicas, los tres contribuyen al desarrollo de la tercera competencia de la carrera referida al “desarrollo e implementación de servicios y aplicaciones de telecomunicaciones y sistemas de distribución de contenidos sobre Internet, web móvil y video bajo demanda que cumplan con criterios de seguridad, confiabilidad y bajo retardo”, gracias a la oportunidad de participar en diversos proyectos de servicios TICs a lo largo de la carrera.

## **1.2 Situación Actual**

Actualmente la tecnología juega un papel importante en el desarrollo de las empresas puesto que les permite brindar innovadoras soluciones y fomentar competitividad en el mercado [1] [2][3]. La implementación de sistemas informáticos, como aplicaciones web y móviles, no sólo les brinda una gestión más rápida de la información y la comunicación desde cualquier dispositivo, sino que también les permite agilizar los procesos de su negocio, la prestación de sus servicios, etc. En consecuencia, son más las empresas u organizaciones, de cualquier tipo, sector y tamaño, que han notado la importancia de adaptarse a este cambio tecnológico para poder ofrecer servicios más rápidos y de mejor calidad de entrega [1] [2][3].

Es así que, el uso de la tecnología se vuelve cada vez más indispensable en las empresas para la agilización de los procesos de su negocio, optimización de tiempo y recursos humanos, ampliación de su alcance en el mercado, etc. permitiéndoles obtener beneficios como mayor rentabilidad, aumento de productividad, etc. [1][2][3]. Por lo tanto, se evidencia una transformación digital de las organizaciones.

Cabe destacar que siguen existiendo entidades que aún no han dado el paso crucial de adaptación a este cambio, lo que podría llevar a su obsolescencia en el mercado si no se adoptan



iniciativas digitales en sus operaciones en el plazo más corto posible [2]. En el contexto de la educación vial en el Perú, las escuelas de manejo juegan un papel importante en la formación de conductores capaces y responsables. Según los resultados de una encuesta distribuida a nivel nacional entre diversas escuelas de manejo, con 16 de ellas respondiendo, se recopiló información relevante sobre la manera en cómo llevan a cabo sus procesos administrativos. Se debe destacar que, si bien se cuenta con la autorización de las escuelas para usar los datos proporcionados para la tesis, por razones de confidencialidad, los nombres de las escuelas participantes no se divulgarán.

Al analizar las respuestas de la encuesta, se identificaron aspectos para mejorar en las escuelas de manejo, como por ejemplo, en la gestión de sus clases de manejo y alumnos. Como se aprecia en la Figura 1.1, un gran porcentaje de las escuelas señaló que continúan confiando y empleando enfoques manuales y tradicionales basados en papel, una práctica que resulta tediosa y propensa a errores humanos, especialmente en relación a la programación de sus clases de manejo, como se observa en la Figura 1.2.



*Figura 1.1. Uso de Métodos manuales en la Programación de Clases de Manejo en Escuelas de Manejo*

*Fuente: Elaboración propia*



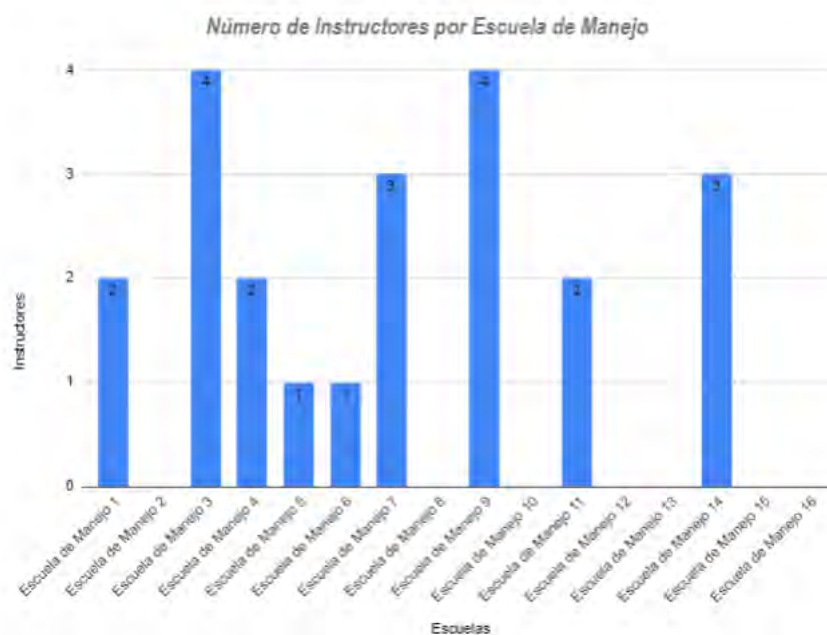
Figura 1.2. Distribución de Opiniones sobre la Programación de Clases de Manejo en Escuelas de Manejo  
Fuente: Elaboración propia

En relación a la gestión de sus alumnos, se observa en la Figura 1.3, que la mayoría de las escuelas indican el empleo de métodos manuales, destacando la ineficiencia y la pérdida de tiempo que esto conlleva, especialmente en tareas como el seguimiento de alumnos, entre otros.

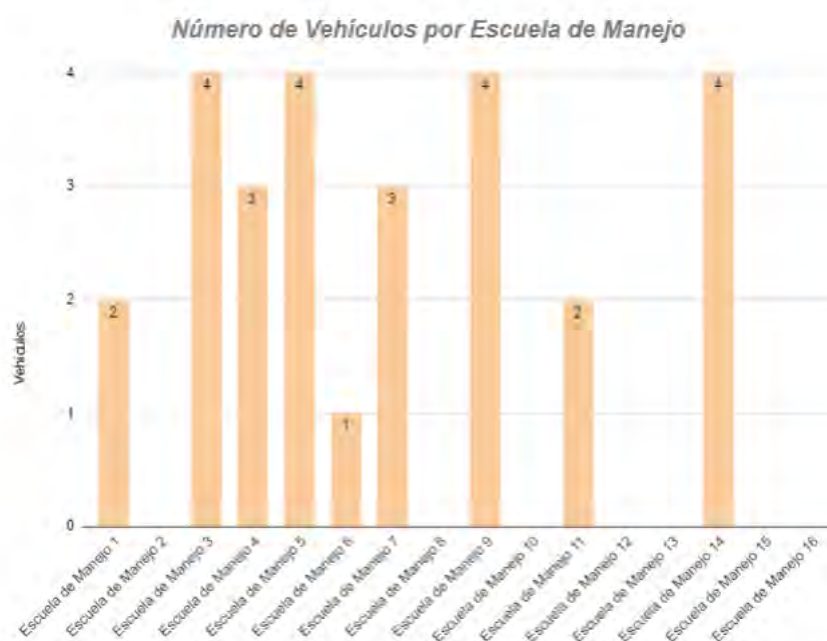


Figura 1.3. Uso de Métodos manuales en la Gestión de Alumnos en Escuelas de Manejo  
Fuente: Elaboración propia

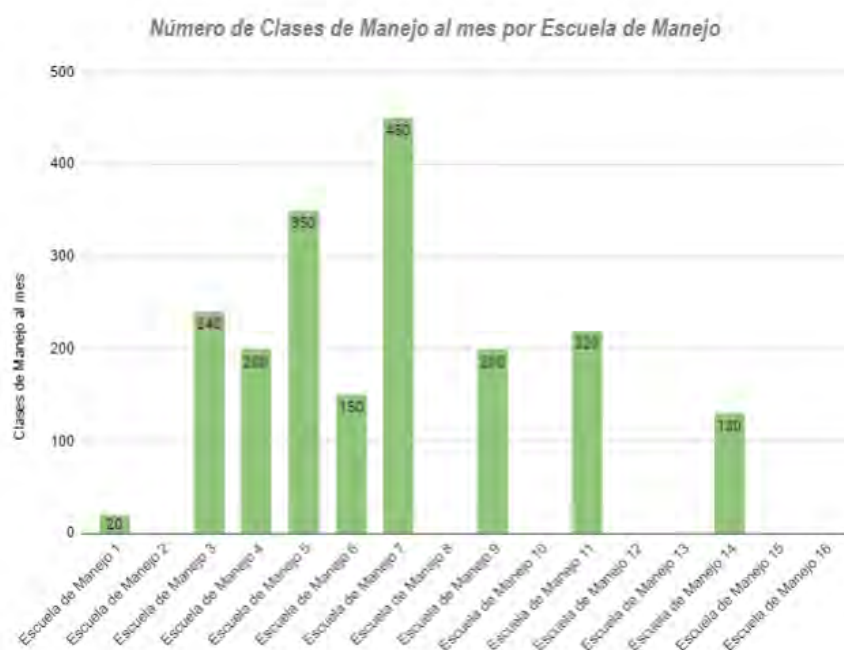
La información recopilada acerca de las escuelas, incluyendo el número de instructores, vehículos, cantidad de clases mensuales y total de alumnos, se presenta en las Figuras 1.4, 1.5, 1.6 y 1.7 respectivamente. El uso de esta información será de importancia para el Capítulo 3 y 4 de la tesis.



*Figura 1.4. Número de Instructores por Escuela de Manejo*  
*Fuente: Elaboración propia*



*Figura 1.5. Número de Vehículos por Escuela de Manejo*  
*Fuente: Elaboración propia*



*Figura 1.6. Número de Clases de Manejo al mes por Escuela de Manejo*  
Fuente: Elaboración propia



*Figura 1.7. Número de Alumnos por Escuela de Manejo*  
Fuente: Elaboración propia

### 1.3 Formulación del problema

De acuerdo a lo mencionado sobre la situación actual en algunas escuelas de manejo, se ha observado que el empleo de métodos manuales en la gestión de sus procesos administrativos,

afectan negativamente a las empresas tanto a la forma en que ofrecen sus servicios como la experiencia que proporcionan a sus clientes. Estas prácticas tradicionales resultan en demoras y limitaciones en términos de tiempo e información, aumentando además la probabilidad de errores humanos, lo que perjudica la operación de la empresa. Así, en un entorno empresarial cada vez más digitalizado, genera que estas escuelas de manejo se encuentren en desventaja, en términos de competitividad.

#### **1.4 Requerimientos de la propuesta de solución**

En función a la problemática definida anteriormente y los hallazgos del estudio preliminar realizado, se plantea el desarrollo de una solución tecnológica que permita la optimización de procesos administrativos en una escuela de manejo. Esto le permitirá obtener beneficios en cuanto a:

- **Experiencia del usuario:** Permitirá mejorar la interacción de los usuarios con el sistema, beneficiando a los clientes como al personal de la escuela de manejo.
- **Eficiencia:** Permitirá minimizar la carga de trabajo manual del personal de la escuela de manejo, generando ahorro de tiempo y recursos humanos, así como la agilización del modo de operación de la entidad.
- **Productividad:** Al optimizar el tiempo y los recursos humanos, permitirá que el personal de la escuela de manejo se enfoque en actividades de mayor valor.
- **Rentabilidad:** Al reducir costos relacionados a la gestión manual de procesos administrativos y mejorar la forma en que se ofrecen los servicios, se promoverá la retención de alumnos, lo que, en consecuencia, generará ingresos más constantes en el tiempo.

Es por ello que se plantea los siguientes requerimientos para esta solución:

- Optimización de procesos administrativos: Para reducir la carga de trabajo manual en la escuela de manejo.
- Aplicaciones web y móvil: Para facilitar el acceso a información y una interacción mucho más rápida de los usuarios con las escuelas de manejo, desde cualquier dispositivo.
- Interfaz amigable y sencilla de usar: Para brindar una mejor experiencia a los usuarios debido a la fácil interacción con las aplicaciones.
- Seguridad, privacidad y almacenamiento de información en la nube: Para la protección de la información confidencial de los usuarios, así como de asegurar la disponibilidad de la información en todo momento, previniendo una posible pérdida de esta.
- Acceso a información actualizada: Para garantizar que las aplicaciones muestren la información más reciente.

## **1.5 Objetivos de la tesis**

### **1.5.1 Objetivo general**

El presente trabajo de tesis tiene como objetivo principal el diseño e implementación de una plataforma o aplicación web y móvil desplegada en la nube que permita la optimización de procesos administrativos en una escuela de manejo.

### **1.5.2 Objetivos específicos**

Además del objetivo general, se tienen los siguientes objetivos específicos:

- Diseñar e implementar una aplicación web que permita a una escuela de manejo la gestión de alumnos, horario de clases de manejo, información de vehículos y la recopilación de datos relevantes para el balance de operaciones.

- Diseñar e implementar una aplicación móvil Android que complemente a la aplicación web, incluyendo funcionalidades como el envío de notificaciones como recordatorios sobre las clases de manejo programadas y la generación de informes de aprendizaje a los alumnos.
- Diseñar, crear y utilizar una base de datos en la nube para la administración de información.
- Desplegar la aplicación web en la nube.

## 1.6 Motivación personal

El presente trabajo de tesis se desarrolló para abordar el desafío de optimizar la manera en cómo algunas escuelas de manejo, hoy en día, continúan realizando la gestión de sus procesos administrativos, brindándoles una alternativa de modernización y mejora en base a la implementación de una solución tecnológica que se encargue de optimizar la gestión de estos. Desde el punto de vista del autor, el motivo personal por el cual surgió la idea de desarrollar este proyecto se debe a la experiencia que tuvo al presenciar la manera en cómo se gestionaba los procesos administrativos en una escuela de manejo. El autor pudo observar varios aspectos negativos en una escuela de manejo, al experimentar situaciones como, por ejemplo, verse en la obligación de tener que acudir presencialmente a la escuela de manejo para realizar la programación de su clase de manejo, retraso en cuanto a la selección de un horario disponible para hacer la reserva de su clase de manejo, ausencia de recordatorios sobre los horarios programados, y el más perjudicial, cambios en su programación por el cruce de horarios producto del error del personal de la escuela, entre otras situaciones adversas.

Debido a esto, nace el interés en el autor por buscar una solución tecnológica que se encargue de optimizar los procesos administrativos en una escuela de manejo, mejorando el acceso a información y los servicios brindados a los usuarios, permitiéndoles, por ejemplo, programar

sus clases de manejo desde cualquier lugar, con tan solo un dispositivo con conexión a internet y en unos cuantos simples pasos, etc.

Finalmente, el autor considera que el diseño e implementación de estas plataformas, podrá ser utilizado por otras escuelas de manejo que se encuentren en situaciones similares; por lo que, gracias a su aplicación, se espera que las escuelas de manejo puedan mejorar su modo de operación al contar con una solución rápida, confiable y de acceso instantáneo a información requerida.

## **1.7 Importancia del problema para las telecomunicaciones**

### **1.7.1 Importancia social**

Desde el punto de vista social, la implementación de la solución tecnológica propuesta, generará un impacto positivo en las escuelas de manejo, proporcionando una modernización y mejora a la forma en que estas organizaciones ofrecen sus servicios. Esto se traducirá en una mayor facilidad y conveniencia para los alumnos al acceder y programar sus clases de manejo, lo que contribuirá a una experiencia más satisfactoria y eficiente en la forma en que se les brindan los servicios. De esta manera, se fomenta una cultura de innovación y adaptación tecnológica en la sociedad.

### **1.7.2 Importancia técnica**

La importancia técnica de la solución propuesta radica en que permite hacer uso de las herramientas tecnológicas de hoy en día, sacándoles provecho para brindar una alternativa de solución a la problemática que experimentan algunas escuelas de manejo.

Asimismo, la importancia técnica de la solución se relaciona con la optimización de procesos administrativos en escuelas de manejo ya que la integración de sistemas tecnológicos contribuirá a reducir la probabilidad de cometerse errores humanos que se puedan presentar en



el camino, así como la obtención de una mayor eficiencia de estos. En consecuencia, se tendrá una mayor confiabilidad al nuevo modo de operación de las escuelas de manejo.

### **1.7.3 Importancia educativa**

A través de la implementación de la solución tecnológica propuesta, se logrará enriquecer la experiencia educativa de los alumnos ya que permitirá a los instructores poder generar informes de aprendizaje de sus alumnos. Estos permitirán registrar las habilidades adquiridas de los alumnos en cada lección de manejo, así como los aspectos que se deben reforzar, brindándoles una retroalimentación sobre su desempeño. De este modo, permitirá a los instructores realizar los ajustes necesarios a sus planes de lecciones, brindando clases más personalizadas a sus alumnos en función a su progreso de aprendizaje.

### **1.7.4 Importancia económica**

La implementación de la solución tecnológica propuesta en el ecosistema de una escuela de manejo podría generar un ahorro económico, al reducir los gastos relacionados con los recursos humanos en la escuela. Al optimizar procesos administrativos, existe la posibilidad de prever una menor dependencia de la mano de obra para la gestión de sus procesos, generando así un impacto económico positivo para la institución. Asimismo, también podría darse una reasignación del personal existente a otras actividades de valor que generen ingresos adicionales a la escuela de manejo.

### **1.7.5 Importancia legal, política y ética**

Debido al enfoque de la tesis, no se identifica una importancia significativa en el aspecto legal, político o ético.

## **Capítulo 2. Marco teórico y estado del arte de procesos administrativos en una escuela de manejo**

El presente capítulo tiene como objetivo desarrollar los conceptos teóricos que aborda el presente trabajo de tesis y aquellos que se utilizarán para el desarrollo de la aplicación web y móvil. Finalmente, en el estado del arte se describirán algunas de las implementaciones que existen actualmente relacionadas al tema del presente trabajo de tesis, culminando con una síntesis de las implementaciones analizadas.

### **2.1 Transformación Digital**

Para comprender adecuadamente la transformación digital, es importante diferenciarla de la digitalización, términos que a menudo se confunden. La digitalización se refiere a la conversión de componentes o procedimientos físicos a un formato digital, lo cual no implica necesariamente cambios significativos en la operación de las empresas u organizaciones. Aunque es un paso crucial, la transformación digital va mucho más allá, abarcando la adopción de nuevas tecnologías en todas las áreas de una empresa u organización. Esta conlleva a

cambios más amplios y profundos del modo de operar y brindar servicios de las empresas, mejorando significativamente el valor de los servicios que se ofrecen a los usuarios, fomentando una mayor presencia y competitividad en el mercado [2][3].

La transformación digital implica un cambio fundamental en la cultura organizacional, la estructura y los procesos operativos de una empresa, buscando una renovación completa de todas las áreas de una empresa para maximizar su impacto. Por lo tanto, más que una simple integración de tecnologías en el negocio empresarial, representa una transformación profunda en la cultura empresarial [2].

A continuación, se describirán algunos de los beneficios que brinda la transformación digital en las empresas [2][3]:

- Mejora de la productividad: Permite mejorar la eficiencia de los procesos de la empresa, ahorrando tiempo y recursos. En consecuencia, permite que los trabajadores puedan dedicar más tiempo a actividades con mayor valor.
- Reducción de costos: Permitirá reducir los gastos operativos mediante la optimización de procesos y la adopción de tecnologías como “Cloud Computing”, resultando en ahorros en áreas como mantenimiento, logística y recursos humanos.
- Mejora la experiencia del usuario: Permite una interacción más enriquecedora para el cliente con el sistema, lo que resulta en la prestación de servicios de mayor calidad y satisfacción.
- Mejora la rentabilidad: Permite captar una mayor cantidad de usuarios gracias a contar con un mayor alcance y mejor calidad de servicios, lo que consigue la retención de clientes.
- Fomenta una cultura de innovación: Impulsa a las empresas a adoptar soluciones tecnológicas avanzadas, como dispositivos IoT, Machine Learning, Inteligencia Artificial, etc. adaptándose a las necesidades en evolución de los clientes.

## 2.2 Escuelas de Manejo

Las escuelas de manejo o conducción son las entidades facultadas por el MTC para la formación y capacitación teórica-práctica en la conducción de vehículos de transporte terrestre a las personas que buscan la ansiosa obtención de la licencia o permiso de conducción [4][5]. En la formación teórica, las escuelas de manejo se encargan de enseñar las leyes, normas y reglamentos de tránsito a los estudiantes para su concientización, promoviendo una conducción más segura y responsable. Por otro lado, la capacitación práctica implica aplicar dichos conocimientos aprendidos en diversas situaciones de la vida real, con el fin de mejorar su habilidad conductiva.

Los principales actores que se encuentran en una escuela de manejo incluyen [5]:

- **Alumnos:** Personas que buscan formarse y desarrollar sus habilidades para la conducción de vehículos de transporte terrestre.
- **Instructores:** Profesionales autorizados por el MTC, encargados de impartir la enseñanza teórica y práctica, así como de desarrollar las habilidades de conducción en los alumnos.
- **Personal Administrativo:** Persona encargada de las operaciones y gestiones administrativas esenciales para el funcionamiento de la escuela de manejo.

En relación a los procesos administrativos de una escuela de manejo, estos se refieren a las diversas actividades destinadas a garantizar su funcionamiento eficiente y efectivo. Estos procesos, que varían según la escuela, involucran principalmente tareas como la organización, planificación y gestión de recursos fundamentales para la operación de la escuela de manejo. De acuerdo a la encuesta realizada, como se mencionó en la Sección 1.2, se recopiló información adicional sobre los procesos administrativos de las escuelas de manejo encuestadas. A continuación, se detallan estos procesos administrativos:

1. **Gestión de Usuarios:** Abarca el proceso de matrícula, que implica el registro de información personal de los alumnos. También involucra el seguimiento de los alumnos, asegurando un monitoreo efectivo de su aprendizaje en sus clases de manejo. Adicionalmente, incluye la gestión de la comunicación entre los alumnos y la escuela de manejo a través de los canales establecidos para tal fin.
2. **Gestión de Horarios de Clases de Manejo:** Implica la coordinación y programación de los horarios de manejo disponibles, asegurando que los alumnos tengan la capacidad de escoger aquellos que mejor se alineen con su disponibilidad. Además, incluye el proceso de pago, necesario para confirmar y asegurar la reserva del horario elegido. Asimismo, abarca la asignación adecuada de instructores y verificación de la disponibilidad de vehículos, lo que es fundamental para evitar posibles cruces de horarios.
3. **Gestión de Vehículos:** Se encarga del manejo de la información relacionada con la flota vehicular de la escuela de manejo. Permite mantener un registro detallado del estado de los vehículos, incluyendo los mantenimientos realizados, recargas de combustible, etc. Estas prácticas son clave para garantizar el correcto funcionamiento y disponibilidad de los vehículos para las clases de manejo.
4. **Promoción y Marketing:** Se enfoca en incrementar la visibilidad y el atractivo de la escuela de manejo, con el fin de captar la atención de potenciales alumnos. Para ello, se emplean estrategias de marketing y publicidad tanto en plataformas digitales como tradicionales, destacando los servicios y ventajas que ofrece la escuela.
5. **Operaciones y Finanzas:** Abarca la gestión de los flujos financieros de la escuela de manejo. Además, se enfoca en la elaboración de reportes que proporcionan información de las operaciones de la escuela de manejo, facilitando un seguimiento efectivo.

## 2.3 Aplicaciones Web

Una aplicación web es un software que se ejecuta en un servidor web, permitiendo a los usuarios acceder a éste desde cualquier dispositivo conectado a internet y navegador. Estas aplicaciones funcionan bajo un modelo cliente-servidor. En el lado del cliente, se encarga de la interacción con la interfaz de usuario. Por su parte, el servidor se encarga de procesar las solicitudes del cliente y devolver respuestas adecuadas a estas peticiones [6].

### 2.3.1 Modelos de Arquitecturas de Aplicaciones Web

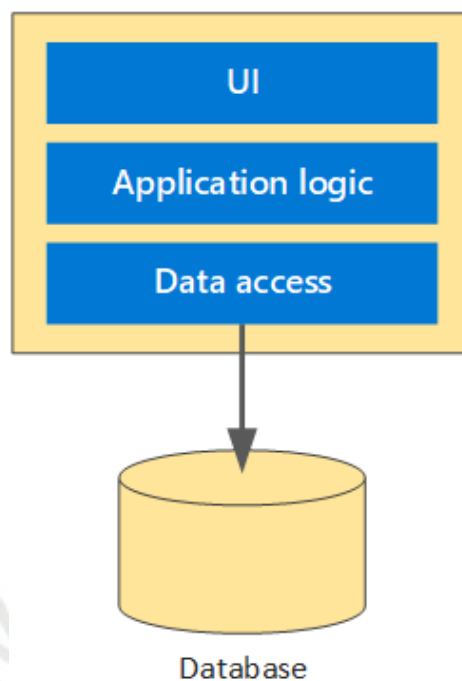
Existen varios modelos de arquitectura que pueden ser utilizados para el diseño y desarrollo de una aplicación web. Los modelos de arquitectura web de gran importancia en el desarrollo de software son:

#### 2.3.1.1 Arquitectura Monolítica

Es aquel modelo donde la aplicación web se construye como un único bloque. Esto quiere decir que todos sus componentes se encuentran en la misma unidad comunicándose entre sí a través de funciones internas, haciendo que su ejecución, compilación y despliegue se maneje de forma conjunta [7].

Una arquitectura monolítica consta de los siguientes componentes fundamentales [7][8], tal como se observa en la Figura 2.1:

- **Interfaz de Usuario:** Permite la interacción de los usuarios con la plataforma, permitiendo el acceso a sus servicios.
- **Lógica de Negocio:** Contiene la lógica de la aplicación, definiendo su comportamiento y gestionando los datos conforme a las políticas de negocio establecidas.
- **Acceso a Datos:** Responsable de la comunicación con la base de datos, recupera y procesa la información solicitada.



*Figura 2.1. Esquema de una Arquitectura Monolítica*  
*Fuente: [9]*

Aunque la arquitectura monolítica se destaca por la simplicidad de su implementación y despliegue, presenta desventajas en mantenimiento y escalabilidad. Esto se debe a que cualquier cambio en el código de la aplicación web requiere que se vuelva a desplegar por completo para que los cambios tomen efecto. Además, la complejidad del código tiende a incrementarse con la adición de nuevas funcionalidades [7][10].

### **2.3.1.2 Arquitectura de Microservicios**

Es aquel modelo que se caracteriza por dividir a la aplicación web en pequeños componentes independientes, conocidos como microservicios, cada uno encargado de una funcionalidad específica del negocio y comunicándose entre ellos mediante APIs, como se muestra en la Figura 2.2. Este esquema permite el desarrollo y operación autónoma de cada microservicio, aumentando la tolerancia a fallos y evitando afectar a los demás componentes [7][11].

Además, ofrece escalabilidad al permitir que cada microservicio se adapte individualmente según la demanda, y brinda libertad tecnológica al permitir el uso de diferentes lenguajes de programación o tecnologías para el desarrollo de cada servicio [7][11].

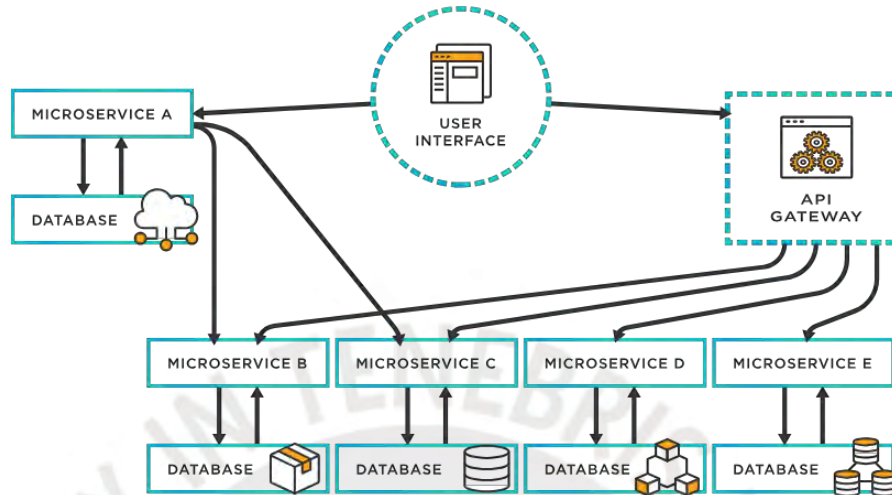


Figura 2.2. Esquema de una Arquitectura de Microservicios  
Fuente: [12]

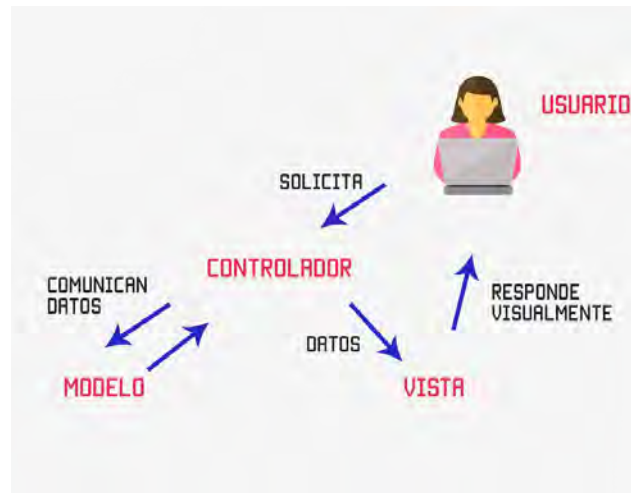
### 2.3.2 Patrón de Diseño Modelo-Vista-Controlador (MVC)

Patrón de diseño que brinda una estructura para los componentes de una aplicación web, dividiéndolas en tres capas esenciales [13][14]:

- Modelo: Esta capa gestiona el acceso a los datos de la aplicación.
- Vista: Se encarga de mostrar la información al usuario en la Interfaz de Usuario.
- Controlador: Maneja la interacción del usuario, operando en conjunto con el Modelo.

En la Figura 2.3, se muestra que las solicitudes hechas por el usuario son dirigidas al Controlador. Este último interactúa con el Modelo para ejecutar las acciones requeridas y obtener los datos solicitados. Luego, el Controlador procesa la información obtenida y los transfiere a la Vista, que se encarga de presentarlos al usuario [13][14].





*Figura 2.3. Patrón de Diseño MVC  
Fuente: [15]*

### 2.3.3 Herramientas Tecnológicas para el desarrollo de Aplicaciones Web

Actualmente existen diversas herramientas tecnológicas para el desarrollo web, tanto para el lado del servidor (backend) que se encarga de la lógica y procesamiento de datos, como para el lado del cliente (frontend), que gestiona la interfaz gráfica y la interacción con el usuario. Es por ello que, para simplificar la lista de estas tecnologías a describir, sólo se mencionará las que se empleó en la tesis.

#### 2.3.3.1 JavaScript

Lenguaje de programación clave para crear páginas web interactivas. Funciona junto con HTML y CSS, donde estos últimos forman la estructura estática y JavaScript añade interactividad. Asimismo, como lenguaje interpretado, no necesita compilación antes de ejecutarse. Dentro de sus tecnologías asociadas, Angular es un framework que permite el desarrollo de aplicaciones de una sola página en el frontend, utilizando HTML y TypeScript, el cual mejora JavaScript agregando tipos de sintaxis para una mejor detección de errores [16][17].



*Figura 2.4. Logo de Angular  
Fuente: [18]*

Por otro lado, Node.js es un entorno de ejecución eficiente para el desarrollo de aplicaciones backend con JavaScript [19]. Su arquitectura, basada en eventos y naturaleza no bloqueante, lo hacen ideal para aplicaciones que requieren un alto rendimiento y procesamiento en tiempo real [20].



*Figura 2.5. Logo de Node.js  
Fuente: [21]*

### **2.3.3.2 Java**

Lenguaje de programación orientado a objetos, que, según Microsoft Azure, es el más popular en el desarrollo de aplicaciones. Es un lenguaje multiplataforma, que al compilarse, genera un archivo de código de bytes ejecutable en cualquier plataforma que tenga la Java Virtual Machine (JVM) [22][23]. Entre sus tecnologías asociadas, Spring Framework es notable por facilitar la creación de aplicaciones Java de carácter empresarial, manejando la infraestructura de la aplicación, permitiendo que los desarrolladores solo se enfoquen en la lógica y funcionalidades [24].



*Figura 2.6. Logo de Spring Framework  
Fuente: [25]*

## **2.4 Aplicaciones Móviles**

Es un software diseñado específicamente para funcionar en dispositivos móviles, como smartphones, tablets, etc. [26].

### **2.4.1 Tipos de Aplicaciones Móviles**

Cuando se habla de tipos de aplicaciones móviles se hace referencia a cómo se construirá la aplicación. Existen diversos tipos, entre los cuales tenemos:

#### **2.4.1.1 Aplicaciones Nativas**

Las aplicaciones nativas son desarrolladas específicamente para una plataforma, ya sea iOS o Android, utilizando los lenguajes y frameworks propios de cada uno. Sus principales ventajas son que tienen acceso a todas las funcionalidades del dispositivo, como la cámara, el GPS, etc. Además, pueden operar sin conexión a internet y se ejecutan en el sistema operativo del dispositivo, incluso en segundo plano, proporcionando notificaciones al usuario cuando ocurre algún evento [26][27].

#### **2.4.1.2 Aplicaciones Web Móviles**

Las aplicaciones web móviles se ejecutan en el navegador del dispositivo y se desarrollan utilizando lenguajes web como HTML, CSS y JavaScript. Estas aplicaciones ofrecen actualizaciones inmediatas, puesto que los cambios se realizan en el servidor web que aloja a la aplicación. Además, son más económicas y rápidas de desarrollar debido a su compatibilidad

con múltiples plataformas, lo que las hace una alternativa eficiente frente a las aplicaciones nativas [26][27].

### **2.4.1.3 Aplicaciones Híbridas**

Las aplicaciones híbridas resultan de una mezcla entre las aplicaciones nativas y web, ejecutándose en contenedores web dentro de dispositivos móviles. Este tipo de aplicaciones ofrecen acceso a ciertas funcionalidades del dispositivo y facilitan el desarrollo de multiplataforma al mantener una única base de código. Aunque permite una mayor flexibilidad, su rendimiento suele ser inferior al de las aplicaciones nativas [26][27].

## **2.4.2 Herramientas Tecnológicas para el desarrollo de Aplicaciones Móviles**

La elección de un lenguaje de programación para el desarrollo móvil depende de la plataforma a la que vaya a estar enfocada dicha aplicación. A continuación, se describirán las dos tecnologías más empleadas en el desarrollo [28]:

### **2.4.2.1 Flutter**

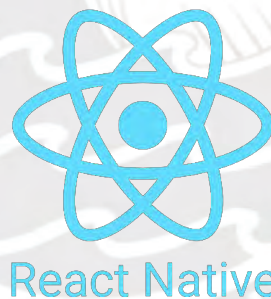
Framework desarrollado por Google que utiliza el lenguaje de programación Dart. Actualmente, es compatible con las siguientes plataformas: iOS, Android, Web, Windows, MacOS y Linux. Esta tecnología facilita a los desarrolladores la creación de interfaces de usuario mediante widgets personalizables, donde todo en la pantalla se construye con estos. Su ventaja principal radica en el uso de un motor gráfico propio para renderizar los widgets, asegurando así un aspecto consistente y similar en todas las plataformas [29].



*Figura 2.7. Logo de Flutter  
Fuente: [30]*

#### **2.4.2.2 React Native**

Framework que fusiona las ventajas del desarrollo nativo con las capacidades de React, una de las mejores bibliotecas de JavaScript para interfaces de usuario. Facilita el desarrollo de aplicaciones móviles nativas tanto para iOS como para Android. Este framework se centra en el uso de componentes, lo que simplifica la escritura y reutilización de código. Además, permite a los desarrolladores crear aplicaciones con un rendimiento cercano al nativo y una experiencia de usuario fluida, aprovechando la eficiencia y flexibilidad de React [31].



*Figura 2.8. Logo de React Native  
Fuente: [32]*

## **2.5 Bases de Datos**

Una base de datos es una colección de información o datos estructurados que se encuentran almacenados informáticamente. Además, se encuentra controlada por un sistema de gestión de bases de datos (DBMS), que es el software que le permite poder almacenar, editar y recuperar información dentro de ella [33][34].

Debido a que existen diversos tipos de bases de datos, el presente trabajo de tesis hará enfoque en describir los tipos de bases de datos más conocidos.

### 2.5.1 Bases de Datos Relacionales

También conocido como base de datos SQL. En este tipo de base de datos, la información almacenada se organiza en forma de tablas, compuesta por columnas y filas. Cada fila representa los valores de los datos y cada columna representa una característica o atributo de los datos [33][34]. Además, a través del uso de instrucciones en SQL, lenguaje de consulta estructurado, se va a poder realizar un CRUD con la información guardada en la base de datos [35]. Asimismo, las operaciones que se realizan en una base de datos relacional cumplen con las propiedades de atomicidad, coherencia, aislamiento y durabilidad (ACID) [36]:

- **Atomicidad:** Garantiza que las transacciones se ejecuten por completo o no se realicen.
- **Consistencia:** Asegura que la información almacenada se ajuste al esquema de la base de datos tras confirmarse una transacción.
- **Aislamiento:** Mantiene la ejecución de las transacciones separadas unas de otras.
- **Durabilidad:** Capacidad de la base de datos para recuperarse después de un fallo inesperado.

### 2.5.2 Bases de Datos No Relacionales

También conocidos como bases de datos NoSQL, y como su nombre lo dice, la información en esta base de datos ya no se guarda en forma de tablas. Este enfoque permite guardar la información en una estructura de datos, por ejemplo, en un documento JSON, brindando una mayor flexibilidad al manejar grandes volúmenes de datos, puesto que se rompe con la dependencia de una estructura tradicional. Asimismo, estas bases de datos pueden escalar horizontalmente para satisfacer una mayor cantidad de tráfico [37][38]. Entre sus tipos se incluyen:

- **Almacenes de Clave-Valor:** Cada elemento se estructura a través de un diccionario de pares clave-valor. Entre las bases de datos de este tipo tenemos a Riak, Berkeley DB, etc.
- **Orientado a Documentos:** Aquí los datos se representan como un documento de tipo JSON, BSON o XML. La base de datos más conocida en este tipo es MongoDB.
- **Almacenes de Grafos:** Los datos se organizan como nodos y aristas, siendo estos últimos las relaciones o conexiones entre los nodos. Algunas bases de datos de este tipo son Neo4J, Giraph, etc.
- **Orientado a Columnas:** Almacenan la información en columnas, permitiendo realizar consultas a columnas específicas obteniendo únicamente los valores requeridos. Aquí encontramos las bases de datos como Cassandra, Hbase, etc.

La mayoría de bases de datos NoSQL suelen ajustar algunas propiedades ACID de las bases de datos SQL, para ofrecer un modelo más flexible y escalable horizontalmente. Esta característica las hace adecuadas para situaciones que requieren un rendimiento elevado y baja latencia [36].

## 2.6 Estado del arte

En esta sección, se describirán las distintas implementaciones existentes relacionadas con el tema del presente trabajo de tesis.

En 2015, se desarrolló un sistema de gestión administrativa para la Escuela de Conductores CASYC LTDA. en Hualpén, Chile [39]. Se utilizó Yii-Framework, un framework de PHP para aplicaciones web, y MySQL como sistema de gestión de bases de datos, buscando mejorar la eficiencia y organización de la escuela.

Años después, en 2017, se desarrolló un sistema web automatizado para la Escuela de Manejo Cortez en Managua, Nicaragua [40]. Este sistema estaba enfocado en mejorar la gestión

administrativa de la escuela y para su construcción se empleó C# como lenguaje de programación y SQL Server 2016 para gestión de la base de datos.

Justo en el mismo año, se implementó una aplicación móvil y web para la gestión administrativa y académica en la Escuela de Conductores Profesionales Ambato, Ecuador [41], que, a diferencia de las anteriores, esta escuela está orientada a la formación y capacitación de los conductores a través de diversos módulos de enseñanza, sin enfocarse en la parte práctica de conducción. Su plataforma web la desarrolló empleando el framework ASP.NET, que utiliza el lenguaje de programación C# y para su aplicativo, que está enfocado en los dispositivos móviles con sistema operativo Android, empleó Java para su construcción. Asimismo, como motor de base de datos empleó SQL Server 2012.

En [42], también en 2017, el autor desarrolló un simulador de conducción basado en realidad virtual que contribuye a mejorar la habilidad conductiva de la persona. En este caso, se usó como motor de gráficos a Unity 3D para el desarrollo del entorno del sistema y se integró con Kinect, un dispositivo usado para la detección de movimientos de la persona durante la conducción.

Al año siguiente, 2018, en [43], el autor realizó una aplicación móvil para Android llamada “SafeCar” que estimula una conducción eficiente contribuyendo a mejorar la habilidad conductiva de la persona. Como sistema de gestión de base de datos se usó SQLite y como lenguaje de programación a Java.

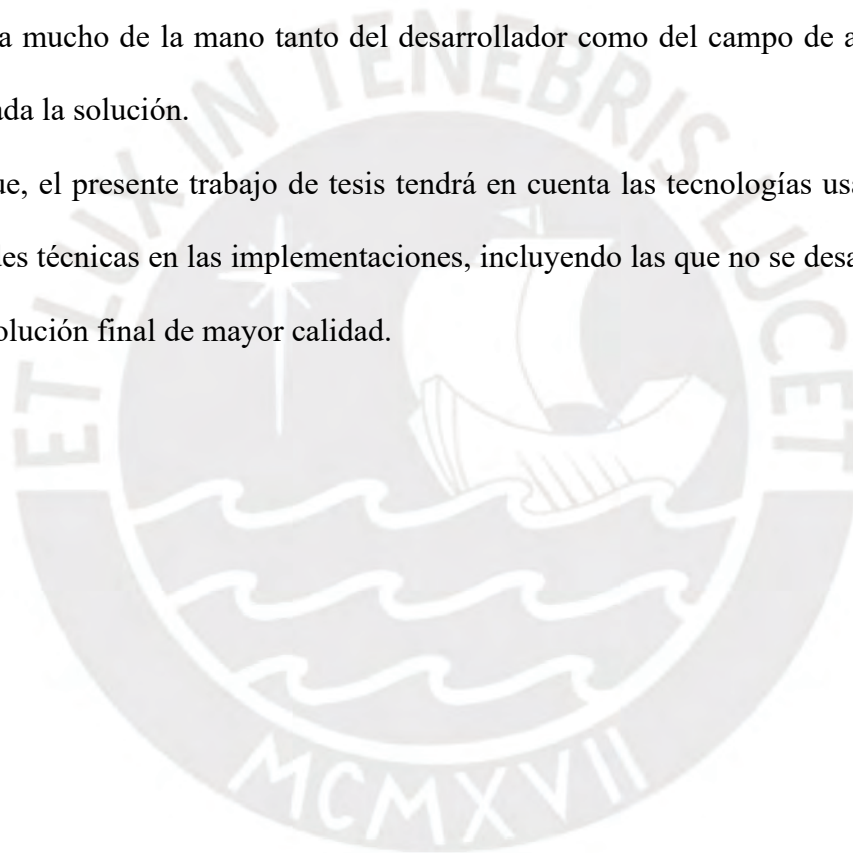
Finalmente, una implementación más reciente se realizó el año pasado [44], en 2022. El MTC implementó una plataforma web para los exámenes médicos de brevets de clase A y B, la cual podría ser usada por todos los centros médicos autorizados por el estado eliminando la gestión manual que antes estas entidades realizaban.



### 2.6.1 Síntesis de las implementaciones descritas

En resumen, de todas las implementaciones descritas, se puede apreciar que existen coincidencias en cuanto a las tecnologías usadas para el desarrollo de sus soluciones. Con respecto a las aplicaciones móviles, se puede ver un aprecio por usar Java como lenguaje de programación. En contraste, no se puede decir lo mismo de las aplicaciones web ya que hay variedad tecnológica. Asimismo, de las implementaciones sobre aplicaciones web y móvil, todas emplearon sistemas de bases de datos relacionales. Cabe mencionar que la elección de las tecnologías va mucho de la mano tanto del desarrollador como del campo de aplicación a lo que va enfocada la solución.

Es por eso que, el presente trabajo de tesis tendrá en cuenta las tecnologías usadas como las funcionalidades técnicas en las implementaciones, incluyendo las que no se desarrollaron para brindar una solución final de mayor calidad.



### **Capítulo 3. Diseño e implementación de la propuesta de solución**

En el presente capítulo se presentarán los requerimientos de la aplicación web y móvil, la arquitectura de la solución propuesta y el análisis y descripción de sus componentes. Además, se explicará la implementación y flujo de operación de la funcionalidad de control de acceso de la aplicación web y móvil, reserva de clases de manejo en la aplicación web y envío de mensajes de notificación a usuarios. Por último, se mostrarán las principales interfaces de la aplicación web y móvil.

#### **3.1 Requerimientos de la solución**

Para el desarrollo y pruebas de las aplicaciones, se utilizará el nombre de “EduCar (EC)”.

##### **3.1.1 Aplicación web**

A continuación, se listan las funcionalidades requeridas para el desarrollo de la plataforma web:

- **R1. Gestión de Usuarios basada en Roles:** Se requiere implementar un sistema de control de acceso diferenciado por roles, de acuerdo a los actores involucrados en una escuela de manejo, tal como se detalla en la Sección 2.2.

- **R2. Programación, Pago y Reserva de Clases de Manejo:** El sistema brindará a los alumnos la capacidad de seleccionar horarios de clases de manejo según su disponibilidad y la del instructor deseado. Además, podrá completar una pasarela de pagos online para confirmar y reservar su clase.
- **R3. Bloqueo de Horarios de Manejo:** El sistema facilitará al personal administrativo la posibilidad de bloquear horarios para las clases de manejo, inhabilitando su programación, en función de la disponibilidad de los instructores.
- **R4. Registro de Operaciones de Vehículos:** El personal administrativo podrá registrar información detallada de los vehículos como de sus operaciones, incluyendo recargas de combustible y mantenimientos efectuados.
- **R5. Visualización de Informes de Aprendizaje:** Los alumnos podrán visualizar los informes de aprendizaje, que son elaborados por los instructores después de cada clase de manejo. Estos informes incluirán detalles de la clase, observaciones y feedback proporcionados por el instructor.
- **R6. Concurrencia de Alumnos:** El sistema deberá soportar una cantidad de 31 alumnos concurrentes, basándose en el promedio de alumnos mostrado en la Figura 1.7.
- **R7. Descarga de información del Balance de Operaciones:** Se propone que el personal administrativo pueda descargar un archivo .xlsx que contenga información relevante para el balance de operaciones y la opción de filtrar por fecha. Según la encuesta realizada, el informe deberá incluir datos como las horas de clase impartidas por instructor, horas de uso de los vehículos, cantidad de combustible recargado, monto total de los mantenimientos de los vehículos, registro de pagos de los alumnos por sus clase de manejo y el seguimiento de los alumnos matriculados.

### 3.1.2 Aplicación móvil

A continuación, se listan las funcionalidades requeridas para el desarrollo de la plataforma móvil:

- **R8. Envío de Notificaciones como Recordatorios:** Se propone implementar el envío de notificaciones como recordatorios a los alumnos 15 minutos antes de su clase de manejo programada.
- **R9. Visualización de Clases Programadas:** Los alumnos como instructores podrán visualizar sus clases programadas, organizadas desde las más próximas a las más lejanas.
- **R10. Creación de Informes de Aprendizaje:** Deberá permitir a los instructores generar informes de aprendizaje para sus alumnos después de cada clase impartida, incluyendo capacidad de registrar información especificada en R5, como la subida de imágenes de la galería.

## 3.2 Arquitectura de la solución

La arquitectura de la solución propuesta, detallada en la Figura 3.1, incluye los siguientes módulos:

- Microsoft Azure
  - Azure DNS
  - Azure Virtual Machines
  - Azure Blob Storage
  - Azure Database for MySQL
- Stripe
- Google Cloud Platform
  - Firebase Cloud Messaging

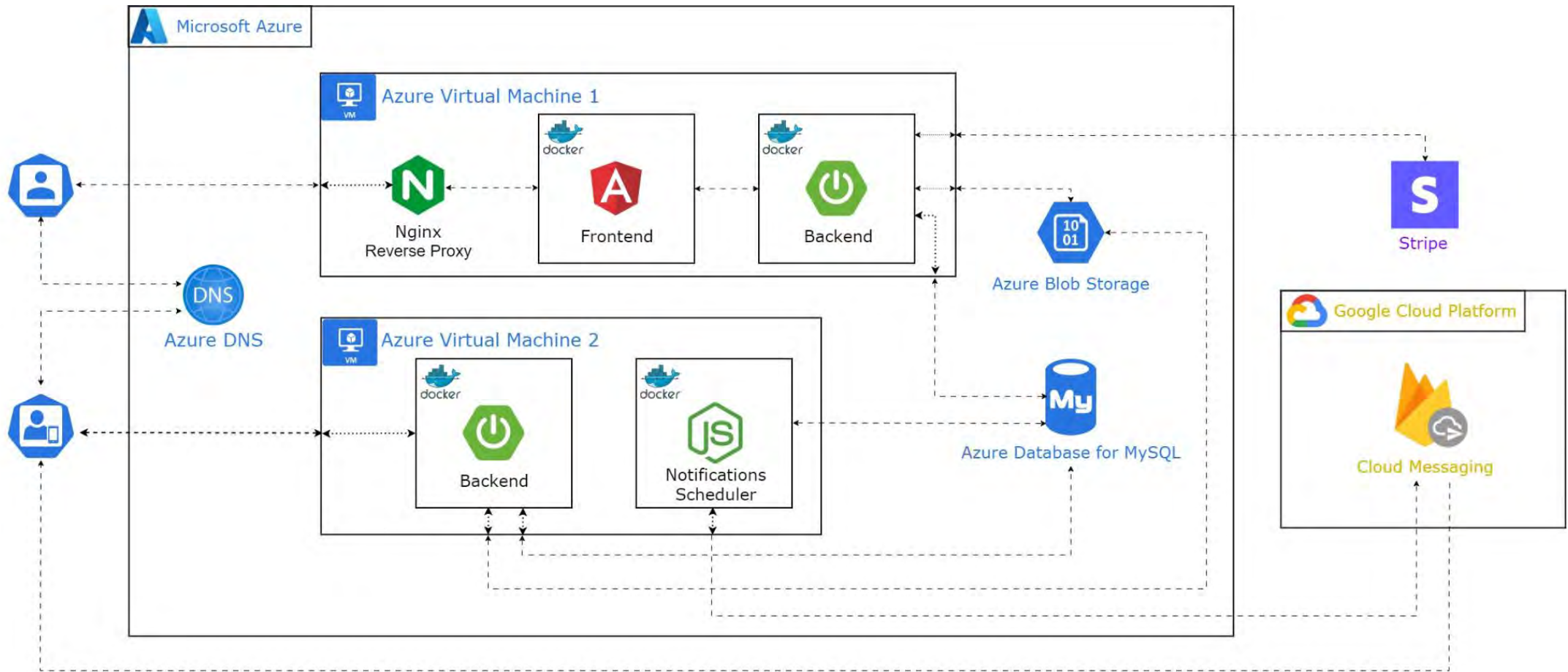


Figura 3.1. Arquitectura de la solución propuesta  
Fuente: Elaboración propia

A continuación, se describirán los componentes propuestos en la arquitectura de solución:

### 3.2.1 Azure DNS

Servicio de administración de nombres de dominio (DNS) que proporciona escalabilidad, seguridad y alta disponibilidad para resolver nombres de dominio a través de la infraestructura de Microsoft Azure [45]. Dentro de la solución propuesta, se empleará un nombre de dominio adquirido específicamente para las plataformas. El uso de este servicio permitirá registrar y gestionar dicho dominio de una manera rápida y eficiente, como se muestra en la Figura 3.2. Se usará un nombre de dominio en particular ya que le brinda a los usuarios, quienes interactúan con la plataforma web, una forma más accesible de acceder al servidor que aloja dicha plataforma, en lugar de tener que recordar su IP pública.

Esto permitirá dirigir las consultas de los usuarios, redirigiendo sus solicitudes a la IP pública de las Azure Virtual Machines específicas, en función del subdominio solicitado. Como resultado, podemos enrutar eficazmente las solicitudes a las máquinas virtuales correspondientes, ya sea para acceder a la aplicación web o aplicación móvil, como se visualiza en la Figura 3.3.

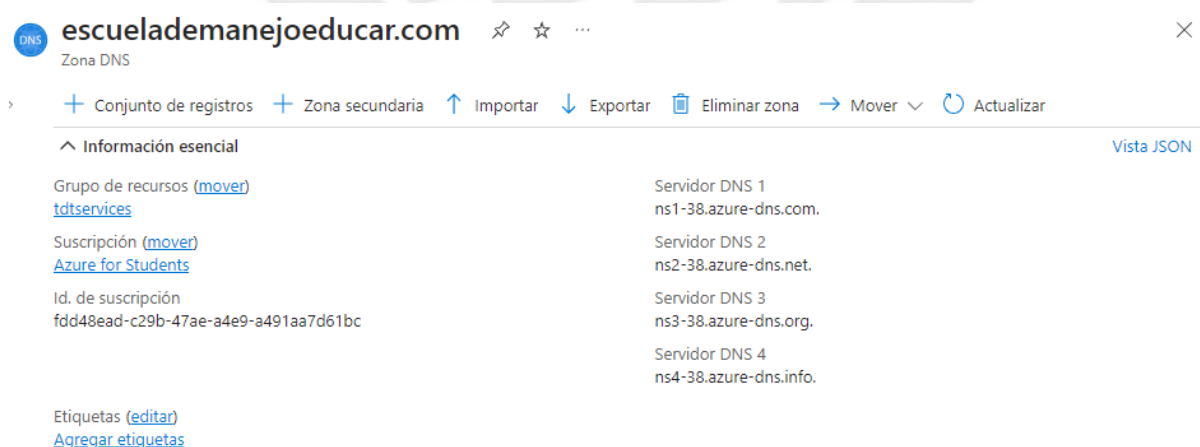


Figura 3.2. Dominio registrado en Azure DNS  
Fuente: Elaboración propia

Nombre	Tipo	TTL	Valor	Tipo de recurso del ...	Destino del alias
@	NS	172800	ns1-38.azure-dns.com, ns2-38.azure-dns.net, ns3-38.azure-dns.org, ns4-38.azure-dns.info.		***
@	SOA	3600	Correo electrónico: a... Host: ns1-38.azure-d... Actualizar: 3600 Reintentar: 300 Expira el: 2419200 TTL mínimo: 300 Número de serie: 1		***
app-mobile	A	3600	-	Dirección IP pública	vm2-ip
www	A	3600	-	Dirección IP pública	vm1-ip

Figura 3.3. Configuración para redirección de tráfico a la Azure Virtual Machine por nombre de dominio

Fuente: Elaboración propia

### 3.2.2 Azure Virtual Machines

Servicio que proporciona virtualización de recursos de cómputo, como memoria, almacenamiento y CPU en una instancia, ofreciendo la flexibilidad de escalar vertical (hacia arriba o abajo) u horizontalmente según las necesidades de la aplicación [46]. Se utilizará específicamente para levantar servidores web en estas máquinas virtuales.

#### 3.2.2.1 Azure Virtual Machine 1 (Aplicación web)

##### a. NGINX como Reverse Proxy

Servidor HTTP de código abierto y totalmente gratuito que se puede configurar para funcionar como proxy inverso (reverse proxy), balanceador de carga, entre otras funciones [47]. Este servidor es conocido por estar diseñado para ofrecer una alta concurrencia y un bajo consumo de recursos. Esto se logra gracias a su arquitectura escalable basada en eventos que opera de manera asíncrona, el cual hace que las solicitudes de los usuarios se manejen en un solo hilo, lo que permite el procesamiento de muchas solicitudes al mismo tiempo [48].

En la solución propuesta, NGINX se configurará como proxy inverso, el cual hará que actúe como un intermediario entre el cliente y el módulo Frontend, interceptando las solicitudes de los clientes y reenviándolas al módulo mencionado anteriormente. También, se integrará con el

software gratuito Certbot para la obtención y renovación automática de certificados SSL, brindado por *Let's Encrypt*, entidad de certificación [49]. Esto permitirá habilitar HTTPS en el servidor NGINX, permitiendo cifrar el tráfico entre el usuario y la plataforma web.

### **b. Frontend**

Se propone el desarrollo de este módulo en Angular, ya que es uno de los frameworks más utilizados para el desarrollo del frontend en aplicaciones web [50]. Además, se destaca por su escalabilidad y su amplia gama de características, lo que lo convierte en una solución eficiente para el desarrollo de aplicaciones de cualquier tamaño. Este se encargará de la presentación, es decir, se encargará de crear y renderizar la interfaz de usuario en la plataforma web. Estas vistas contendrán datos que serán proporcionados por el módulo Backend a través de servicios API REST. Esto permitirá que los usuarios interactúen con la plataforma y puedan visualizar la información de manera estructurada y organizada.

### **c. Backend**

Se propone el desarrollo de este módulo en Spring Boot, framework que es ampliamente reconocido y utilizado por los desarrolladores debido a su robustez y eficiencia en el desarrollo del backend de aplicaciones web [50]. Spring Boot se destaca por su configuración y arranque automático, así como por su facilidad de integración con servicios de terceros, lo que lo hacen ideal para el desarrollo de aplicaciones empresariales en Java [24].

Este se encargará de la lógica de negocio y se comunicará con Azure Database for MySQL para la gestión de información de la aplicación. Los datos gestionados por esta aplicación se devolverán al módulo de Frontend para su presentación en la plataforma web, permitiendo que los usuarios puedan visualizarlos. Además, se comunicará con Azure Blob Storage para el almacenamiento y obtención de archivos, como imágenes y documentos PDF. También, se



integrará con Stripe para habilitar la pasarela de pago en línea que permitirá a los alumnos reservar horarios para clases de manejo.

### **3.2.2.2 Azure Virtual Machine 2 (Aplicación móvil)**

#### **a. Backend**

Se propone el desarrollo de este módulo en Spring Boot, cuyas razones se encuentran explicadas anteriormente. Este se comunicará con Azure Database for MySQL para la gestión de información y que será devuelta a la aplicación móvil cuando sea solicitada. Además, se comunicará con Azure Blob Storage para el almacenamiento de imágenes y obtención de documentos PDF generados.

#### **b. Notifications Scheduler**

Se propone el desarrollo de este módulo en Node.js debido a su capacidad para el manejo de eventos asíncronos, la ejecución de tareas programadas y su fácil integración con Firebase, que se empleará para el manejo de las notificaciones. Además, se comunicará con Azure Database for MySQL e integrará con Firebase Cloud Messaging (FCM) para el envío de notificaciones a modo de recordatorio a aquellos que tengan clases programadas próximamente, 15 minutos antes que estas comiencen.

### **3.2.3 Azure Blob Storage**

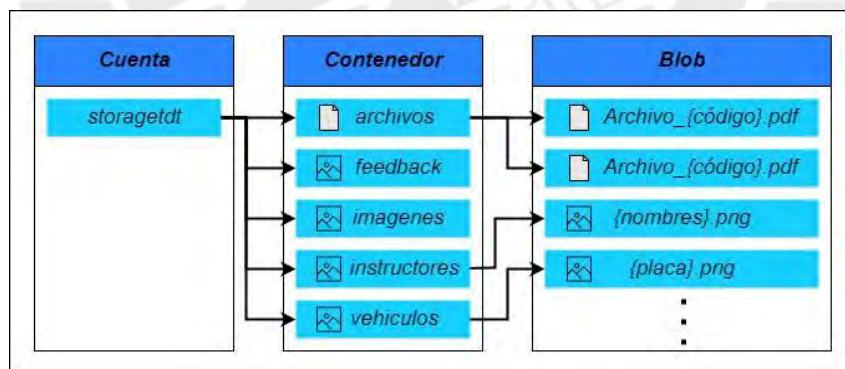
Servicio que permite el almacenamiento masivo de archivos en la nube de Azure. Este servicio se compone de tres recursos fundamentales [51]:

1. Cuenta de almacenamiento: Proporciona un espacio de nombre único para los datos, el cual constituirá la base de los puntos de conexión (URLs) para el acceso de los archivos subidos.

2. Contenedor en la Cuenta de Almacenamiento: Espacio que permite organizar un conjunto de blobs, ofreciendo una estructura similar a un directorio de archivos.
3. Blob en un Contenedor: Es el archivo que se sube a la nube de Azure para su almacenamiento.

Para la solución propuesta, este servicio se usará para el almacenamiento de imágenes y documentos en formato .pdf. Es importante destacar que no se recurrirá al disco de la Azure Virtual Machine como medio de almacenamiento. La razón de esto radica en que con el tiempo, el espacio de almacenamiento de la máquina virtual se va a llenar debido a la generación y almacenamiento recurrente de documentos en formato .pdf, lo que resultaría en problemas de memoria, errores y ralentización en el procesamiento de datos. Además, la disminución en la velocidad de procesamiento aumentaría el tiempo para acceder a la información almacenada en la Azure Virtual Machine, lo que la convierte en una opción ineficiente.

Con el fin de garantizar una correcta organización de los archivos en el Azure Blob Storage, la propuesta de solución establecerá la siguiente estructura:



*Figura 3.4. Estructura de recursos definidos en Azure Blob Storage  
Fuente: Elaboración propia*

Según lo representado en la Figura 3.4, la estructura de archivos es la siguiente:

- El contenedor de “archivos” contendrá las boletas de pago generadas al completar la transacción para la reserva de horarios de clases de manejo.

- El contenedor de “feedback” contendrá las imágenes de retroalimentación que los instructores cargan en los informes de aprendizaje.
- El contenedor de “imágenes” contendrá las imágenes relacionadas a la escuela de manejo de la aplicación web.
- El contenedor de “instructores” contendrá las imágenes de los instructores registrados en la escuela de manejo.
- El contenedor de “vehículos” contendrá las imágenes de los vehículos registrados en la escuela de manejo.

#### **3.2.4 Azure Database for MySQL**

La plataforma web y móvil necesitará de una capa de persistencia de datos debido a los requerimientos mencionados en la Sección 3.1. Es de vital importancia almacenar los datos de los usuarios, clases de manejo, vehículos, informes de aprendizaje y pagos para su gestión. Esto incluye el almacenamiento de usuarios registrados en las plataformas, control de acceso de usuarios basado en roles, almacenamiento de horarios de clase de manejo de los alumnos e instructores, almacenamiento de los horarios de instructores en los que no será posible reservar clases de manejo, almacenamiento de información de pagos de los alumnos por la reserva de sus clases de manejo, almacenamiento de los vehículos utilizados en la escuela para las clases de manejo así como sus operaciones de mantenimiento y recarga de combustible, y almacenamiento de información de los informes de aprendizaje y las retroalimentaciones que contiene por clase de manejo. Como se mencionó en la Sección 2.5, existen dos opciones de elección de bases de datos: las relacionales (SQL) y las no relacionales (NoSQL). Para la elección se analizará el tipo de información que debe ser registrada y la forma en que se va a guardar. Para esto, se organiza la información de acuerdo a los requerimientos previamente mencionados, lo cual se detalla en la Tabla 3.1.

Se puede observar en la Tabla 3.1 que la información tendrá relaciones entre ellas. En primer lugar, los usuarios con roles de alumno e instructor tendrán horarios de clase de manejo asignados. En segundo lugar, cada clase programada estará vinculada a un vehículo utilizado, un pago correspondiente y un informe de aprendizaje generado. En tercer lugar, cada instructor tiene asignado horarios bloqueados en los que no podrá impartir clases de manejo. Por último, cada vehículo tiene asignado operaciones de mantenimiento y recargas de combustible.

En adición, se puede notar que la información almacenada puede tener una estructura definida que no necesariamente debe ser dinámica. Por ejemplo, cada clase programada que se almacene tendrá características uniformes, como el alumno que reservó la clase, el horario de la clase de manejo, el instructor escogido y el tipo de vehículo a utilizar. Lo mismo aplica a las entidades de Usuario, Pagos, Informes de Aprendizaje, Vehículos, Mantenimientos y Recargas de Combustible.

*Tabla 3.1. Análisis de la información a ser almacenada*  
*Fuente: Elaboración propia*

<i>Operación</i>	<i>Entidades</i>	<i>Descripción</i>
Gestión de Usuarios	Usuario	Representa a los usuarios que harán uso de la plataforma web y móvil. Se almacenarán sus datos personales (nombres, correo, contraseña, celular, dni, fecha de nacimiento) y se les asignará un rol (alumno, instructor o administrador).
Gestión de Clases de Manejo	Clases Programadas	Representa las clases de manejo que el alumno ha reservado. Se almacenará el horario seleccionado, el instructor escogido y el tipo de vehículo que utilizará.
	Horarios Bloqueados	Representa los horarios en los que los instructores no estarán disponibles para dictar las clases de manejo. Se almacenará el horario no disponible y el motivo correspondiente.

Gestión de Pagos	Pagos	Representa los pagos efectuados por los alumnos para asegurar la reserva de sus clases de manejo. Se almacenará el monto total, código de la operación, fecha de la transacción, información del usuario que realizó (nombres, correo), enlace al comprobante de pago y la clase por la cual se efectuó el pago.
Gestión de Vehículos	Vehículos	Representa los vehículos que se utilizarán en la escuela de manejo. Se almacenarán sus datos principales (modelo, marca, placa, kilometraje) y qué tipo de vehículo es (automático o mecánico).
	Mantenimientos	Representa las operaciones de mantenimiento realizadas a los vehículos. Se almacenarán la descripción de la operación, fecha y hora en que se efectuó, monto total de la operación, lugar donde se llevó a cabo, persona responsable de la operación y el vehículo al que se le realizó el mantenimiento.
	Recargas de Combustible	Representa las operaciones de recargas de combustible realizadas a los vehículos. Se almacenarán la cantidad de combustible (en galones) llenado, monto total de la recarga, fecha y hora en que se efectuó y el vehículo al que se le realizó la recarga de combustible.
Gestión de Informes de Aprendizaje	Informes de Aprendizaje	Representa los informes de aprendizaje que se generan para los alumnos después de sus clases de manejo. Se produce un informe por cada clase y se almacenarán los detalles de la clase, lo que se llevó a cabo durante la misma, observaciones y comentarios de mejora dirigidos al alumno.

Considerando lo mencionado anteriormente, se optará por el uso de una base de datos relacional, ya que es la que mejor se adecua. De las bases de datos SQL que existen, se usará MySQL, ya que es la más usada como capa de persistencia de datos para aplicaciones web, debido a su velocidad de lectura de datos gracias a la rápida ejecución de queries, en comparación con los demás [52].

Para la solución propuesta se usará el servicio de Azure Database for MySQL. Este servicio se encarga de la completa administración de la base de datos, incluyendo tareas como

actualizaciones de software, copias de seguridad, etc., liberando así al usuario de la necesidad de configurar estas tareas y eliminando sus preocupaciones al respecto [53].

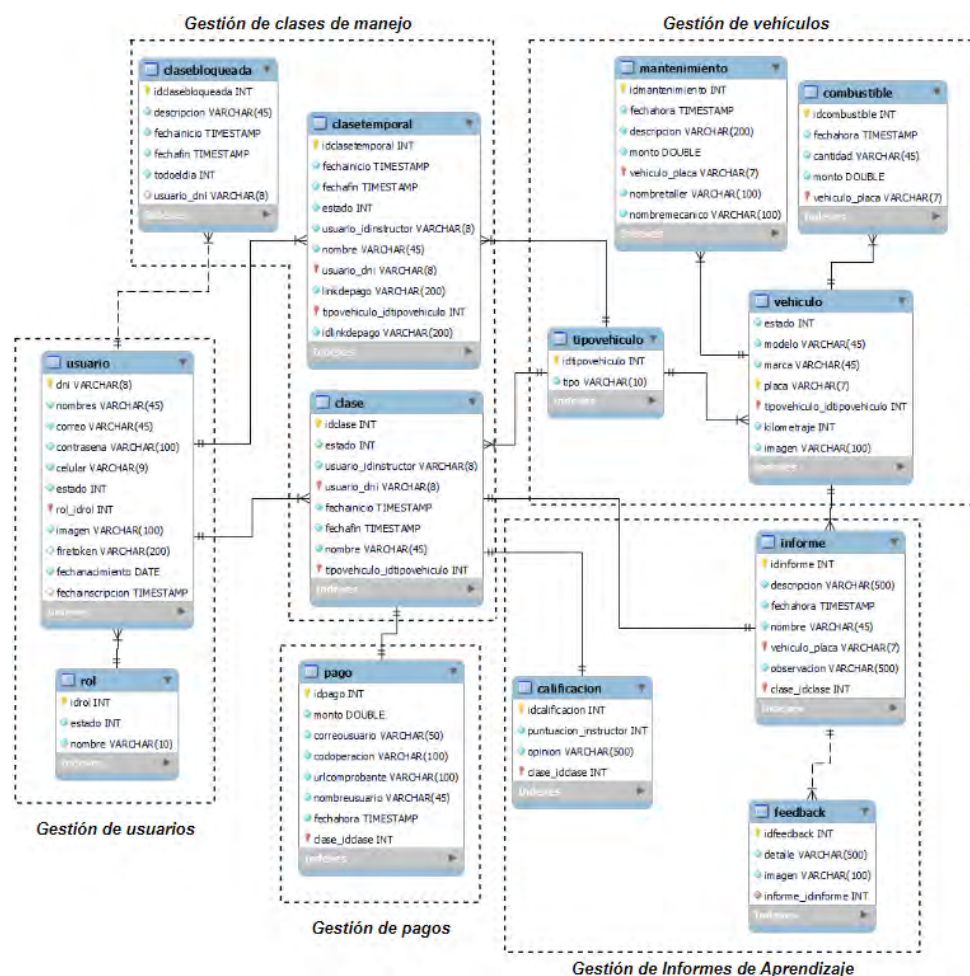


Figura 3.5. Diseño del diagrama entidad-relación de la base de datos  
Fuente: Elaboración propia

Finalmente, se propone el diagrama entidad-relación de la base de datos a implementar representando en la Figura 3.5, la cual se ha desarrollado basándose en la estructura de información detallada en la Tabla 3.1.

Tabla 3.2. Funcionalidades obtenidas del diagrama entidad-relación propuesto  
Fuente: Elaboración propia

<i>Operación</i>	<i>Tabla</i>	<i>Descripción</i>
Gestión de Usuarios	usuario	Representa a la entidad “Usuario” de la Tabla 3.1.
	rol	Tabla que almacenará los roles de alumno, instructor y administrador para el control de acceso de los usuarios.

Gestión de Clases de Manejo	clase	Representa a la entidad “Clases Programadas” de la Tabla 3.1.
	clasebloqueada	Representa a la entidad “Horarios Bloqueados” de la Tabla 3.1.
	clasetemporal	Tabla que almacenará temporalmente, durante un periodo de 5 minutos, las clases que los usuarios han intentado reservar pero aún no han completado el proceso de pago respectivo. Los campos almacenados son idénticos a los de la entidad “Clases Programadas” de la Tabla 3.1, incluyéndose el ID y enlace de pago proporcionado por Stripe.
Gestión de Pagos	pago	Representa a la entidad “Pagos” de la Tabla 3.1.
Gestión de Vehículos	vehiculo	Representa a la entidad “Vehículos” de la Tabla 3.1.
	tipovehiculo	Tabla que almacenará los tipos de vehículos: automático y mecánico.
	mantenimiento	Representa a la entidad “Mantenimientos” de la Tabla 3.1.
	combustible	Representa a la entidad “Recargas de Combustible” de la Tabla 3.1.
Gestión de Informes de Aprendizaje	informe	Representa a la entidad “Informes de Aprendizaje” de la Tabla 3.1.
	calificacion	Tabla que almacenará las puntuaciones y opiniones de los alumnos sobre la calidad de enseñanza de un instructor por clase impartida.
	feedback	Tabla que almacenará las retroalimentaciones que contiene un informe. Se almacenarán las imágenes subidas junto con sus descripciones respectivas.

### 3.2.5 Stripe

Stripe es una plataforma de pagos en línea que brinda a las empresas, de cualquier tamaño, la capacidad de aceptar pagos de sus clientes de todo el mundo [54]. Para la solución propuesta, específicamente para la pasarela de pago online destinada a la reserva de horarios clases de

manejo, se utilizará uno de los diversos servicios que la infraestructura de Stripe ofrece a los desarrolladores para integrar el procesamiento de pagos en línea en sus aplicaciones web y móviles.

En particular, se hará uso del servicio de “Enlaces de Pago” de Stripe. Este permitirá la creación de enlaces de pago personalizados que contienen una página web con información detallada sobre el pago y un formulario para llevar a cabo la transacción [55]. Estos enlaces de pago se utilizan para el pago de “productos”, que son elementos que han sido creados y configurados en la plataforma de Stripe, adaptados a las preferencias del vendedor, en este caso, para paquetes de clases de manejo de 30 minutos. Además, estos enlaces admiten una amplia gama de métodos de pago, incluidas las tarjetas de crédito y débito, que son las que estarán disponibles únicamente en la solución propuesta.

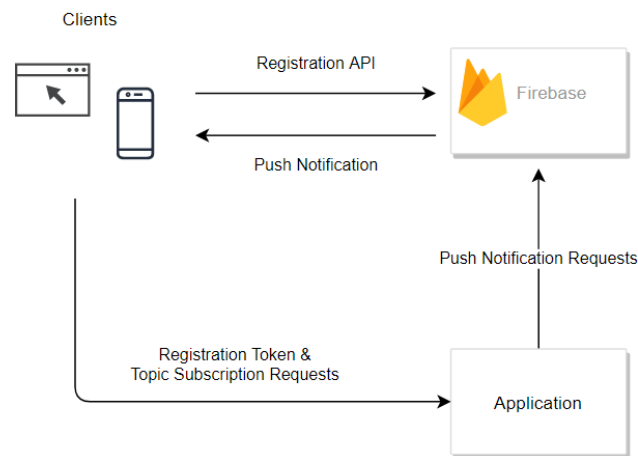
De esta manera, se optimizará el tiempo de desarrollo al evitar la necesidad de crear una página de pago desde cero y su posterior integración con las funcionalidades de procesamiento de pagos de Stripe en la aplicación web.

### **3.2.6 Firebase Cloud Messaging**

Para el envío de notificaciones en tiempo real a los usuarios se propone el uso de Firebase Cloud Messaging (FCM), que es un servicio gratuito de mensajería multiplataforma de Google Cloud Platform (GCP) que permite el envío seguro y sin costo de mensajes a dispositivos Android, iOS y web [56]. FCM permite el envío de dos tipos de mensajes: mensajes de notificación, que son breves mensajes de alertas dirigidos a los usuarios para notificarles sobre eventos definidos, y mensajes de datos, que son más extensos y se emplean para transmitir información a los usuarios [56]. Para la solución propuesta, se utilizarán mensajes de notificación, ya que el objetivo es enviar notificaciones a modo de recordatorios a aquellos usuarios que tengan clases programadas próximamente. Esto será posible gracias a que el servicio permite la segregación



de mensajes, que es el envío de mensajes a aplicaciones de forma individual, a grupos de dispositivos o a dispositivos suscritos a tópicos [56].



*Figura 3.6. Flujo de operación de envío de mensajes de notificación de FCM  
Fuente: [57]*

Según lo representado en la Figura 3.6, para llevar a cabo el envío de mensajes a través de FCM, se siguen varios pasos clave que en toda aplicación se debe realizar. En primer lugar, se debe registrar la aplicación móvil en el servidor de FCM. Luego, es necesario configurar el SDK de FCM en la aplicación móvil para permitir a los desarrolladores enviar solicitudes de mensajes al backend de FCM.

Una vez configurado, se registra una instancia de la aplicación móvil (cliente) para recibir mensajes y se obtiene un token de registro que permite identificar en forma única una instancia de la aplicación móvil (cliente). Este token es esencial para el envío de mensajes y se obtiene mediante una solicitud al backend de FCM. En el backend de la aplicación móvil, se implementa la lógica para la generación de mensajes de notificación. Luego, se envía una solicitud de mensaje al backend de FCM, que recibe esta solicitud. Utilizando el token de registro recibido, el backend de FCM identifica la instancia de la aplicación móvil (cliente) y realiza el envío del mensaje de notificación configurado. Finalmente, el dispositivo cliente recibe la notificación configurada.

### 3.3 Implementación de la solución

#### 3.3.1 Implementación de la aplicación web

##### 3.3.1.1 Control de Acceso

Para gestionar el control de acceso de los usuarios en la aplicación web, definido como R1, se sigue el flujo descrito en la Figura 3.7. Cuando un usuario ingresa sus credenciales para iniciar sesión, el módulo Backend valida si la cuenta existe. Si no existe la cuenta, se muestra un mensaje de error; en caso contrario, se permite el acceso del usuario a los servicios de la aplicación web en función a su rol.

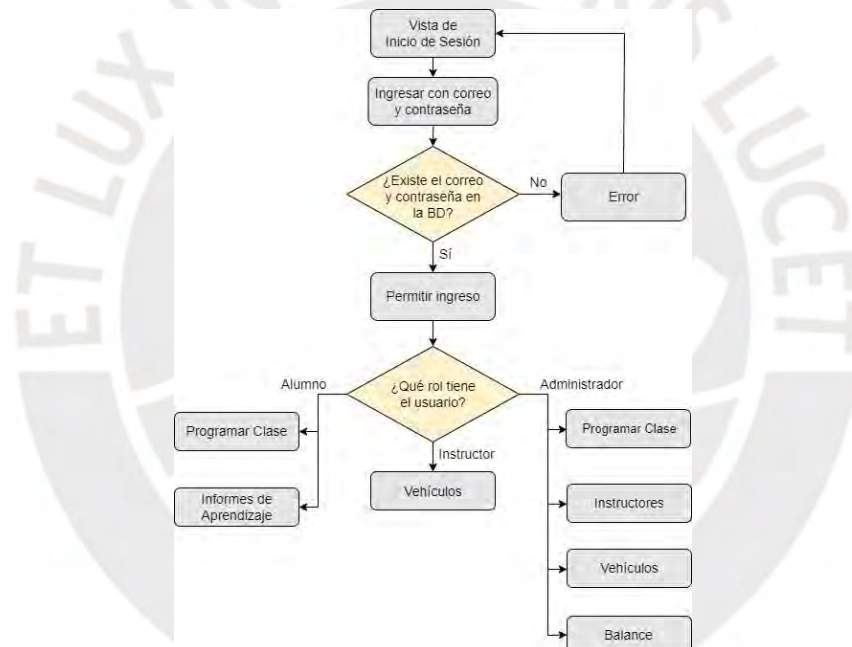


Figura 3.7. Flujo de Control de Acceso en la aplicación web  
Fuente: Elaboración propia

##### 3.3.1.2 Pasarela de pago online para reserva de clase de manejo

Como se mencionó en la Sección 3.2.5, para usar el servicio de “Enlaces de Pago” de Stripe, es necesario configurar primero un producto en su plataforma. Para ello se debe dirigir a la sección de “Catálogo de Productos” y darle a “Añadir Producto”. En la vista que aparece, como se muestra en la Figura 3.8, se deberá añadir la información del producto que se va a vender. Los

campos que se deben llenar son: nombres, imagen, descripción y precio del producto. Finalmente, se deberá dar en “Guardar Producto”.

**Información del producto**

Datos del producto

Nombre ⓘ  
Plan preferente, gafas de sol, etc.

Imagen ⓘ Opcional  
Cargar

Descripción ⓘ Opcional

Opciones adicionales: ▾

**Información sobre precios**

Detalles de las tarifas

Modelo de tarifas ⓘ  
Tarifas estándar

Precio ⓘ  
USD 0 USD

+ Añadir más divisas

Recurrente Único

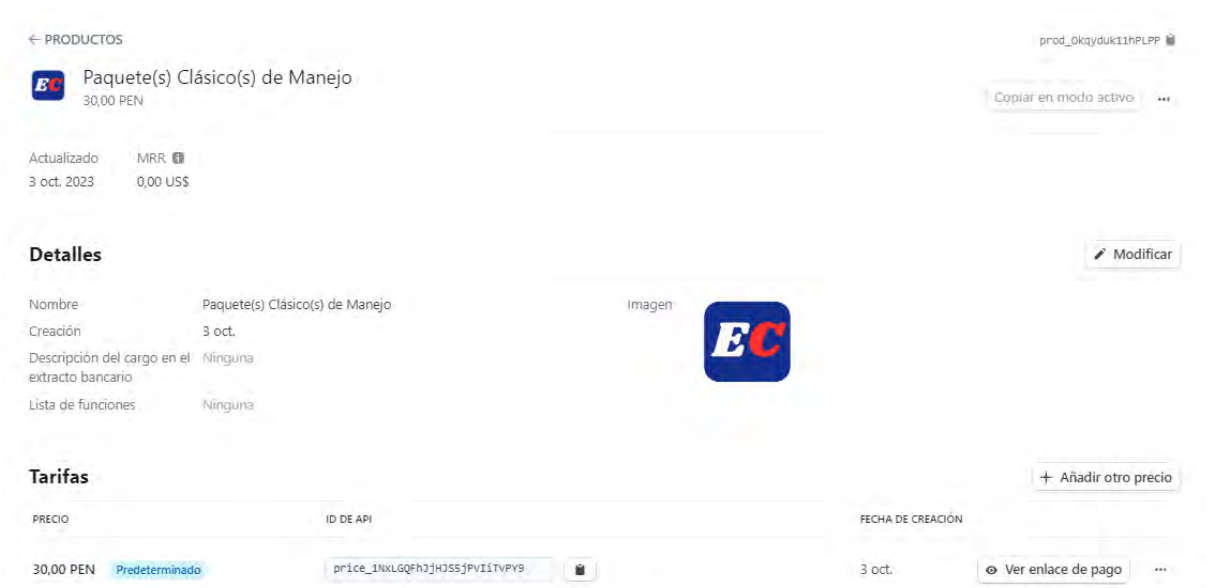
Vista previa de precios

Estados Unidos Ubicación actual

Subtotal  
Impuesto ⓘ  
Total

*Figura 3.8. Vista de información de producto en la plataforma de Stripe  
Fuente: Elaboración propia*

Para la solución, se creó el producto “Paquete(s) Clásico(s) de Manejo” cuya información se observa en la Figura 3.9. Un “Paquete(s) Clásico(s) de Manejo” mapea a la reserva de una clase de manejo por 30 minutos, por lo que, en caso de reservarse más tiempo para una clase, se añaden más cantidades de este producto para que se llegue al tiempo establecido en la reserva. La finalidad de esto es evitar la creación y gestión de diversos productos en la plataforma.



*Figura 3.9. Producto creado para su venta en la Pasarela de Pagos online  
Fuente: Elaboración propia*

La Figura 3.10 muestra el flujo de interacción a seguir en el módulo Backend para el proceso de pago con Stripe, mientras que la Figura 3.11 detalla las operaciones a seguir en el Frontend para la programación de clases de manejo, alineándose con el requerimiento R2. Cabe destacar que este módulo Backend tiene entre sus funcionalidades la capacidad de calcular la cantidad de 30 minutos que hay en el horario seleccionado por el alumno. Esta información proviene del módulo Frontend a través de un servicio API REST, y es esencial para determinar la cantidad de “Paquete(s) Clásico(s) de Manejo” que deben ser generados.

El enlace de pago, generado por la plataforma Stripe, conduce al alumno a una página web similar a la ilustrada en la Figura 3.12. Sin embargo, es importante tener en cuenta que este enlace de pago posee una vigencia limitada de 5 minutos. En otras palabras, el usuario cuenta con ese breve lapso de tiempo para completar la transacción y asegurar la reserva de su clase. Si el pago no se efectúa en el plazo establecido, el enlace de pago se invalida automáticamente, y la clase que había sido temporalmente registrada, el cual sirve para que se bloquee temporalmente dicho horario para evitar que otros alumnos reserven una clase en dicho horario

mientras el usuario está efectuando el pago, se libera, poniendo en disponibilidad dicho horario seleccionado hacia los demás alumnos.

Por otro lado, si el pago se completa dentro del intervalo de tiempo asignado, se procede a confirmar y registrar la reserva de la clase de manejo del alumno. La clase temporalmente bloqueada se elimina, y en enlace de pago se invalida, concluyendo el proceso de reserva exitosamente. Esta meticulosa gestión de tiempos y transacciones asegura la efectividad y equidad en la reserva de clases de manejo en la aplicación web.

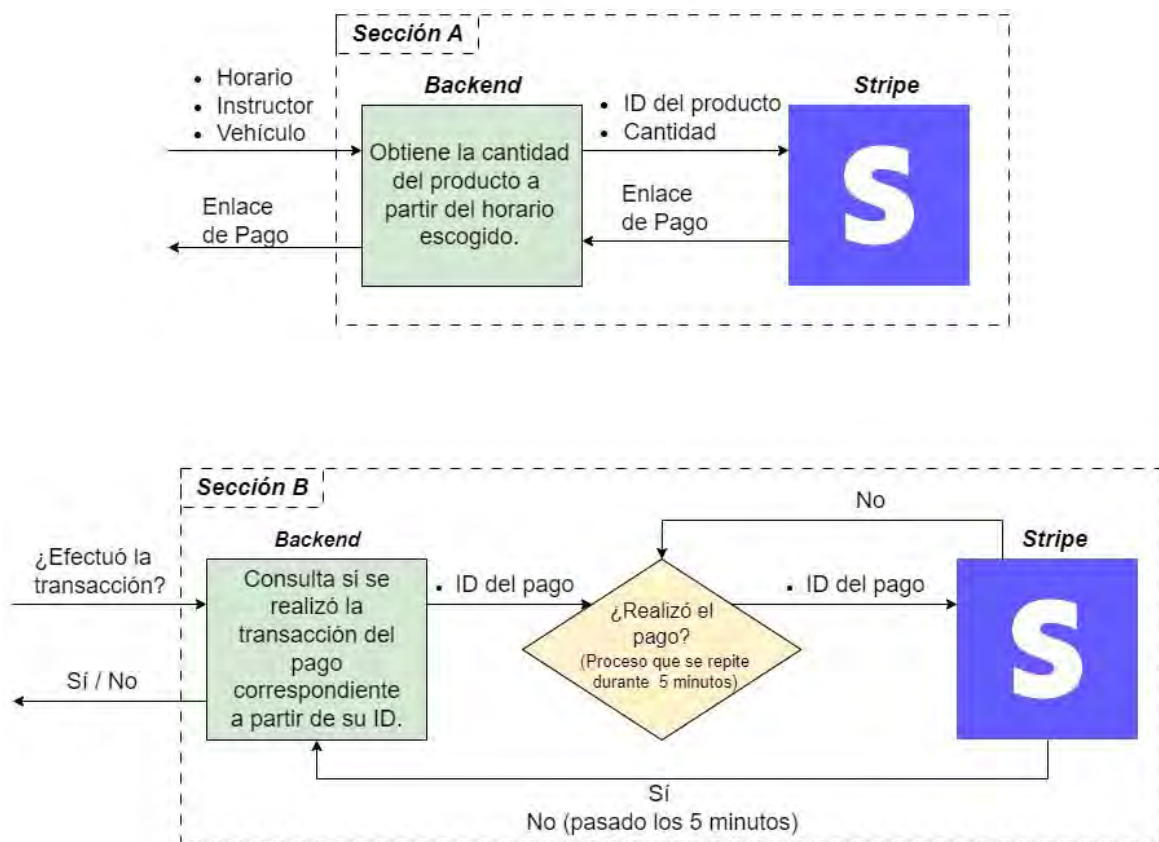


Figura 3.10. Diagrama de Flujo de Proceso del Pago Integrado con Stripe  
Fuente: Elaboración propia

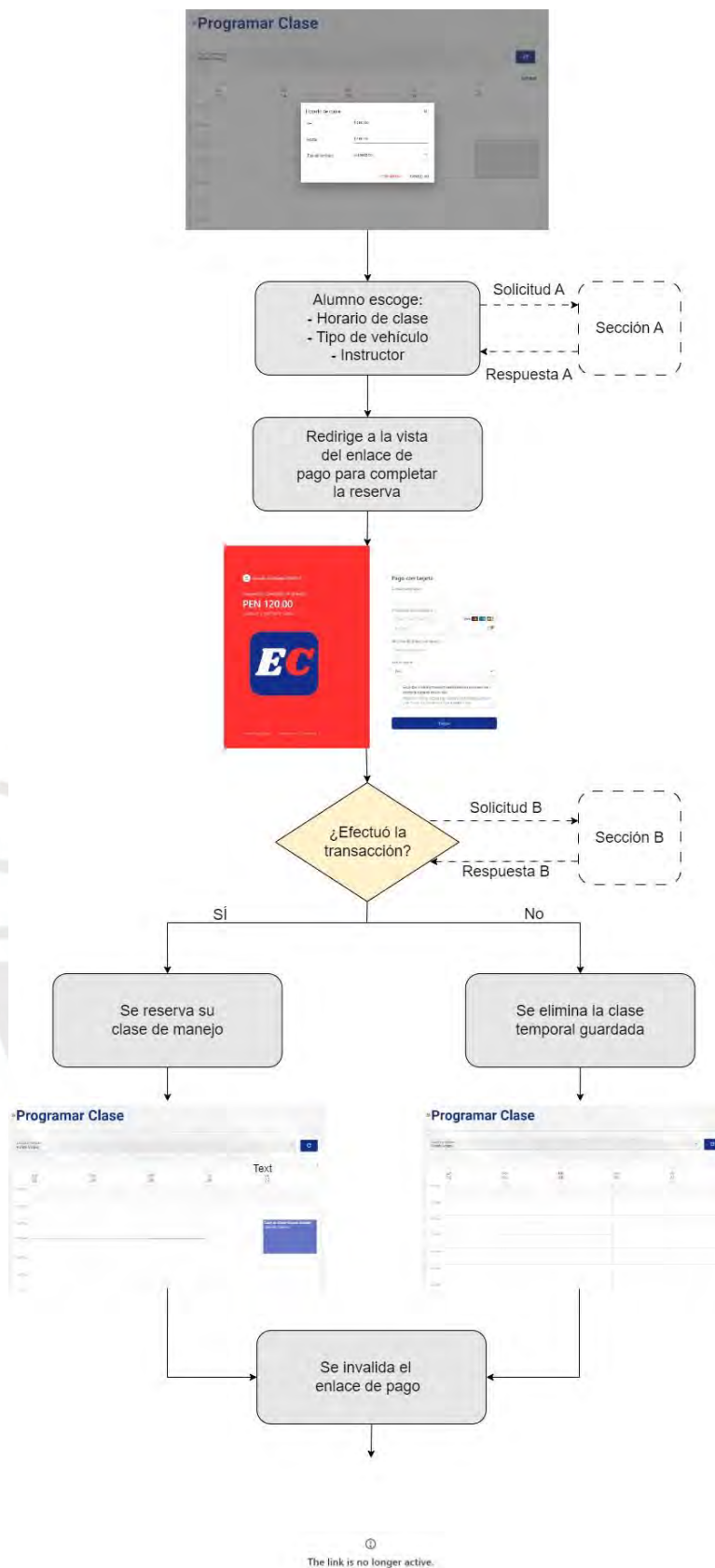
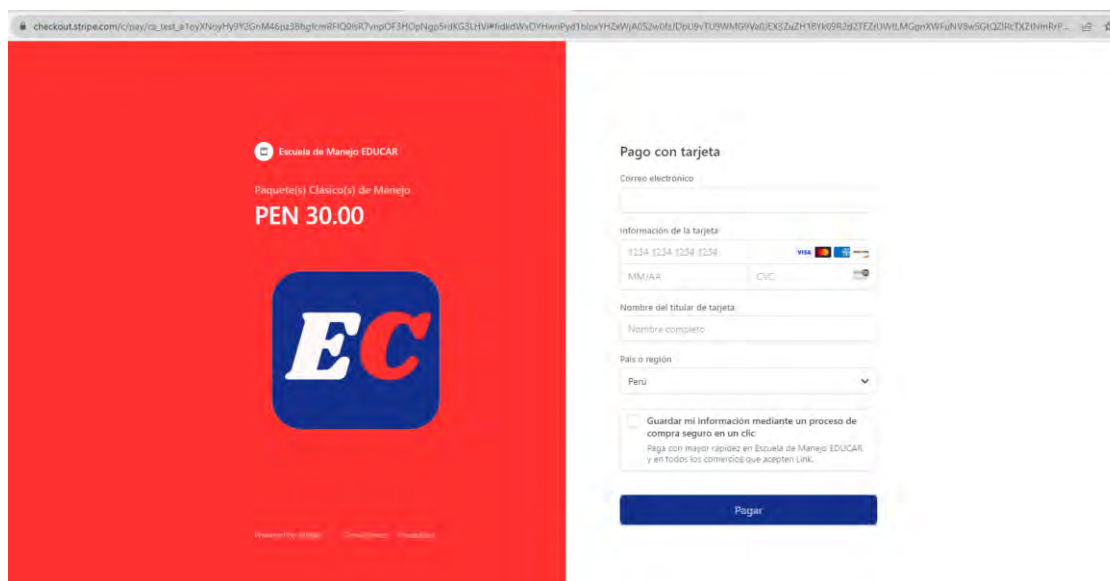


Figura 3.11. Flujo de operación de Pasarela de Pago online para reserva de clase de manejo  
Fuente: Elaboración propia



*Figura 3.12. Página web que contiene el enlace de pago generado por Stripe  
Fuente: Elaboración propia*

### 3.3.2 Implementación de la aplicación móvil

#### 3.3.2.1 Control de Acceso

Para gestionar el control de acceso de los usuarios en la aplicación móvil, se sigue el flujo descrito en la Figura 3.13. Cuando un usuario proporciona sus credenciales para iniciar sesión, el módulo Backend valida si la cuenta existe. Si no existe la cuenta, se muestra un mensaje de error; en caso contrario, se permite el acceso del usuario a los servicios de la aplicación móvil. Adicionalmente, cuando un usuario inicia sesión de manera exitosa, se crea el token de registro de FCM, el cual va a permitir identificar a los usuarios para el envío de mensajes de notificación, como se mencionó en la Sección 3.2.6. Además, se elimina el token de la base de datos cuando el usuario cierra sesión.

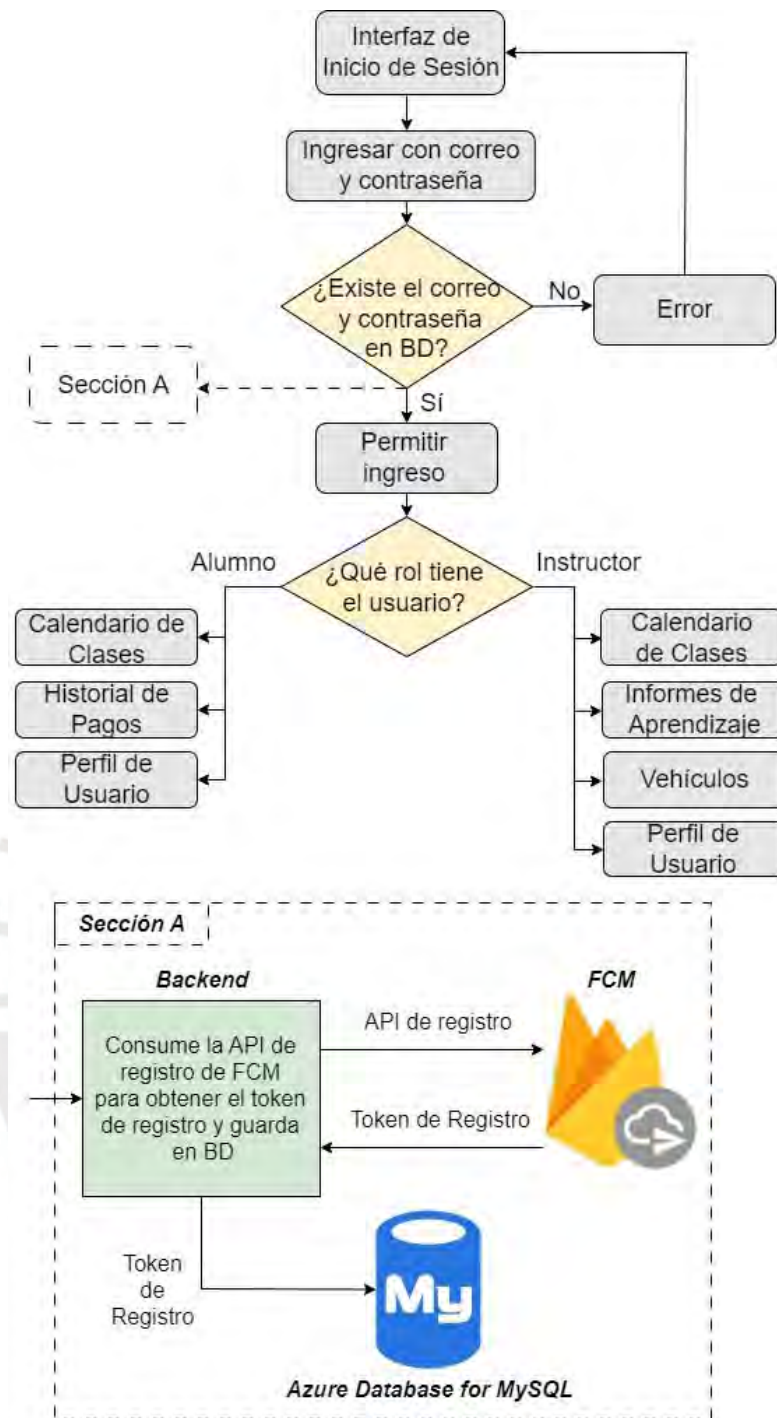


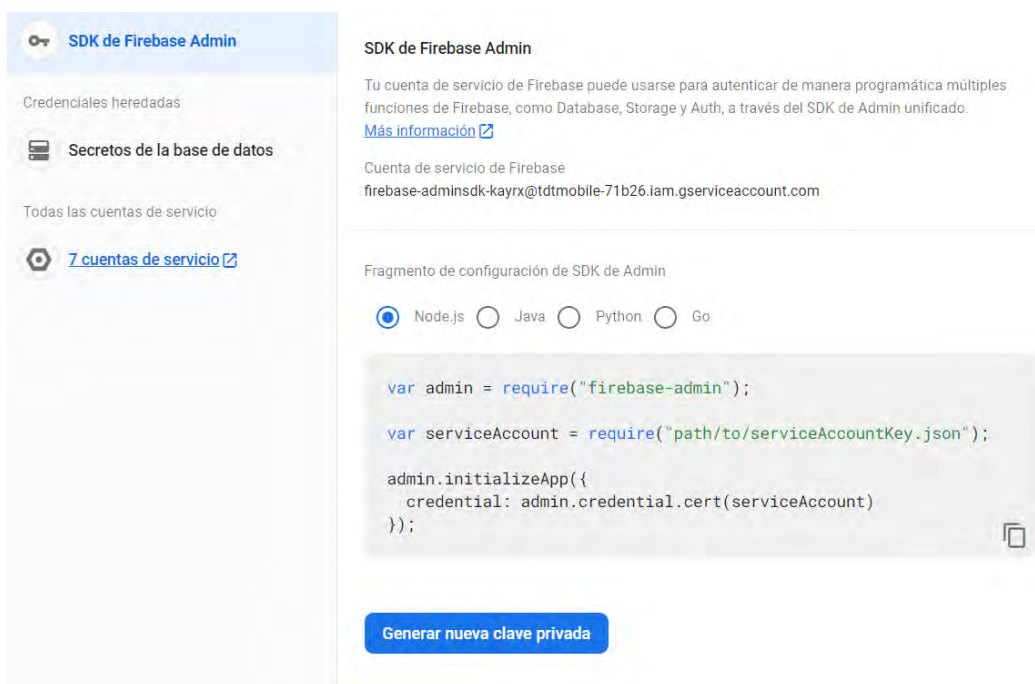
Figura 3.13. Flujo de Control de Acceso en la aplicación móvil  
Fuente: Elaboración propia

### 3.3.2.2 Envío de notificaciones 15 minutos antes de una clase programada

Como se mencionó en la Sección 3.2.6, para comenzar a utilizar FCM en la aplicación de Node.js, es esencial registrar la aplicación móvil en un proyecto de Firebase. Una vez que se haya realizado el registro, se debe dirigir a la sección de “Cuentas de Servicio”, como se



observa en la Figura 3.14. Aquí, se obtendrá un fragmento de configuración del SDK de Firebase Admin dependiendo de la tecnología utilizada, el cual es un conjunto de bibliotecas que permite interactuar con Firebase. Para finalizar esta configuración, se debe generar una nueva clave privada, el cual es un archivo “.json” que contendrá información relacionada al proyecto y otorgará acceso a los servicios de Firebase en nuestra aplicación, y deberá ser cargada en la aplicación.



*Figura 3.14. Sección de Cuentas de Servicio de un proyecto en Firebase  
Fuente: Elaboración propia*

Para que la aplicación se comunique con la Azure Database for MySQL, es fundamental configurar las credenciales establecidas durante la creación de este servicio. Asimismo, para permitir la conexión, se debe descargar el certificado SSL del servicio, el cual se encuentra en la sección de redes según la Figura 3.15, el cual sirve para autenticar al servidor de Azure Database for MySQL. Posteriormente, se debe cargar en la aplicación y añadirlo a la configuración, de acuerdo a la Figura 3.16, para asegurar una comunicación segura y autorizada.

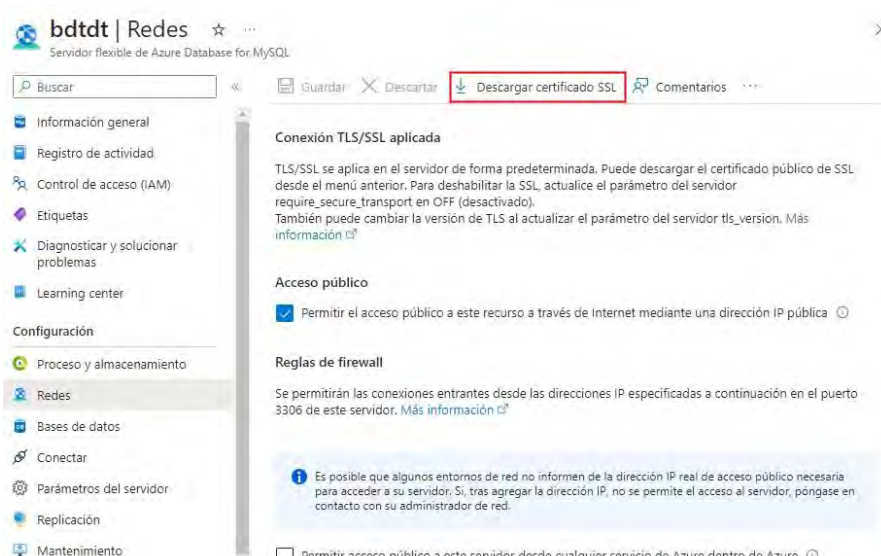


Figura 3.15. Sección de Redes de la Azure Database for MySQL  
Fuente: Elaboración propia

```
const mysql = require('mysql2');
const admin = require("firebase-admin");
const serviceAccount = require("./tdtmobile-71b26-firebase-adminsdk-kayrx-3e06ba9a15.json");
const cron = require('node-cron');
var fs = require('fs');
const path = require('path');
const certLocation = path.resolve(__dirname, "DigiCertGlobalRootCA.crt.pem");
const serverCa = [fs.readFileSync(certLocation, "utf8")];

/** Configuraciones*/
admin.initializeApp({
  credential: admin.credential.cert(serviceAccount)
});

let conn = mysql.createConnection({
  host: 'bdttdt.mysql.database.azure.com',
  user: 'alonsotesis',
  password: 'Tesis2023@',
  database: 'tdtback',
  ssl: {
    rejectUnauthorized: true,
    ca: serverCa
  }
});
```

Figura 3.16. Configuración de la Aplicación en Node.js para la comunicación con FCM y Azure Database for MySQL  
Fuente: Elaboración propia

Finalmente, se sigue el flujo descrito en la Figura 3.17, para cumplir R8. Se establece una tarea programada que se ejecutará cada 15 minutos, la cual consultará a la base de datos si existen usuarios, alumnos e instructores, que tienen alguna clase próxima en dicho tiempo. De ser así, se obtienen los tokens de registro de FCM de estos usuarios, como se mencionó en la Sección 3.2.6, el cual se registra cuando se inicia sesión en la aplicación móvil y se guarda en la base de

datos. Finalmente, se configura el mensaje de notificación y utilizando el SDK de Firebase Admin se realiza el envío del mensaje de notificación a los usuarios.

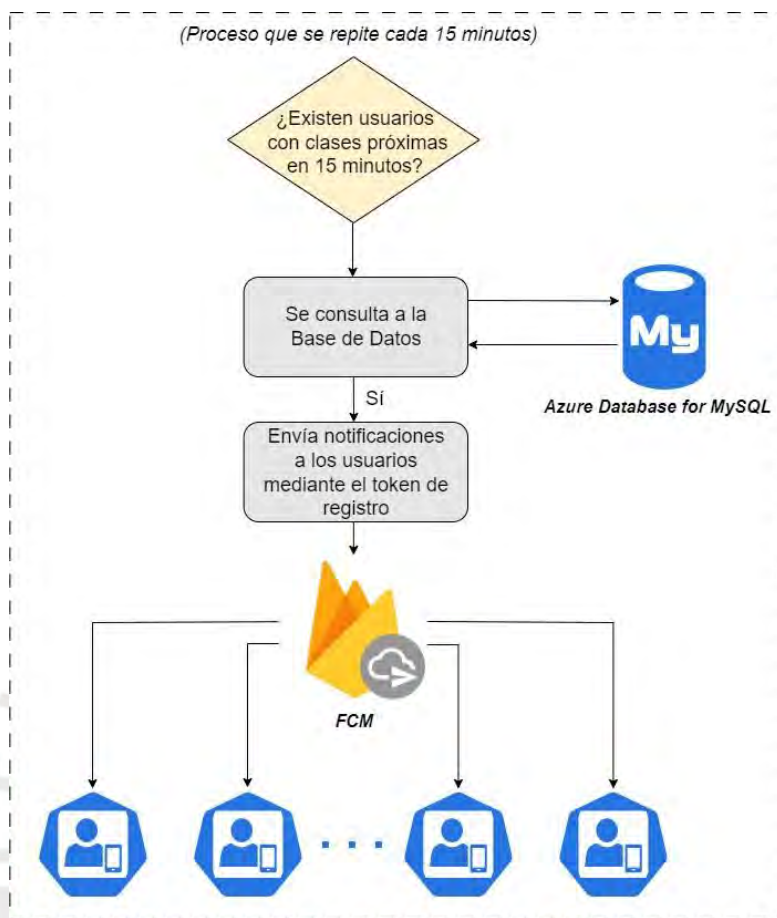


Figura 3.17. Flujo de operación para el envío de mensajes de notificación  
Fuente: Elaboración propia

### 3.4 Interfaces de las aplicaciones tecnológicas

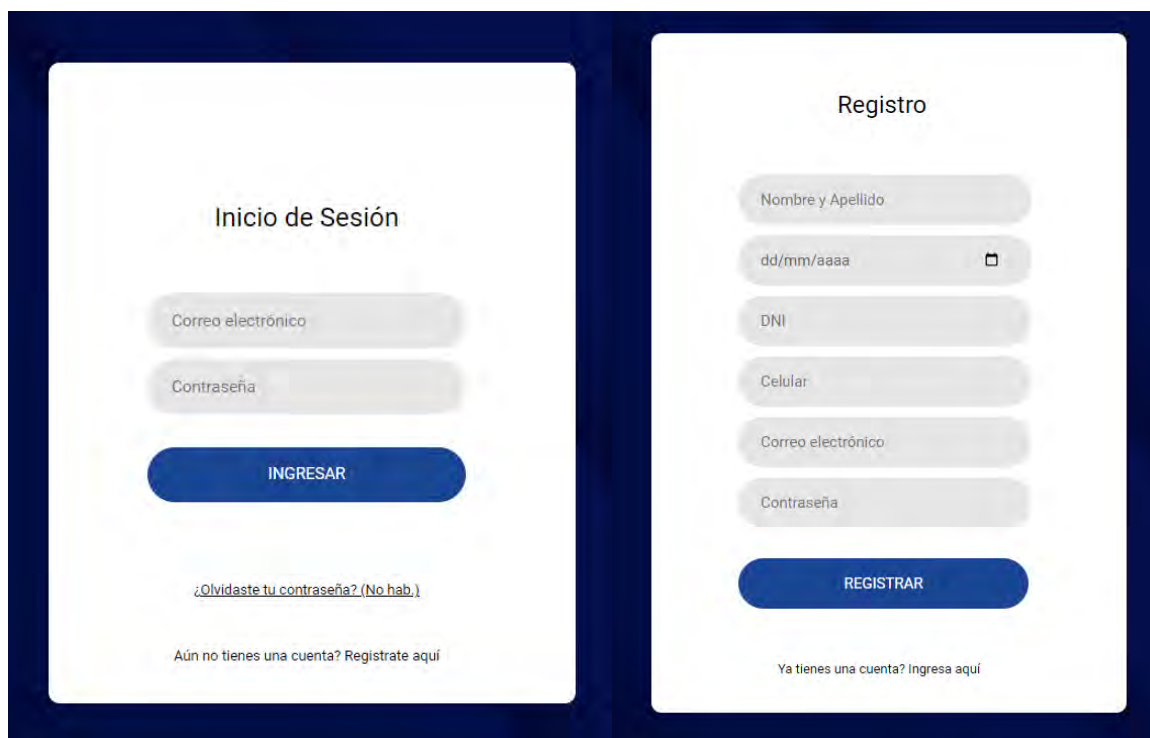
A continuación, se presentarán las principales interfaces desarrolladas en las aplicaciones, que tienen como principio el ser amigables y sencillas de usar.

#### 3.4.1 Interfaces de la aplicación web

##### 3.4.1.1 Interfaz de Inicio de Sesión y Registro

En la Figura 3.18 se muestran la interfaz de “Inicio de Sesión” y “Registro”. En la primera, los usuarios deben proporcionar su correo y contraseña para poder acceder a los servicios de la

aplicación. En caso de no tener una cuenta, tienen la opción de registrarse haciendo click en “Regístrate Aquí”, lo que los llevará a la interfaz de “Registro”. En esta última, los usuarios deberán ingresar sus datos personales como nombres y apellidos, DNI, celular, fecha de nacimiento, correo electrónico y contraseña, con el fin de crear su cuenta.

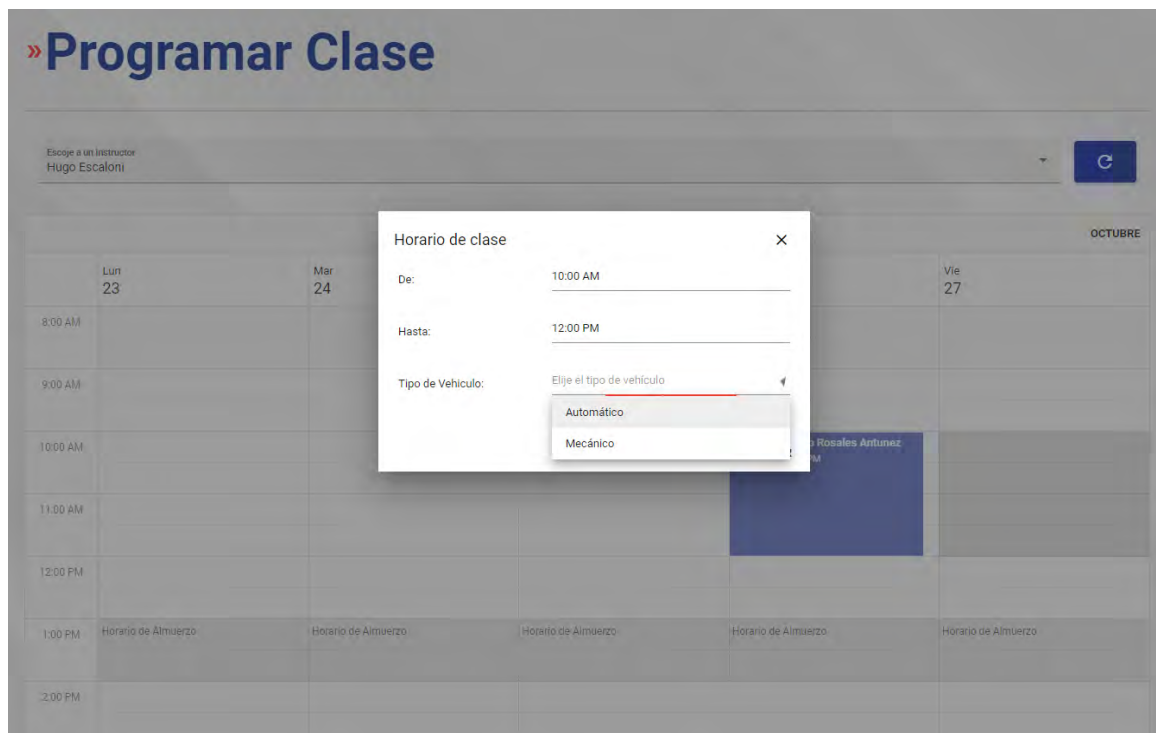


*Figura 3.18. Interfaz de “Iniciar sesión” y “Registro” de la aplicación web  
Fuente: Elaboración propia*

### 3.4.1.2 Interfaz de Programar Clase

En la interfaz de “Programar Clase”, tal como se visualiza en la Figura 3.19, los alumnos podrán seleccionar a los instructores registrados en la escuela de manejo para poder visualizar la disponibilidad de su horario. Los horarios están definidos de Lunes a Viernes de 8am a 8pm, con la posibilidad de ajustes según las políticas de la escuela de manejo. Una vez que hayan elegido un horario adecuado, deberán seleccionar el rango de horas en el que desean tomar su clase de manejo. A continuación, deberán escoger el tipo de vehículo que desean manejar durante la clase. Luego, se le redireccionará a una página de vista de pago que mostrará los

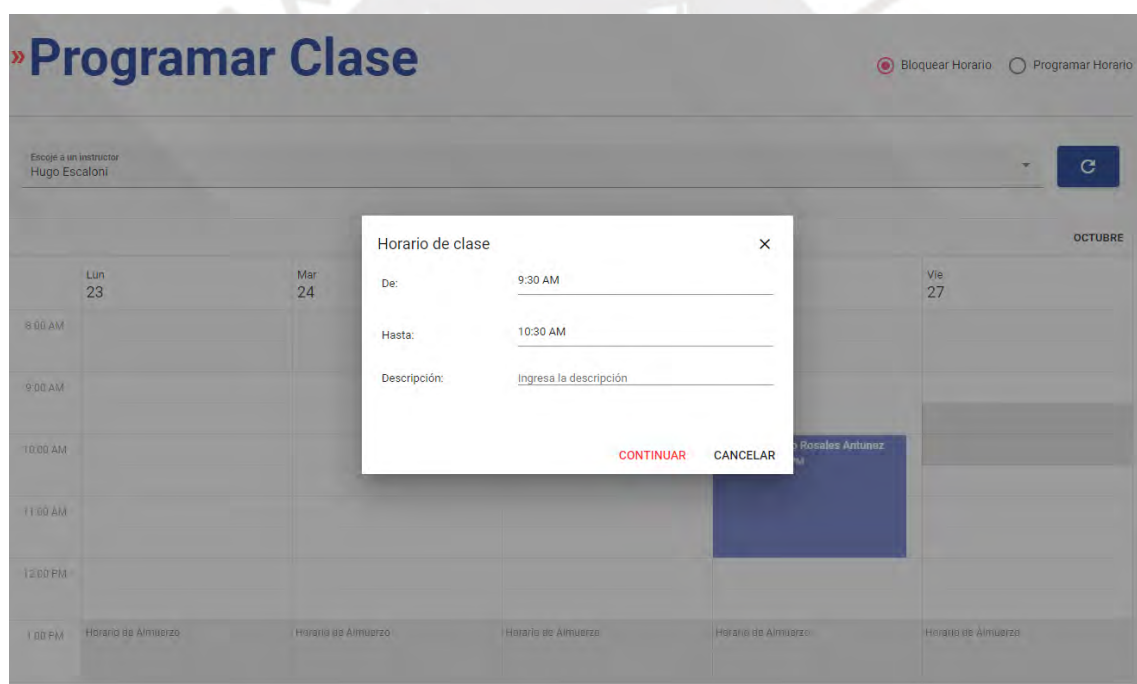
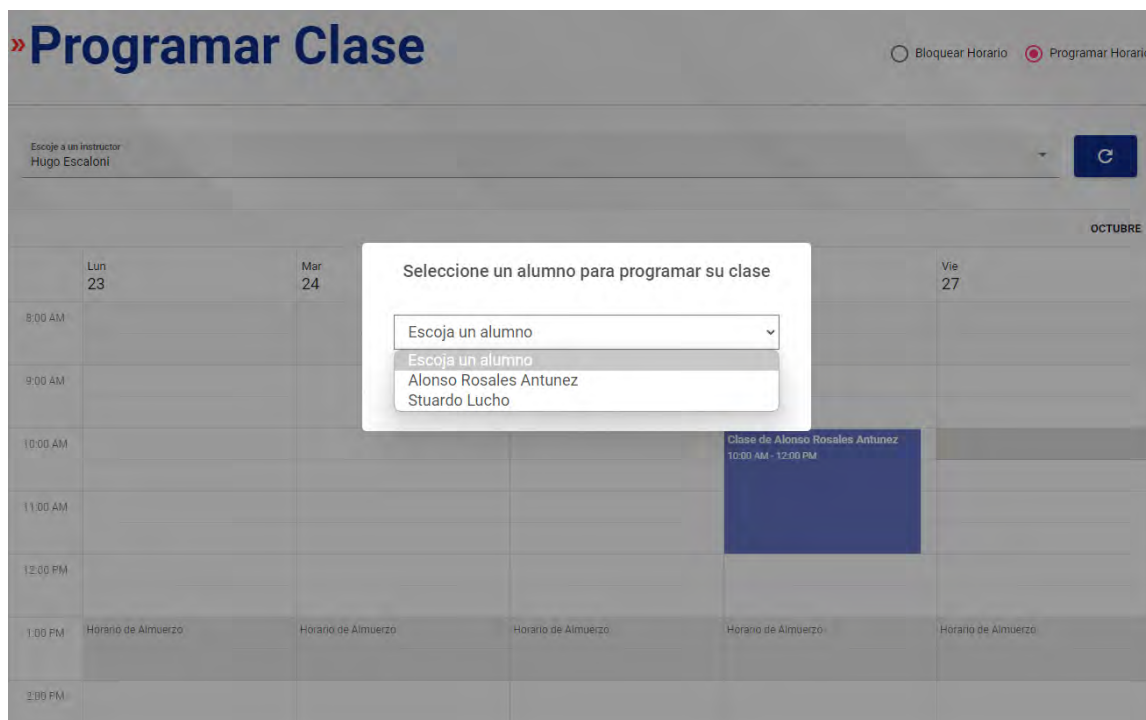
detalles del pago y el formulario para llevar a cabo la transacción, tal como se describió en la Sección 3.3.1.2. Una vez efectuado el pago, su clase quedará reservada, cumpliendo R2.



*Figura 3.19. Interfaz de “Programar Clase” de la aplicación web  
Fuente: Elaboración propia*

Por otro lado, los administradores, en la interfaz de “Programar Clase”, como se observa en la Figura 3.20, tendrán dos opciones: “Programar Horario” y “Bloquear Horario”. La opción de “Programar Horario” les permitirá realizar los mismos pasos que un alumno, excepto que no tendrán que pasar por la pasarela de pago. Después de elegir un horario de clase, deberán seleccionar al alumno para el cual reservarán la clase.

La opción de “Bloquear Horario” permitirá a los administradores bloquear ciertos rangos de hora en el horario de un instructor, como se define en R3. Esto impedirá que los alumnos reserven clases de manejo durante ese horario. Para llevar a cabo este bloqueo, el administrador deberá seleccionar al instructor y definir el rango de horas a bloquear, además de proporcionar una breve descripción del motivo de dicho bloqueo.



*Figura 3.20. Opciones de “Programar Horario” y “Bloquear Horario” para un administrador en la interfaz de “Programar Clase” de la aplicación web  
Fuente: Elaboración propia*

### 3.4.1.3 Interfaz de Informes de Aprendizaje

En la interfaz de “Informes de Aprendizaje”, como se observa en la Figura 3.21, los alumnos podrán acceder a los informes que se les han asignado por cada clase de manejo que han tomado.

Estos informes, como se detalló en R5, incluirán detalles de la clase, una descripción de lo que se realizó durante la misma, así como observaciones y comentarios de mejora por el instructor y dirigidos al alumno.



*Figura 3.21. Interfaz de “Informes de Aprendizaje” de la aplicación web  
Fuente: Elaboración propia*

#### **3.4.1.4 Interfaz de Balance**

En la interfaz de “Balance”, como se ve en la Figura 3.22, los administradores podrán visualizar la frecuencia de los instructores en relación a su cantidad de horas de clases y la frecuencia de los vehículos en relación a su cantidad de horas de uso y consumo de combustible y monto total de mantenimientos, contando con opciones de filtrado por día, semana o mes. También, se podrá observar una tabla con información acerca de los pagos realizados por los alumnos con capacidad con la opción de ser filtrado por día o semana o mes, código de operación, nombre o correo del usuario. Otra tabla se dedicará al seguimiento de los alumnos matriculados. Finalmente, tendrá la posibilidad de exportar toda esa información en un archivo .xlsx seleccionando el rango de fechas de la información a obtener.

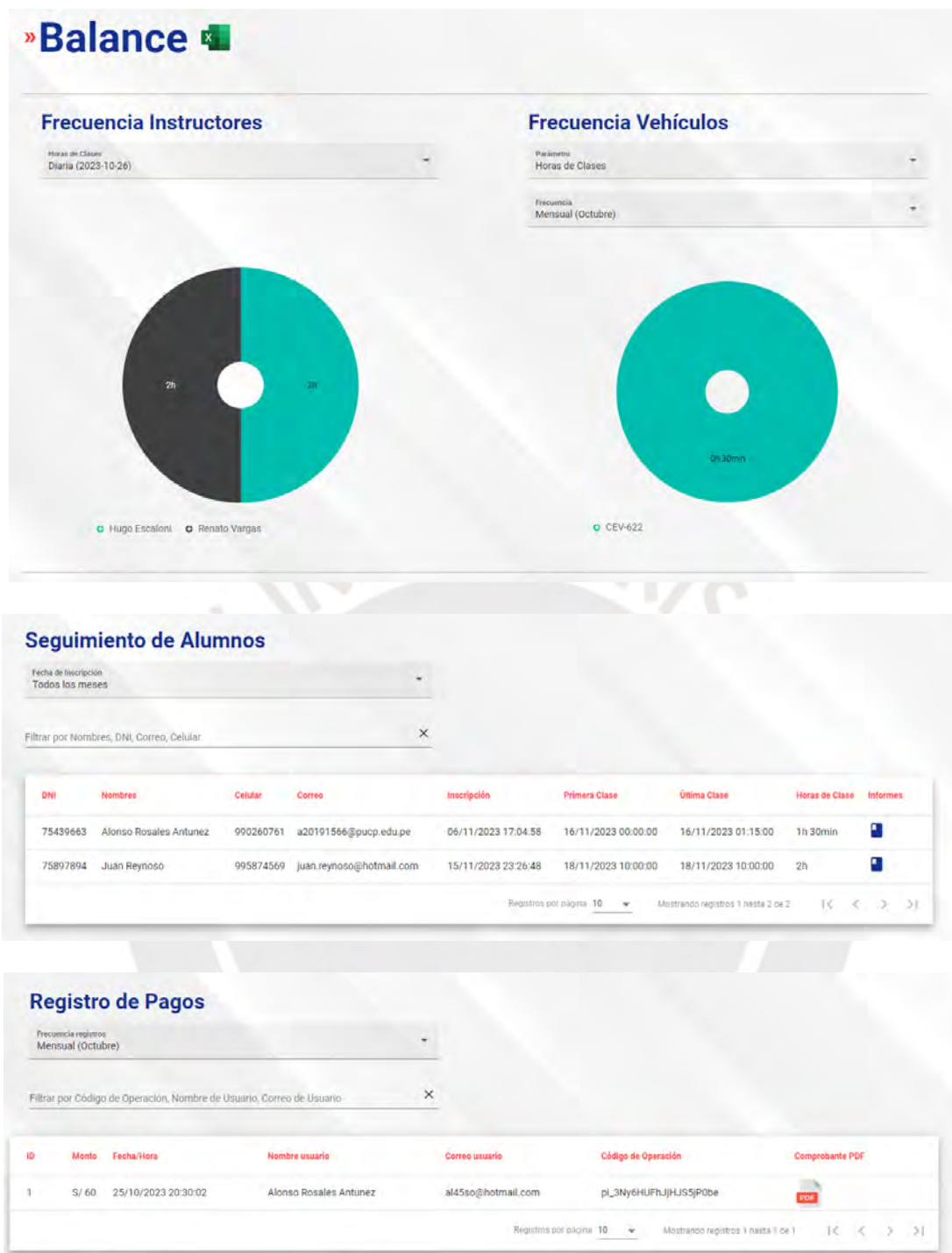


Figura 3.22. Interfaz de "Balance" de la aplicación web  
Fuente: Elaboración propia

### 3.4.2 Interfaces de la aplicación móvil

#### 3.4.2.1 Interfaz de Calendario de Clases

Después de iniciar sesión, se aplicará el control de acceso de los usuarios en función a su rol.

La interfaz de calendario de clases estará disponible tanto para los alumnos como para los



instructores, descrito en R9, pero la información que se muestra variará en función de su rol. Como se observa en la Figura 3.23 (imagen de la izquierda), los alumnos verán un listado de sus clases programadas, ordenadas de las más próximas a las más distantes, observándose detalles como el horario, el instructor y el tipo de vehículo que seleccionaron durante la programación de su clase. Por otro lado, los instructores, como se observa en la Figura 3.23 (imagen de la derecha), observarán un listado de sus clases programadas, también ordenadas de las más próximas a las más distantes, donde podrán ver el horario, el alumno que programó la clase y el tipo de vehículo seleccionado.



Figura 3.23. Interfaz de “Calendario de Clases” de la aplicación móvil  
Fuente: Elaboración propia

### 3.4.2.2 Interfaz de Informes de Aprendizaje

La interfaz de informes de aprendizaje sólo podrá ser accedida por los usuarios con rol de instructor. Como se visualiza en la Figura 3.24, los instructores observarán un listado de los alumnos que han tomado al menos una clase con él, contando con un buscador para poder

facilitar la búsqueda por su nombre. Al seleccionar un alumno, podrán visualizar los informes que el alumno tiene registrado, con capacidad de edición de aquellos a los cuales él ha creado. Asimismo tendrá la opción de agregar informes de aprendizaje, definido en R10, ingresando datos como el nombre del informe, el vehículo usado, el horario de clase impartida, la descripción de la clase y observaciones y comentarios de mejora dirigidos al alumno.

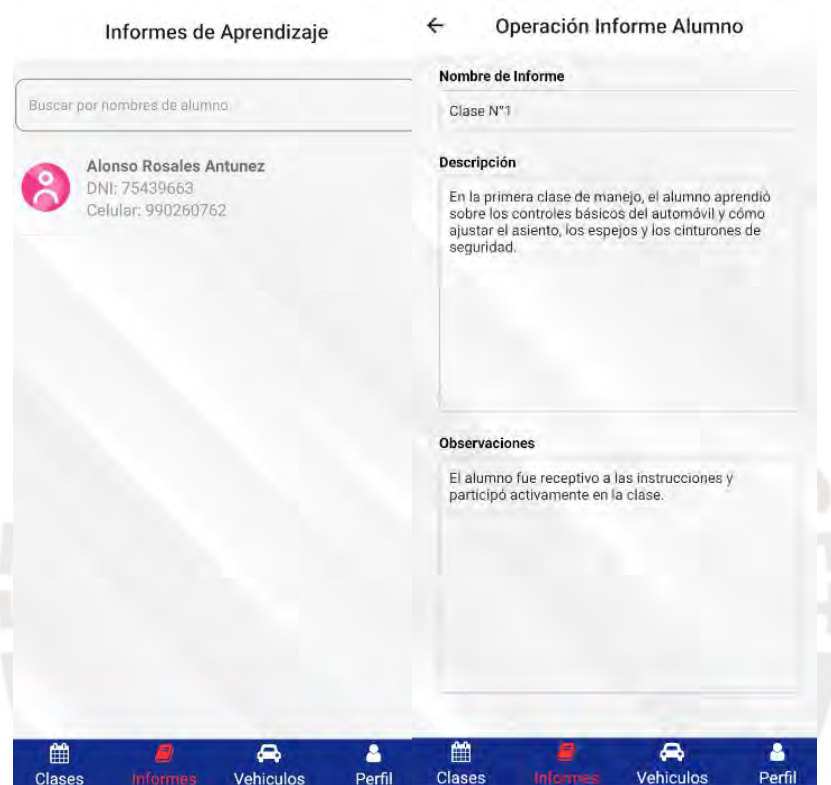


Figura 3.24. Interfaz de “Informes de Aprendizaje” de la aplicación móvil  
Fuente: Elaboración propia

## **Capítulo 4. Despliegue, pruebas y análisis económico**

En el presente capítulo, se llevará a cabo la presentación del despliegue de los componentes que se utilizarán en la nube de Microsoft Azure. También se mostrarán las pruebas efectuadas a la aplicación web y móvil para verificar el cumplimiento de los requerimientos establecidos en el capítulo 3. Finalmente, se llevará a cabo un análisis económico de la propuesta de solución.

### **4.1 Arquitectura de Despliegue**

A continuación, se procederá a realizar el despliegue de los diversos componentes de la arquitectura de diseño propuesta, como se detalló en la Sección 3.2. Para una gestión más eficiente, los servicios se organizarán en pequeños componentes modulares, como se visualiza en la Figura 4.1. Esto facilitará un seguimiento más claro durante el proceso de despliegue, permitiendo identificar con precisión cada componente.

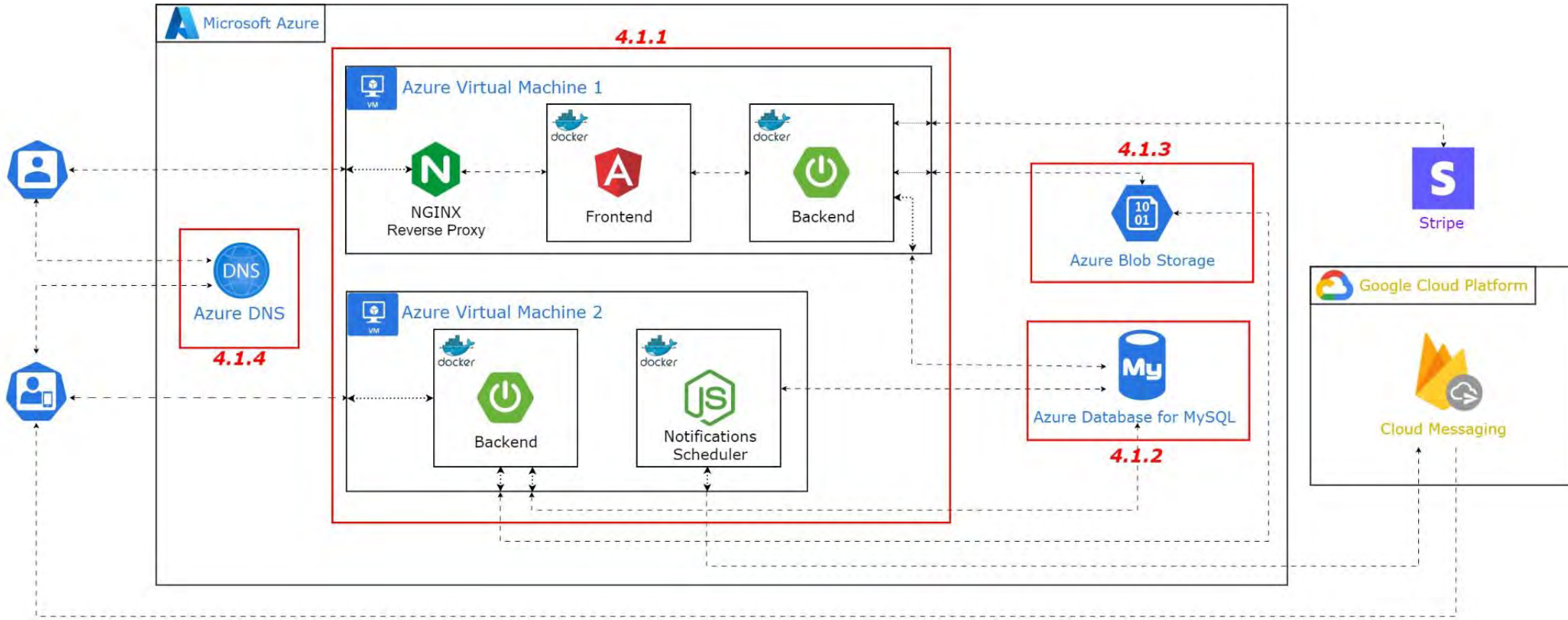


Figura 4.1. Despliegue de la Arquitectura de Diseño  
Fuente: Elaboración propia

### 4.1.1 Azure Virtual Machines

Las especificaciones técnicas de las dos máquinas virtuales alquiladas se pueden observar en la Tabla 4.1. Los recursos asignados son suficientes para soportar una concurrencia de 31 alumnos en ambas aplicaciones, cumpliendo lo definido en R6. Asimismo, de acuerdo a las pruebas de carga realizadas a las aplicaciones, se obtuvo que, con dichas características, se puede llegar a soportar hasta una concurrencia de 130 alumnos.

*Tabla 4.1. Especificaciones técnicas de las Azure Virtual Machines  
Fuente: Elaboración propia*

Tipo de instancia	Standard_B1s
vCPU	1
Memoria RAM	1 GiB
Almacenamiento	30 GiB
Sistema Operativo	Ubuntu 22.04 LTS

La configuración de red para los contenedores en la Azure Virtual Machine 1 se detalla en la Figura 4.2. El contenedor denominado “angular-app”, que aloja el frontend, está designado para recibir solicitudes externas a través del puerto 8080 de la máquina virtual. En contraste, el contenedor “spring-app”, que aloja el backend, utiliza el puerto 8081 de la máquina virtual, que será utilizado por el frontend para comunicarse con el backend.

```

root@vm1:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                                                                                     NAMES
e59995f593d0   angular:latest  "nginx -g 'daemon of_..." 2 weeks ago   Up 2 weeks   0.0.0.0:8080→80/tcp, :::8080→80/tcp                         angular-app
eddcf5392e13   spring:latest   "java -Djava.library_..." 2 weeks ago   Up 22 hours  0.0.0.0:8081→8081/tcp, :::8081→8081/tcp                     spring-app

```

*Figura 4.2. Configuración de red de contenedores de la Azure Virtual Machine 1  
Fuente: Elaboración propia*

Para la Azure Virtual Machine 2, la configuración de contenedores se detalla en la Figura 4.3. El contenedor “spring-app” se encarga de alojar el backend de la aplicación móvil y utiliza el

puerto 8081 de la máquina virtual para recibir solicitudes, mientras que el contenedor “notify-app” no presenta una configuración de red particular ya que opera de manera independiente.

```
root@vm2:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
a8f44c75ad07   spring:latest  "java -jar tdtback.j..." 2 weeks ago   Up 2 weeks   0.0.0.0:8081->8081/tcp, :::8081->8081/tcp   spring-app
e1fcd5962b84   notify:latest  "docker-entrypoint.s..." 4 weeks ago   Up 4 weeks                               notify-app
root@vm2:~#
```

*Figura 4.3 Configuración de red de contenedores de la Azure Virtual Machine 2*  
Fuente: Elaboración propia

Asimismo, el archivo de configuración Dockerfile para la construcción y despliegue de imágenes Docker de los contenedores de las Azure Virtual Machines 1 y 2 se encuentran en el Anexo 1 y 2 respectivamente.

#### 4.1.2 Azure Database for MySQL

Las especificaciones técnicas del Azure Database for MySQL alquilado se pueden apreciar en la Tabla 4.2. Los recursos asignados son suficientes para la escritura y almacenamiento de la información de los 31 alumnos concurrentes que acceden a la plataforma.

*Tabla 4.2. Especificaciones técnicas del Azure Database for MySQL*  
Fuente: Elaboración propia

Tipo de instancia	Standard_B1s
vCPU	1
Memoria RAM	1 GiB
Almacenamiento	20 GiB

#### 4.1.3 Azure Blob Storage

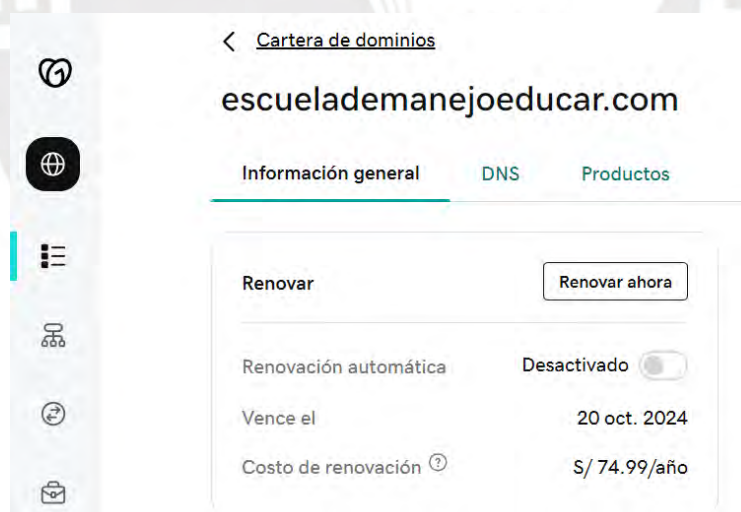
Para almacenar imágenes y documentos en formato PDF, se empleará el servicio Azure Blob Storage. La creación de los contenedores se realizará acorde a la estructura detallada en la Figura 3.4, asegurando un almacenamiento organizado de los archivos.

Nombre	Última modificación	Nivel de acceso anó...	Estado de concesión
<input type="checkbox"/> \$logs	19/10/2023, 3:22:09 ...	Privada	Disponible ***
<input type="checkbox"/> archivos	15/11/2023, 11:08:45...	Contenedor	Disponible ***
<input type="checkbox"/> estaticos	15/11/2023, 11:08:45...	Contenedor	Disponible ***
<input type="checkbox"/> feedback	15/11/2023, 11:08:45...	Contenedor	Disponible ***
<input type="checkbox"/> imagenes	15/11/2023, 11:08:45...	Contenedor	Disponible ***
<input type="checkbox"/> instructores	22/11/2023, 2:37:32 ...	Contenedor	Disponible ***
<input type="checkbox"/> vehiculos	15/11/2023, 11:08:45...	Contenedor	Disponible ***

*Figura 4.4. Creación de Contenedores para el almacenamiento de archivos en Azure Blob Storage  
Fuente: Elaboración propia*

#### 4.1.4 Azure DNS

Para la configuración del servicio Azure DNS, primero se debe realizar la selección y compra de un dominio mediante un proveedor de dominios, en este caso se utilizará GoDaddy. Se debe verificar la disponibilidad del dominio deseado y, si está libre, se procede con la adquisición, que otorga derechos de uso por un año con la opción de renovación.



*Figura 4.5. Dominio adquirido en GoDaddy  
Fuente: Elaboración propia*

Después de lo mencionado anteriormente, para delegar la gestión del DNS del dominio adquirido a Microsoft Azure, es necesario configurar los servidores DNS proporcionados por el servicio Azure DNS en el apartado de “Administración de DNS” de GoDaddy.

Microsoft Azure

Inicio > escuela**demanejoeducar.com** Zona DNS

Conjunto de registros Zona secundaria Importar Exportar

Información esencial Vista JSON

Grupo de recursos (mover) : [tdtservices](#)

Suscripción (mover) : [Azure for Students](#)

Id. de suscripción : fdd48ead-c29b-47ae-a4e9-a491aa7...

Servidor DNS 1	: ns1-38.azure-dns.com.
Servidor DNS 2	: ns2-38.azure-dns.net.
Servidor DNS 3	: ns3-38.azure-dns.org.
Servidor DNS 4	: ns4-38.azure-dns.info.

Etiquetas (editar) : [Agregar etiquetas](#)

Cartera de dominios escuela**demanejoeducar.com**

Información general DNS Productos

Registros DNS Reenvío Servidores de nombres DNS Premium Nombres de host Registros

Los **servidores de nombres** determinan dónde está alojado tu DNS y dónde agregas, editas o eliminas los registros de DNS.

Uso de nombres de servidor personalizados [Cambiar servidores de nombres](#)

Servidores de nombres

- ns1-38.azure-dns.com
- ns2-38.azure-dns.net
- ns3-38.azure-dns.org
- ns4-38.azure-dns.info

Figura 4.6. Configuración de los Servidores DNS de Azure en GoDaddy  
Fuente: Elaboración propia

## 4.2 Pruebas Iniciales

Al desplegar la arquitectura de solución propuesta se pudo experimentar demoras significativas en la carga del contenido visual de la aplicación web, como se observa en la Figura 4.7.

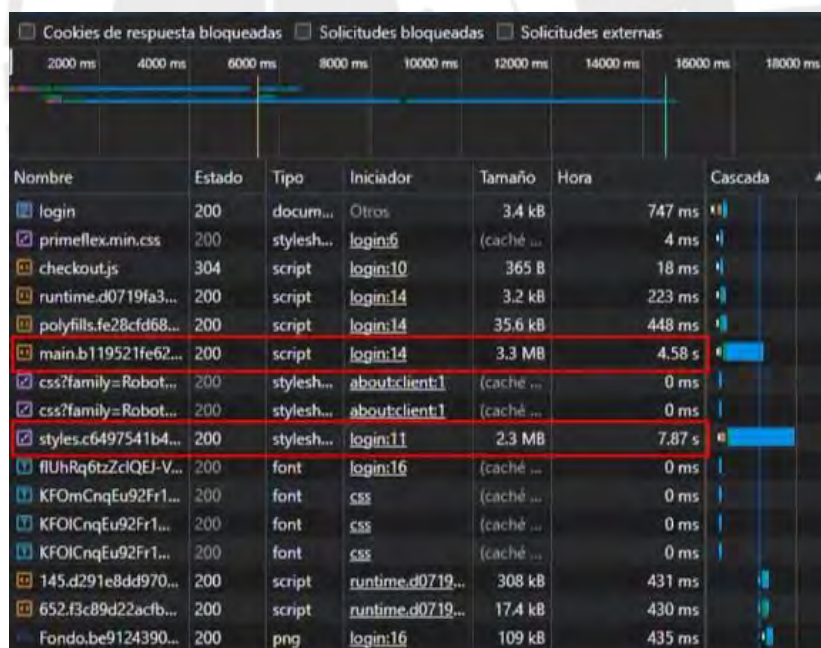


Figura 4.7. Tamaño y Tiempo de descarga de los archivos estáticos del Frontend  
Fuente: Elaboración propia

Esto se debía al tamaño considerable de los archivos estáticos generados durante la compilación del frontend. En particular, el archivo “main.js”, que contiene todas las funciones JavaScript del proyecto, presentaba un tamaño aproximado de 3.5 MB. Asimismo, el archivo “styles.css”,



que contiene todos los estilos del proyecto, alcanzaba a tener unos 2.1 MB. Como resultado, cada vez que una vista de la aplicación web era cargada, se volvía a descargar todos los archivos compilados del frontend, lo que ocasionaba un tiempo de espera de hasta 20 segundos, dependiendo de la velocidad de internet del usuario, para la carga completa del contenido web, afectando a la experiencia de usuario.

### **4.3 Optimización implementada**

En primer lugar, para solucionar el problema en la demora de la carga visual del contenido de la aplicación web, se propuso optimizar las hojas de estilos utilizados en el frontend, reduciendo código repetitivo e innecesario y eliminando importaciones innecesarias de los paquetes instalados en el “node\_modules”; sin embargo, el tamaño de la hoja de estilos no generó cambio considerable por lo que el tiempo de carga no se vió afectado. Es por ello, que se decidió utilizar estrategias de caché para así evitar la recarga completa de los archivos estáticos en cada visita, aplicando el almacenamiento en caché del navegador. Para esto se utilizó el servicio de Azure CDN (Content Delivery Network), el cual es un servicio de distribución de contenido que almacena dicho contenido en caché en servidores que se encuentran cerca a los usuarios permitiendo mejorar el tiempo de carga del contenido web [58]. Finalmente, luego de realizar esta mejora, se reduce significativamente el tiempo de descarga de los archivos estáticos mencionados a milisegundos, por debajo de un segundo, como se observa en la Figura 4.8.

Nombre	Estado	Tipo	Iniciador	Tama...	Hora	Cascada
login	200	docu...	Otros	3.5 kB	155 ms	
primeflex.min.css	200	styles...	login:7	(cach...	6 ms	
css?family=Robot...	200	styles...	about:client:1	(cach...	0 ms	
runtime.ef8a1d78...	200	script	login:14	(cach...	4 ms	
css?family=Robot...	200	styles...	about:client:1	(cach...	0 ms	
polyfills.fe28cfd6...	200	script	login:14	(cach...	4 ms	
main.ca59390324...	200	script	login:14	(cach...	40 ms	
styles.ec959901a...	200	styles...	login:11	(cach...	24 ms	
flUhRq6tzZclQEJ...	200	font	login:16	(cach...	0 ms	
KFOmCnqEu92Fr...	200	font	css	(cach...	0 ms	
KFOICnqEu92Fr1...	200	font	css	(cach...	0 ms	
KFOICnqEu92Fr1...	200	font	css	(cach...	0 ms	
145.a6d6a500f4ff...	200	script	runtime.ef8...	383 kB	714 ms	
652.91c054658ae...	200	script	runtime.ef8...	19.3 kB	293 ms	
Fondo.png	200	png	login:16	(cach...	0 ms	

Figura 4.8. Reducción en el tiempo de descarga de los archivos estáticos del Frontend  
Fuente: Elaboración propia

#### 4.4 Arquitectura Final de Despliegue

Tras la optimización realizada, se ha añadido un componente adicional a la arquitectura de despliegue, como se muestra en la Figura 4.9. Este nuevo componente se sitúa estratégicamente después del Azure DNS. El propósito es que las consultas se dirijan primero a este componente antes de proceder hacia la Azure Virtual Machine, en relación a los archivos estáticos.

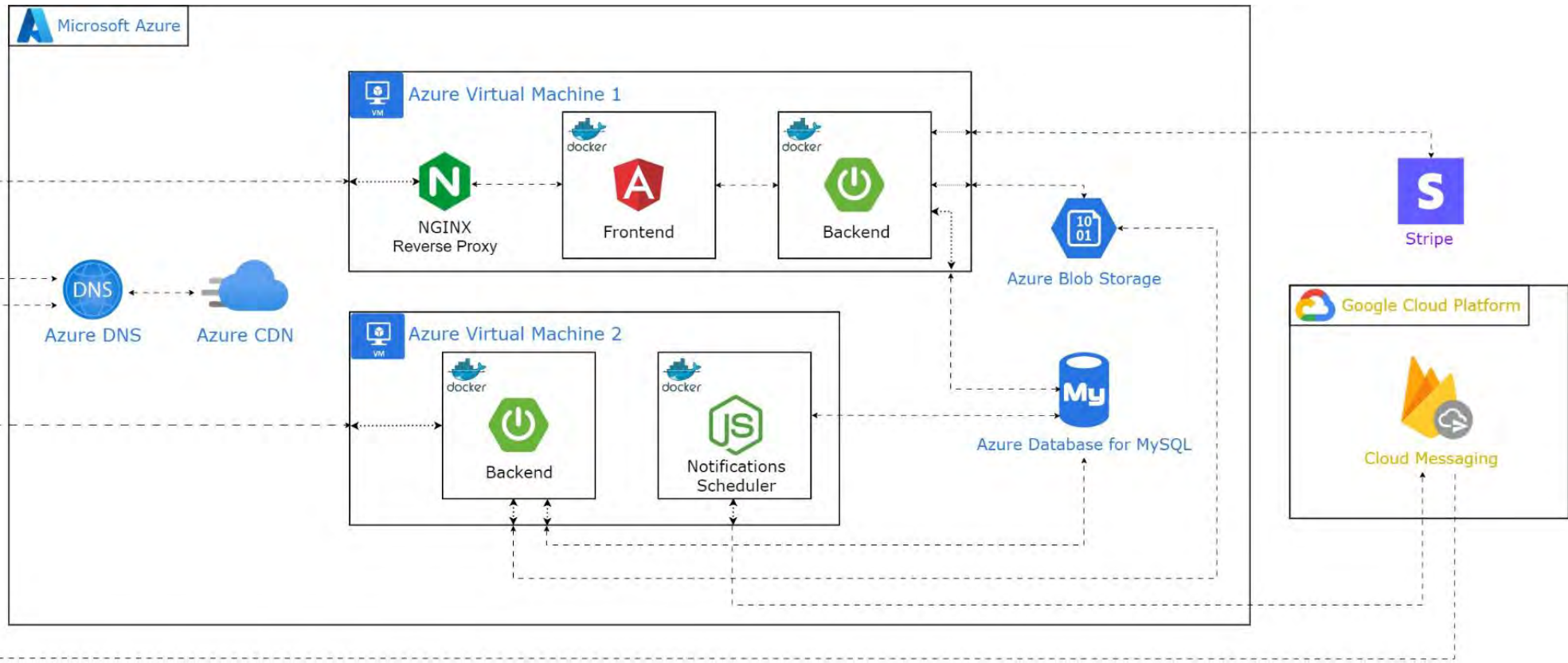


Figura 4.9. Arquitectura Final de Despliegue  
Fuente: Elaboración propia

## 4.5 Pruebas Funcionales de las Aplicaciones

A continuación, se presentarán las pruebas ejecutadas a las funcionalidades más destacables de las aplicaciones desarrolladas.

### 4.5.1 Aplicación Web

#### 4.5.1.1 Reserva de Clase de Manejo

Como se detalla en la Sección 3.4.1.2, después de elegir el horario de clase, el instructor y el tipo de vehículo, el usuario es redirigido a una página de pago en línea para concretar el pago de la clase de manejo, ilustrado en la Figura 3.11. Una vez completada la transacción, se muestra un mensaje confirmando el éxito de la operación, como se muestra en la Figura 4.10. Finalmente, se visualiza la clase reservada en el calendario del instructor elegido, como puede observarse en la Figura 4.11, cumpliendo con R2.

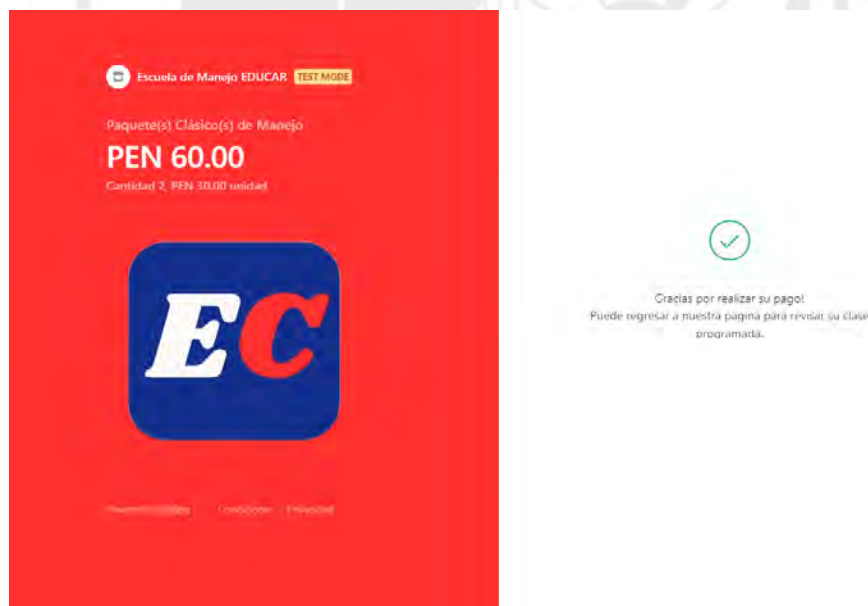


Figura 4.10. Operación exitosa en la Página de pago en línea  
Fuente: Elaboración propia

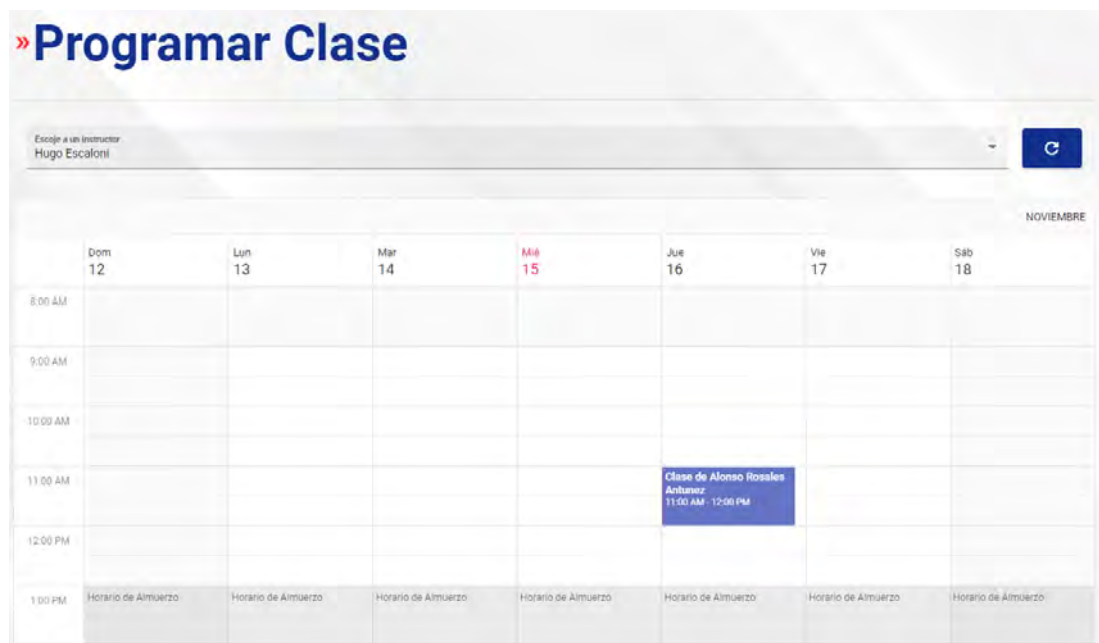



Figura 4.11. Visualización de clase reservada en el calendario del instructor elegido  
Fuente: Elaboración propia

#### 4.5.1.2 Registros de Operaciones en un Vehículo

Para presentar de forma organizada la información sobre las recargas de combustible y los mantenimientos de un vehículo en particular, descritos en R4 y mostrados en la Figura 4.12 y Figura 4.13, se emplea el componente “MatTable” de Angular Material.

ID Operación	Fecha y Hora de la Operación	Cantidad (galones)	Monto total de la Operación	Acción
16	15/11/2023 23:23:36	2	S/ 55.7	Editar
15	25/10/2023 15:12:29	1 galón	S/ 6.7	Editar
14	25/10/2023 15:11:40	3 galones	S/ 70.9	Editar

Figura 4.12. Registro de recargas de combustible de un vehículo  
Fuente: Elaboración propia

» **Mantenimientos** 

ID Operación	Fecha y Hora de la Operación	Taller	Mecánico	Descripción de la Operación	Monto total de la Operación	Acción
12	24/10/2023 22:00:38	Autoland	Victor Santivañez	Cambio de aceite y filtro del motor para mantener el motor funcionando correctamente.	S/ 120	<a href="#">Editar</a>
13	25/10/2023 15:16:37	Toyota	José Pelaez	Revisión de los frenos para garantizar la seguridad del vehículo.	S/ 250	<a href="#">Editar</a>
14	25/10/2023 15:16:58	Toyota	Juan Mancilla	Revisión de los neumáticos para garantizar la seguridad y el rendimiento del vehículo.	S/ 80	<a href="#">Editar</a>

Operaciones por página 10    Mostrando operaciones 1 hasta 3 de 3

Figura 4.13. Registro de mantenimientos realizados de un vehículo  
Fuente: Elaboración propia

#### 4.5.1.3 Descarga de Balance de Operaciones

Para descargar la información del Balance de Operaciones, filtrada por un rango de fechas, como puede observarse en la Figura 4.14, se hace uso de la dependencia Apache Poi. Esta herramienta es una dependencia que posibilita la creación de archivos .xlsx en Spring, facilitando así el cumplimiento de R7.

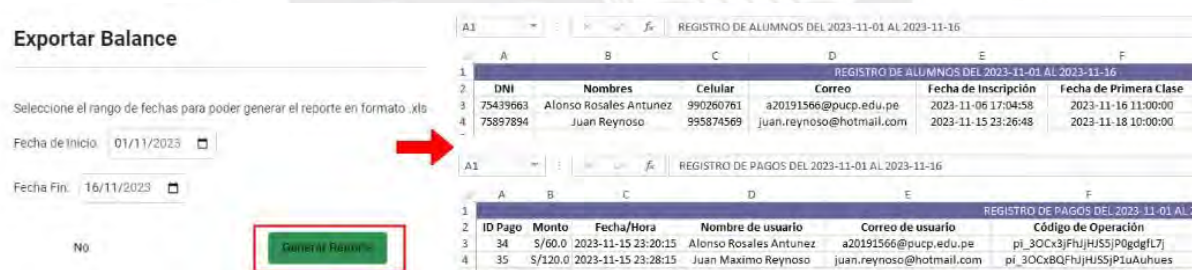
**Exportar Balance**

Seleccione el rango de fechas para poder generar el reporte en formato .xls

Fecha de Inicio: 01/11/2023

Fecha Fin: 16/11/2023

No [Generar Reporte](#)



DNI	Nombres	Celular	Correo	Fecha de Inscripción	Fecha de Primera Clase
75439663	Alonso Rosales Antunez	990260761	a20191566@puccp.edu.pe	2023-11-06 17:04:58	2023-11-16 11:00:00
75897894	Juan Reynoso	955874569	juan.reynoso@hotmail.com	2023-11-15 23:26:48	2023-11-18 10:00:00

ID Pago	Monto	Fecha/Hora	Nombre de usuario	Correo de usuario	Código de Operación
34	S/60.0	2023-11-15 23:20:15	Alonso Rosales Antunez	a20191566@puccp.edu.pe	pi_30Cx3[FhJ]H[SS]P0gdg[L7]
35	S/120.0	2023-11-15 23:28:15	Juan Maximo Reynoso	juan.reynoso@hotmail.com	pi_30Cx3B[FhJ]H[SS]P1uAuhues

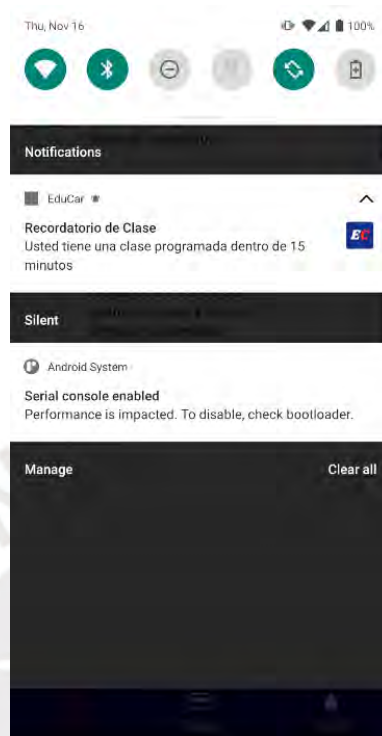
Figura 4.14. Descarga de Balance de Operaciones en archivo .xlsx  
Fuente: Elaboración propia

## 4.5.2 Aplicación Móvil

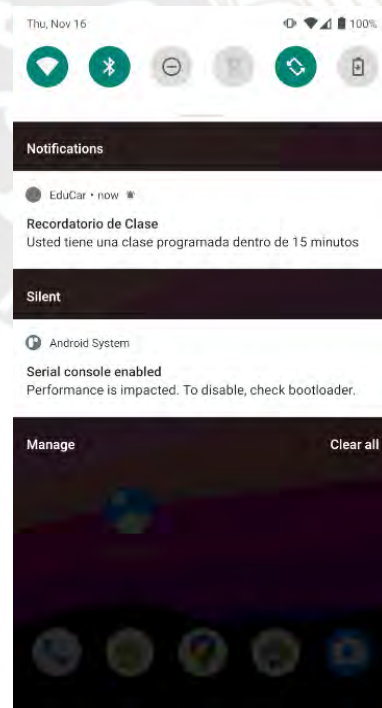
### 4.5.2.1 Recordatorio de Clase de Manejo Programada

En el caso de R8, la notificación como recordatorio para una clase de manejo programada, que se envía 15 minutos antes de su inicio, puede ser entregada de dos maneras distintas. Primero, si la aplicación está activa en primer plano, la notificación aparecerá como se muestra en la

Figura 4.15. Segundo, si la aplicación está cerrada o en segundo plano, la notificación se recibirá como se muestra en la Figura 4.16.



*Figura 4.15. Notificación recibida cuando la aplicación móvil está en primer plano  
Fuente: Elaboración propia*



*Figura 4.16. Notificación recibida cuando la aplicación móvil está cerrada o en segundo plano  
Fuente: Elaboración propia*

#### 4.5.2.2 Creación de Informe de Aprendizaje de un alumno

La Figura 4.17 demuestra la exitosa generación de un informe de aprendizaje para un alumno específico, en línea con lo establecido en R10. Para confirmar la correcta creación de dicho informe, se hace el seguimiento del alumno a través de la aplicación web. Este proceso que cumple con R5, se detalla en la Figura 4.18.

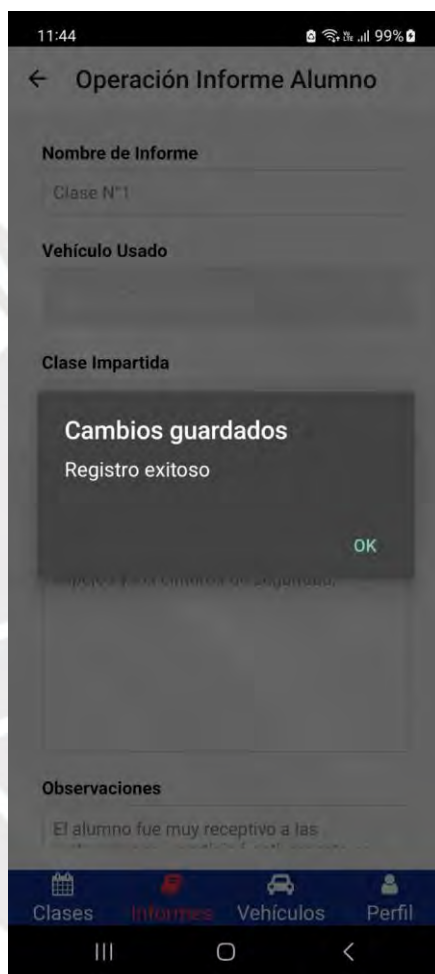


Figura 4.17. Creación exitosa de un informe de aprendizaje de un alumno  
Fuente: Elaboración propia





*Figura 4.18. Visualización del informe de aprendizaje creado desde la Aplicación Web  
Fuente: Elaboración propia*

## 4.6 Análisis Económico

Para realizar el análisis de costos del proyecto, se dividirán en 2 tipos: los costos de mano de obra para el desarrollo de software, que se clasifican como CAPEX debido a la naturaleza de inversión única, y los costos operativos mensuales por el uso de los servicios en la nube, clasificados como OPEX. Se debe mencionar que, a la fecha de redacción de la tesis, el tipo de cambio de dólares a soles es de S/.3.76.

### 4.6.1 Costos de Mano de Obra

La Tabla 4.3 detalla al responsable y el costo por hora de la mano de obra en soles para la implementación del proyecto. Esta estimación de costos laborales se realizó bajo la premisa de que el equipo está compuesto únicamente por un estudiante con un salario de practicante mensual, que es S/. 1025.00. Para calcular el costo por hora, se divide este salario mensual entre el total de horas trabajadas en el mes, que son 120 horas, correspondiente a una jornada semanal de 30 horas durante 4 semanas.

Tabla 4.3. Costos de Mano de Obra  
Fuente: Elaboración propia

Responsable	Precio por hora (S/)
Tesista	9

La cotización de la mano de obra, en función al dato proporcionado en la Tabla 4.3, se presenta a continuación:

Tabla 4.4. Cotización de Mano de Obra  
Fuente: Elaboración propia

Proceso	Rol	Responsable	Días	Horas/día	Horas Totales	Precio/hora (S/)	Subtotal (S/)
Diseño de la Solución	Diseñador	Tesista	5	2	10	9	90
Construcción de la Base de Datos	Desarrollador	Tesista	3	3	9	9	81
Desarrollo de la Aplicación Web	Desarrollador	Tesista	55	7	385	9	3465
	QA	Tesista	10	5	50	9	450
Desarrollo de la Aplicación Móvil	Desarrollador	Tesista	10	7	70	9	630
	QA	Tesista	3	5	15	9	135
<b>Monto Total por Mano de Obra (S/)</b>							<b>S/.4.851,00</b>

En conclusión, el costo total de CAPEX es de S/.4.851,00 soles.

#### 4.6.2 Costos de Operación

Para calcular el costo operativo (OPEX) de la propuesta de solución, se utilizará la calculadora de precios de Microsoft Azure [59], la cual permitirá determinar las tarifas asociadas a cada

servicio utilizado. Es importante mencionar que el servicio de Firebase Cloud Messaging no tiene costo alguno.

#### **4.6.2.1 Costo de la Azure Virtual Machine**

Las máquinas virtuales ofrecidas bajo el esquema “Pay as you go” tienen un costo de 0.0104 USD por hora de uso. Multiplicando esta tarifa por las 730 horas que Azure toma como referencia para un mes, el costo es  $730 \times 0.0104 = 7.59$  USD mensuales.

#### **4.6.2.2 Costo de la Azure Database for MySQL**

El servicio Azure Database for MySQL tiene un costo aproximado de 0.0085 USD por hora de uso. Entonces, para un mes, el costo es  $730 \times 0.0085 = 6.20$  USD mensuales. Además, el almacenamiento tiene un costo de 0.115 USD por GiB, resultando en  $20 \times 0.115 = 2.30$  USD mensuales. Finalmente, el costo total por el servicio es de 8.50 USD mensuales.

#### **4.6.2.3 Costo de Azure DNS**

El dominio “escuelademanejoeducar.com”, adquirido a través de GoDaddy, incurre en un costo anual de 19.94 USD, es decir, 1.66 USD mensuales. Adicionalmente, la zona DNS asociada tiene un costo de 0.50 USD por mes. Entonces, se tiene un costo total de 2.16 USD mensuales.

#### **4.6.2.4 Costo de Azure Blob Storage**

El costo del servicio de Azure Blob Storage se determina a partir del número total de operaciones de lectura/escritura, así como de los GB almacenados y descargados. El cálculo del almacenamiento en GB se basa en el promedio de usuarios, instructores, vehículos y clases al mes, datos que se encuentran de manera gráfica en la Sección 1.2. Se estableció que cada clase puede tener hasta tres imágenes de feedback. Utilizando estos datos, se estimó el peso total (en

GB) de los archivos almacenados, detallado en la Tabla 4.5, donde se considera que el peso promedio de un archivo PDF es de 29,26 KB y el tamaño máximo de cada imagen es de 2MB.

*Tabla 4.5. Estimación de GB almacenados en el servicio de Azure Blob Storage al mes  
Fuente: Elaboración propia*

Azure Blob Storage	Peso / Archivo	Archivo Totales	Peso Total
Contenedor Archivos	29,26 KB	218	6,37868 MB
Contenedor Estáticos	-	-	6,13983 MB
Contenedor Feedback	2 MB	654	1,308 GB
Contenedor Imágenes	-	-	1,48207 MB
Contenedor Instructores	2 MB	3	6 MB
Contenedor Vehículos	2 MB	3	6 MB
<b>PESO TOTAL (GB)</b>			<b>1,33400058 GB</b>

Utilizando esta información, se puede estimar el costo mensual del servicio. Las operaciones de escritura se basan en la suma de los datos promedio proporcionados por las escuelas de manejo, mientras que las operaciones de lectura dependen del número de usuarios promedio. El costo mensual del servicio se presenta en la Tabla 4.6.

*Tabla 4.6. Estimación de costo mensual del servicio Azure Blob Storage  
Fuente: Elaboración propia*

Azure Blob Storage	Cantidad	Costo (\$) x 10000 operaciones	Total (\$) / mes
Operaciones Escritura	878	0,065	0,005707
Operaciones Lectura	27218	0,005	0,013609
GB almacenados	-	-	0,03
GB descargados	-	-	-
<b>COSTO TOTAL (\$)</b>			<b>\$0,05</b>
<b>COSTO TOTAL (S/)</b>			<b>S/.0,19</b>

#### 4.6.2.5 Costo de Azure CDN

El costo del servicio de Content Delivery Network se calcula a partir de la cantidad de datos (en GB) transferidos y la zona geográfica desde la cual se servirán estos datos. El consumo de datos por usuario depende del tamaño total de los archivos estáticos almacenados en el Azure Blob Storage, como se muestra en la Tabla 4.7.

*Tabla 4.7. Estimación de datos transferidos a 1 usuario  
Fuente: Elaboración propia*

Azure CDN	Peso de Contenido
Contenedor Estáticos	6,13983 MB
Contenedor Imágenes	1,48207 MB
<b>PESO CONTENIDO (GB)</b>	0,0076219 GB

Determinando el volumen total de datos transferidos (en GB) a los usuarios promedio de la aplicación y seleccionando la zona geográfica correspondiente, en este caso, la Zona 3 (Brasil) que es la más cercana a Perú según las categorías que brinda Azure, se calcula el costo mensual del servicio, como se muestra en la Tabla 4.8.

*Tabla 4.8 Estimación de costo mensual del servicio Azure CDN  
Fuente: Elaboración propia*

Azure CDN	Peso Contenido (GB)	Usuarios	Total Peso Contenido (GB)	Total (\$) / mes
<b>GB descargados</b>	0,0076219	31	0,2362789	0,06
<b>COSTO TOTAL (\$)</b>				\$0,06
<b>COSTO TOTAL (S/)</b>				S/.0,23

#### 4.6.3 Resumen de Costos

Para elaborar el flujo de caja del proyecto se debe consolidar una tabla que resuma los costos de operación y de mano de obra, los cuales se detallan en las Tablas 4.9 y 4.10 respectivamente.

Tabla 4.9. Resumen de costos de tipo OPEX  
Fuente: Elaboración propia

	Precio (\$) / mes	Cantidad	Costo (\$) / mes
<b>Costos de OPEX</b>			
<b>Azure Virtual Machine</b>	\$7,59	2	\$15,18
<b>Azure Database for MySQL</b>	\$8,50	1	\$8,50
<b>Azure DNS</b>	\$2,16	1	\$2,16
<b>Azure Blob Storage</b>	\$0,05	1	\$0,05
<b>Azure CDN</b>	\$0,06	1	\$0,06
<b>Firestore Cloud Messaging</b>	-	-	-
<b>MONTO TOTAL (\$)</b>			<b>\$25,95</b>
<b>MONTO TOTAL (S/)</b>			<b>S/.97,57</b>

Tabla 4.10. Resumen de costos de tipo CAPEX  
Fuente: Elaboración propia

	Costo total
<b>Costos de CAPEX</b>	
<b>Mano de Obra</b>	\$1.290,16
<b>MONTO TOTAL (\$)</b>	<b>\$1.290,16</b>
<b>MONTO TOTAL (S/)</b>	<b>S/.4.851,00</b>

#### 4.6.4 Flujo de Caja Mensual por Escuela de Manejo

Para elaborar el flujo de caja anual, se tomarán en cuenta tanto los costos operativos como los de capital definido en la Sección 4.6.3. Además, se incluirá un egreso mensual S/.810, 00 (equivalente a 215,43 USD), destinado a la remuneración del encargado de dar soporte a la infraestructura en la nube. Esta persona tendrá la responsabilidad de monitorear, supervisar y solucionar problemas como la caída de un servidor, entre otros. Dicha remuneración se calculó estimando 3 horas de trabajo diarias durante un mes, al mismo precio por hora definido en la

Tabla 4.3, partiendo del hecho de que, al estar la infraestructura alojada en la nube, no es necesaria una supervisión a tiempo completo.

Entonces, siguiendo un modelo de venta tipo SaaS, se establece un precio de suscripción mensual de S/. 1025,00 (equivalente a 272,61 USD) por el software. Este precio es menor al salario promedio de una secretaria en Perú, que es de S/. 1184,00 según CompuTrabajo [60], que representaría el ahorro generado a las escuelas de manejo, debido a la reducción de personal, por la implementación del proyecto en su ecosistema. Estas variables permitirán calcular el flujo neto mensual del proyecto, y los detalles se presentan en la Figura 4.19.

Servicio	Mes 0	Mes 1 (Enero)	Mes 2 (Febrero)	Mes 3 (Marzo)	Mes 4 (Abril)	Mes 5 (Mayo)	Mes 6 (Junio)	Mes 7 (Julio)	Mes 8 (Agosto)	Mes 9 (Septiembre)	Mes 10 (Octubre)	Mes 11 (Noviembre)	Mes 12 (Diciembre)
<b>Costos de Inversión</b>													
Mano de Obra	-1.290,16	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
<b>Costos de Operación</b>													
Servicios en la Nube	0,00	-25,95	-25,95	-25,95	-25,95	-25,95	-25,95	-25,95	-25,95	-25,95	-25,95	-25,95	-25,95
Soporte de Infraestructura en la Nube	0,00	-215,43	-215,43	-215,43	-215,43	-215,43	-215,43	-215,43	-215,43	-215,43	-215,43	-215,43	-215,43
Ingresos	0,00	272,61	272,61	272,61	272,61	272,61	272,61	272,61	272,61	272,61	272,61	272,61	272,61
<b>Flujo Neto (\$)</b>	<b>-1.290,16</b>	<b>31,23</b>	<b>31,23</b>	<b>31,23</b>	<b>31,23</b>	<b>31,23</b>	<b>31,23</b>	<b>31,23</b>	<b>31,23</b>	<b>31,23</b>	<b>31,23</b>	<b>31,23</b>	<b>31,23</b>

Figura 4.19. Flujo de caja mensual por escuela de manejo  
Fuente: Elaboración propia

#### 4.6.5 Flujo de Caja Anual estimado

Para evaluar la rentabilidad del proyecto, se establece un horizonte de cinco años. El Costo Promedio Ponderado de Capital (WACC), que para empresas del sector de telecomunicaciones se considera como la Tasa de Costo de Oportunidad (TCO), considerado para la tesis es de 6.80%, de acuerdo a un estudio realizado por KPMG sobre el costo-capital de las industrias en el país [61]. Estimando que el proyecto va a tener una buena aceptación por parte de las escuelas de manejo, se espera la captación de una escuela de manejo suscriptora cada año, lo cual se refleja en el flujo de caja anual estimado en la Figura 4.20.

WACC	6,80%						
<b>Año</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>VAN</b>
Egresos	1290,16	-	-	-	-	-	3.142,15
Ingresos	-	374,81	749,63	1124,44	1499,25	1874,07	<b>TIR</b>
<b>Flujo</b>	<b>-1.290,16</b>	<b>374,81</b>	<b>749,63</b>	<b>1.124,44</b>	<b>1.499,25</b>	<b>1.874,07</b>	56,42%
Valor actual (V.A.)	-1.290,16	350,95	657,21	923,04	1152,37	1348,74	<b>PAYBACK</b>
V.A. acumulado	-1.290,16	-939,21	-282,00	641,04	1.793,41	3.142,15	3 año

*Figura 4.20. Flujo de Caja Anual estimado  
Fuente: Elaboración propia*

De la Figura 4.20, se observa que el Valor Actual Neto (VAN) del proyecto es de 3.142,15 USD y la Tasa Interna de Retorno (TIR) de 56.42%. Estos valores, que incluyen un VAN positivo y una TIR que supera al WACC, indican la rentabilidad del proyecto según las estimaciones. Además, se espera que la rentabilidad del proyecto comenzará a reflejarse a partir del tercer año de operación.

#### **4.7 Análisis Ambiental y de Sostenibilidad**

Debido al tema tratado en la tesis, un análisis relacionado con el medio ambiente y la sostenibilidad no justifica.

#### **4.8 Análisis relacionado a la Salud Pública y Seguridad**

Un análisis en relación a la salud pública y seguridad no amerita, considerando el enfoque de la tesis.

#### **4.9 Análisis relacionado al Bienestar y el Orden Social**

La implementación de esta solución tecnológica propuesta brindará una mejor experiencia de los usuarios, tanto para los clientes como para el personal de la escuela de manejo. Esto fomentará una cultura orientada hacia la vanguardia tecnológica, en relación con la modernización operativa de una escuela de manejo, tal como se detalló previamente en la Sección 1.7.1.



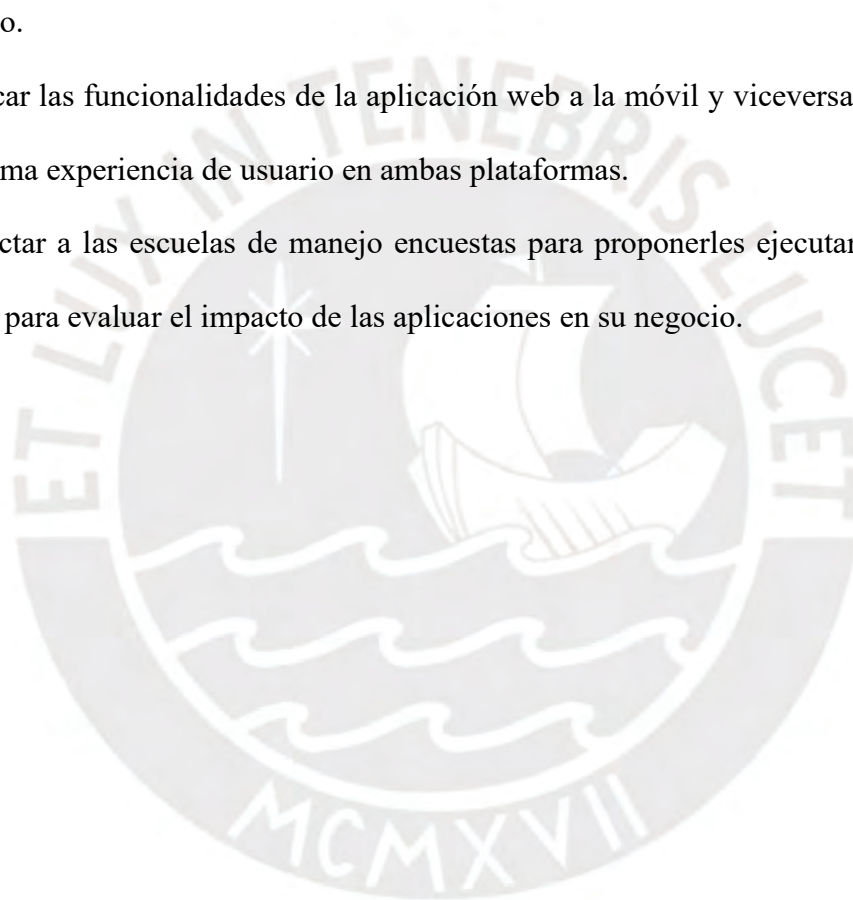
## CONCLUSIONES

1. Se diseñó e implementó satisfactoriamente una aplicación web que permita a una escuela de manejo optimizar sus procesos administrativos, como la gestión de alumnos, horarios de clase de manejo, información de vehículos y recopilación de datos relevantes para el balance de operaciones.
2. Se logró desarrollar una aplicación móvil Android que permite enviar notificaciones como recordatorios 15 minutos antes de cada clase de manejo programada a los alumnos. Además, facilita a los instructores la generación de informes de aprendizaje para sus alumnos.
3. El servicio de “Enlaces de Pago” de Stripe permite simplificar la implementación de pasarelas de pago en aplicaciones, delegando la gestión de las operaciones necesarias a la infraestructura de Stripe.
4. Se utilizó una base de datos relacional en la nube mediante el servicio de Microsoft Azure: Database for MySQL, lo que facilitó el registro y la organización estructurada de la información.
5. Se desplegó la aplicación web en la nube de Microsoft Azure, permitiendo una alta disponibilidad y proporcionando un proceso de despliegue simplificado y eficiente en la nube.

## TRABAJO A FUTURO

Se propone realizar los siguientes trabajos a futuro a partir de la tesis:

- Implementar la aplicación móvil para sistemas operativos iOS.
- Integrar a la aplicación web un módulo de Promoción y Marketing para potenciar la visibilidad de la escuela.
- Integrar otros métodos de pago como Yape y Plin para facilitar la reserva de clases de manejo.
- Replicar las funcionalidades de la aplicación web a la móvil y viceversa para asegurar la misma experiencia de usuario en ambas plataformas.
- Contactar a las escuelas de manejo encuestas para proponerles ejecutar un programa piloto para evaluar el impacto de las aplicaciones en su negocio.



## BIBLIOGRAFÍA

- [1] “¿Cómo ayuda la tecnología a las empresas en la actualidad?,” *BIT*, Sep. 30, 2019. <https://www.bit.lat/posts/como-ayuda-la-tecnologia-a-las-empresas-en-la-actualidad/>
- [2] “¿Qué es la transformación digital? - Explicación de la transformación digital - AWS,” *Amazon Web Services, Inc.* <https://aws.amazon.com/es/what-is/digital-transformation/>
- [3] R. Gestión, “¿Qué es exactamente la transformación digital?,” *Gestión*, Mar. 04, 2019. [Online]. Available: <https://gestion.pe/economia/management-empleo/transformacion-digital-mnda-nnlt-260271-noticia/?ref=gesr>
- [4] «Escuela de Conductores», *Ministerio de Transportes y Comunicaciones*. [https://portal.mtc.gob.pe/transportes/terrestre/licencias/consulta\\_escuela\\_conductores.html](https://portal.mtc.gob.pe/transportes/terrestre/licencias/consulta_escuela_conductores.html)
- [5] «Decreto Supremo n.º 007-2016-MTC», *Normas y documentos legales - Ministerio de Transportes y Comunicaciones - Plataforma del Estado Peruano*. <https://www.gob.pe/institucion/mtc/normas-legales/9961-007-2016-mtc>
- [6] “¿Qué es una aplicación web? - Explicación de las aplicaciones web - AWS,” *Amazon Web Services, Inc.* <https://aws.amazon.com/es/what-is/web-application/>
- [7] Chakray, “DevOps: Arquitectura monolítica vs Microservicios - Chakray,” *Chakray*, Jun. 06, 2023. <https://www.chakray.com/es/devops-arquitectura-monolitica-vs-microservicios/#:~:text=Una%20arquitectura%20monol%C3%ADtica%20es%20un,para%20sus%20servicios%20o%20funcionalidades.>
- [8] E. Gallardo, “What Is a Monolithic Application?,” *Built In*, Dec. 2022, [Online]. Available: <https://builtin.com/software-engineering-perspectives/monolithic-application>
- [9] Lavann, “Conversión de aplicaciones monolíticas en microservicios mediante el diseño basado en dominios - Azure Architecture Center,” *Microsoft Learn*. <https://learn.microsoft.com/es-es/azure/architecture/microservices/migrate-monolith>
- [10] “¿Qué es una arquitectura de aplicaciones? Tipos de arquitecturas de aplicaciones.” [https://www.redhat.com/es/topics/cloud-native-apps/what-is-an-application-architecture#:~:text=En%20la%20actualidad%2C%20las%20principales,servicio%20\(de%20bajo%20acoplamiento\).](https://www.redhat.com/es/topics/cloud-native-apps/what-is-an-application-architecture#:~:text=En%20la%20actualidad%2C%20las%20principales,servicio%20(de%20bajo%20acoplamiento).)
- [11] “¿Qué son los microservicios? | AWS,” *Amazon Web Services, Inc.* <https://aws.amazon.com/es/microservices/>
- [12] “¿Qué son los microservicios?,” *TIBCO Software*. <https://www.tibco.com/es/reference-center/what-are-microservices>

- [13] Ardalís, «Información general de ASP.NET Core MVC», *Microsoft Learn*, 3 de agosto de 2023. <https://learn.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-7.0>
- [14] Baeldung and Baeldung, “Difference between MVC and MVP patterns | Baeldung,” *Baeldung*, Nov. 09, 2022. <https://www.baeldung.com/mvc-vs-mvp-pattern>
- [15] J. Manrique, “Frameworks MVC y Ruby on Rails - Academia Hack - Medium,” *Medium*, Dec. 10, 2021. [Online]. Available: <https://medium.com/academia-hack/frameworks-mvc-y-ruby-on-rails-1afc3c501c49>
- [16] “¿Qué es JavaScript? - Explicación de JavaScript (JS) - AWS,” *Amazon Web Services, Inc.* <https://aws.amazon.com/es/what-is/javascript/>
- [17] “Angular.” <https://docs.angular.lat/guide/architecture>
- [18] B. I. Castro, “¿Cuáles son las principales novedades de Angular 9?,” *Formadores IT*, Feb. 15, 2023. <http://www.formadoresit.es/cuales-son-las-principales-novedades-de-angular-9/>
- [19] “Learn Node.js | Node.js.” <https://nodejs.dev/en/learn/>
- [20] D. C. H. Infante and D. C. H. Infante, “Qué es Node.js: Casos de uso comunes y cómo instalarlo,” *Tutoriales Hostinger*, Apr. 19, 2023. <https://www.hostinger.es/tutoriales/que-es-node-js>
- [21] F. Hernandez, «¿Que es Node.js? - Fernando Hernandez - Medium», *Medium*, 19 de marzo de 2022. [En línea]. Disponible en: <https://medium.com/@FernandoFH/que-es-node-js-9559cef62ff>
- [22] “¿Qué es Java? Guía de Java para principiantes | Microsoft Azure,” Microsoft Azure. <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-java-programming-language/>
- [23] “¿Qué es Java? - Explicación del lenguaje de programación Java - AWS,” *Amazon Web Services, Inc.* <https://aws.amazon.com/es/what-is/java/>
- [24] “Spring Framework,” *Spring Framework*. <https://spring.io/projects/spring-framework>
- [25] Formadores IT, “Cursos Spring - Formación Java para empresas - Formadores IT,” *Formadores IT*, Feb. 17, 2023. <http://www.formadoresit.es/formacion-en-empresas/formacion/cursos-java-madrid/cursos-spring/>
- [26] “IBM documentation.” <https://www.ibm.com/docs/en/spm/7.0.9?topic=requirements-defining-mobile-application>

- [27] L. N. Delía, N. Galdámez, P. J. Thomas, y P. M. Pesado, “Un análisis experimental de tipo de aplicaciones para dispositivos móviles”, 2013. [En línea]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/32397>
- [28] M. Kapias and M. Kapias, “React Native vs Flutter: Which One is Better for 2023?,” *Fireart Studio*, Nov. 13, 2023. <https://fireart.studio/blog/flutter-vs-react-native-what-app-developers-should-know-about-cross-platform-mobile-development/>
- [29] “¿Qué es Flutter? - Explicación de la aplicación Flutter - AWS,” *Amazon Web Services, Inc.* <https://aws.amazon.com/es/what-is/flutter/>
- [30] Admin, “Desarrollo en Flutter - Devtop,” *Devtop*, Oct. 18, 2022. <https://devtop.io/desarrollo-en-flutter/>
- [31] “React Native · Learn once, write anywhere.” <https://reactnative.dev/>
- [32] “React native | SDP Capacitación TI,” *SDP Capacitación TI*. <https://www.sdpcapacitacion.com/reactnative>
- [33] “¿Qué es una base de datos? - Explicación de las bases de datos en la nube - AWS,” *Amazon Web Services, Inc.* <https://aws.amazon.com/es/what-is/database/>
- [34] “¿Qué es una base de datos?” <https://www.oracle.com/pe/database/what-is-database/>
- [35] “¿Qué es SQL? - Explicación de lenguaje de consulta estructurado (SQL) - AWS,” *Amazon Web Services, Inc.* <https://aws.amazon.com/es/what-is/sql/>
- [36] “¿Qué es NoSQL? | AWS,” *Amazon Web Services, Inc.* <https://aws.amazon.com/es/nosql/>
- [37] “¿Qué son las bases de datos NoSQL? | IBM.” <https://www.ibm.com/es-es/topics/nosql-databases>
- [38] MongoDB, “What is NoSQL? NoSQL databases explained,” *MongoDB*. <https://www.mongodb.com/es/nosql-explained>
- [39] G. G. G. P, «Sistema de Gestión Administrativo para Escuela de Conductores CASYC», 2015. <http://repobib.ubiobio.cl/jspui/handle/123456789/1739>
- [40] «Sistema WEB para la Gestión Administrativa de la Escuela de Manejo Cortés de Managua en el II semestre 2017 - Repositorio Institucional UNAN-Managua». <http://repositorio.unan.edu.ni/12874/>
- [41] A. F. C. Fernando, «Aplicación móvil de gestión administrativa y académica en la Escuela de Formación y Capacitación de Conductores Profesionales Ambato utilizando la plataforma Android», 2017. <https://repositorio.uta.edu.ec/handle/123456789/26532>

- [42] G. O. David, «Simulación de conducción con realidad virtual para el estudio de la seguridad y eficiencia del conductor», *Universidad de Valladolid*, 2017.  
<https://uvadoc.uva.es/handle/10324/27572>
- [43] J. M. Francisco, «Estimulando la conducción eficiente mediante una app para Android», *Universidad Autónoma de Madrid*, 1 de julio de 2018. <https://repositorio.uam.es/handle/10486/688434>
- [44] «MTC habilita plataforma tecnológica para evaluaciones médicas para brevets clase A y B», *Noticias - Ministerio de Transportes y Comunicaciones - Plataforma del Estado Peruano*. <https://www.gob.pe/institucion/mtc/noticias/622888-mtc-habilita-plataforma-tecnologica-para-evaluaciones-medicas-para-brevets-clase-a-y-b>
- [45] Jeahyoun, “Register,” *Microsoft Learn*. <https://learn.microsoft.com/es-es/users/register?redirectUrl=https%3A%2F%2Flearn.microsoft.com%2Fes-es%2Fazure%2Fdns%2Fdns-overview>
- [46] Jeahyoun, “Register,” *Microsoft Learn*. <https://learn.microsoft.com/en-us/users/register?redirectUrl=https%3A%2F%2Flearn.microsoft.com%2Fen-us%2Fazure%2Fvirtual-machines%2Foverview>
- [47] Kinsta, “¿Qué Es NGINX y Cómo Funciona? NGINX explicado para principiantes,” *Kinsta®*, Feb. 21, 2022. <https://kinsta.com/es/base-de-conocimiento/que-es-nginx/>
- [48] NGINX Inc., “Welcome to NGINX Wiki’s documentation!,” *NGINX Wiki*, 2015.  
<https://www.nginx.com/resources/wiki/>
- [49] “Let’s Encrypt - Certificados SSL/TLS gratuitos.” <https://letsencrypt.org/es/>
- [50] “Stack overflow developer survey 2023”, *Stack Overflow*. [En línea]. Disponible en: <https://survey.stackoverflow.co/2023/>.
- [51] Jeahyoun, “Register,” *Microsoft Learn*. <https://learn.microsoft.com/es-es/users/register?redirectUrl=https%3A%2F%2Flearn.microsoft.com%2Fes-es%2Fazure%2Fstorage%2Fblobs%2Fstorage-blobs-introduction>
- [52] G. Ongo and G. P. Kusuma, “Hybrid Database System of MySQL and MongoDB in Web Application Development,” in *Proceedings of 2018 International Conference on Information Management and Technology, ICIMTech 2018*, 2018, no. September, pp. 256–260. doi: 10.1109/ICIMTech.2018.8528120.
- [53] Jeahyoun, “Register,” *Microsoft Learn*. <https://learn.microsoft.com/en-us/users/register?redirectUrl=https%3A%2F%2Flearn.microsoft.com%2Fen-us%2Fazure%2Fazure-sql%2Fdatabase%2Fsql-database-paas-overview%3Fview%3Dazuresql>

- [54] H. Ramírez, “Stripe ¿Qué es y cómo funciona este método de pago?,” *Grupo Atico34*, Apr. 16, 2023. <https://protecciondatos-lopd.com/empresas/stripe-metodo-de-pago/>
- [55] “Stripe lanza Payment Links para que se puedan realizar ventas en línea en minutos sin tener que programar.” <https://stripe.com/es-419-us/newsroom/news/stripe-launches-payment-links>
- [56] “Acerca de los mensajes de FCM | Firebase Cloud Messaging,” *Firebase*. <https://firebase.google.com/docs/cloud-messaging/concept-options?hl=es-419>
- [57] P. Sevestre and P. Sevestre, “Using Firebase cloud messaging in Spring Boot applications | Baeldung,” *Baeldung*, Jan. 12, 2023. <https://www.baeldung.com/spring-fcm>
- [58] Duongau, “¿Qué es una red de entrega de contenido (CDN)? -Azure,” *Microsoft Learn*, Mar. 25, 2023. <https://learn.microsoft.com/es-es/azure/cdn/cdn-overview>
- [59] “Calculadora de precios | Microsoft Azure,” *Microsoft Azure*. <https://azure.microsoft.com/es-mx/pricing/calculator/>
- [60] Computrabajo Perú, “¿Cuanto gana un Secretaria? | Sueldo Perú 2023,” *Computrabajo Perú*. <https://pe.computrabajo.com/salarios/secretaria>
- [61] S. SchÖNiger, H. Snellen, and A. TschÖPel, “Cost of Capital Study 2022,” *KPMG*, Oct. 17, 2022. [Online]. Available: <https://kpmg.com/de/en/home/insights/2022/10/cost-of-capital-study-2022.html>

## ANEXOS

### Anexo 1. Archivo de configuración Dockerfile para la creación de imágenes Docker en la Azure Virtual Machine 1

```
-- Archivo de configuración Dockerfile para el "Frontend"

# Crear la etapa de nginx para servir el contenido
FROM nginx:latest

# Copiar el contenido estático generado al compilar la aplicación en Angular al sistema de
archivos del contenedor
COPY /td-tfront/ /usr/share/nginx/html

# El contenedor expone su puerto 80, ejecuta nginx con directiva global y Daemon
desactivado
EXPOSE 80
ENTRYPOINT [ "nginx" , "-g" , "daemon off;" ]

-- Archivo de configuración Dockerfile para el "Backend"

# Crear la etapa de java
FROM openjdk:17-oracle

# Copiar el archivo .jar generado al compilar la aplicación en Spring Boot al sistema de
archivos del contenedor
ADD tdtback.jar tdtback.jar

# El contenedor expone su puerto 8081
EXPOSE 8081

#Correr la aplicación de java
ENTRYPOINT ["java", "-jar", "tdtback.jar"]
```



## Anexo 2. Archivo de configuración Dockerfile para la creación de imágenes Docker en la Azure Virtual Machine 2

```
-- Archivo de configuración Dockerfile para el "Backend"

# Crear la etapa de java
FROM openjdk:17-oracle

# Copiar el archivo .jar generado al compilar la aplicación en Spring Boot al sistema de
archivos del contenedor
ADD tdtback.jar tdtback.jar

# El contenedor expone su puerto 8081
EXPOSE 8081

#Correr la aplicación de java
ENTRYPOINT ["java", "-jar", "tdtback.jar"]

-- Archivo de configuración Dockerfile para el "Notifications Scheduler"

# Crear la etapa de node
FROM node:latest

# Establecer el directorio de trabajo en /app
WORKDIR /app

# Copiar los archivos package.json y package-lock.json al directorio de trabajo
COPY package*.json ./

# Instalar las dependencias usando npm
RUN npm install --force

# Copiar el resto de los archivos de la aplicación al directorio de trabajo
COPY . .

# Establecer la zona horaria
ENV TZ=America/Lima

# Ejecutar la aplicación node
CMD ["node", "notifications.js"]
```