PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE SANTO DOMINGO

Coordinación de Tecnologías de la Información y Diseño

SISTEMA IOT CON MACHINE LEARNING PARA EL CONTROL DE ESTACIONAMIENTO VEHICULAR EN EL COMERCIAL MIÑACA DEL CANTÓN SANTO DOMINGO

TRABAJO DE TITULACIÓN

Previo a la obtención del título de Ingeniero en Tecnologías de la Información

Línea de investigación: Tecnologías de la información y la comunicación

Autoría:

Carrascal Cedeño Byron David

Cruz Zambrano Patricio Ivan

Dirección:

Ocampo Pazos Willian Javier, Mg.

Santo Domingo – Ecuador
Febrero, 2024

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE SANTO DOMINGO

Coordinación de Tecnologías de la Información y Diseño

HOJA DE APROBACIÓN

SISTEMA IOT CON MACHINE LEARNING PARA EL CONTROL DE ESTACIONAMIENTO VEHICULAR EN EL COMERCIAL MIÑACA DEL CANTÓN SANTO DOMINGO

Línea de investigación: Tecnologías de la información y la comunicación

Autoría:

Carrascal Cedeño Byron David

Cruz Zambrano Patricio Ivan

Revisado por:

Ocampo Pazos Willian Javier, Mg. _____
DIRECTOR DEL TRABAJO DE TITULACIÓN

Cordova Galvez Rodolfo Sirilo, Mg. _____
CALIFICADOR

Ulloa Meneses Luis Javier, Mg. _____
CALIFICADOR

Carrasco Ramírez Franklin Andrés, Mg. _____
COORDINADOR DE LA CARRERA DE GRADO

Santo Domingo – Ecuador
Febrero, 2024

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

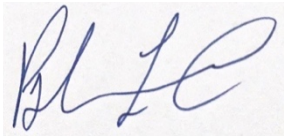
Nosotros, Cruz Zambrano Patricio Ivan, portador de la cédula de ciudadanía 1350176101, y Carrascal Cedeño Byron David, portador de la cédula de ciudadanía 1758605925, declaramos que los resultados obtenidos en la investigación que presentamos como informe final, previo a la obtención del Grado de Ingeniero en Tecnologías de la Información son absolutamente originales, auténticos y personales.

En tal virtud, declaramos que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de nuestra sola y exclusiva responsabilidad legal y académica.

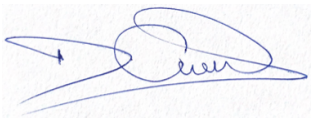
Igualmente, declaramos que todo resultado académico que se desprenda de esta investigación y que se difunda tendrá como filiación la Pontificia Universidad Católica del Ecuador Sede Santo Domingo, reconociendo en las autorías al director del Trabajo de Titulación y demás profesores que amerita.

Además, declaro que el presente trabajo, producto de las actividades académicas y de investigación, forma parte del capital intelectual de la Pontificia Universidad Católica del Ecuador, Sede Santo Domingo, de acuerdo con lo establecido en el artículo 16, literal j), de la Ley Orgánica de Educación Superior.

En tal razón, autorizo a la Pontificia Universidad Católica del Ecuador, Sede Santo Domingo, para que pueda hacer uso, con fines netamente académicos, del Trabajo de Titulación, ya sea de forma impresa, digital y/o electrónica o por cualquier medio conocido o por conocerse, siendo el presente documento la constancia del consentimiento autorizado; y, para que sea ingresado al Sistema Nacional de Información de la Educación Superior del Ecuador para su conocimiento público, en cumplimiento del artículo 103 de la Ley Orgánica de Educación Superior.



Cruz Zambrano Patricio Ivan
C.C. 1350176101



Carrascal Cedeno Byron David
C.C. 1758605925

INFORME DE TRABAJO DE TITULACIÓN ESCRITO DE GRADO

Cano de la Cruz, Yullio, PhD

Dirección de Investigación y Postgrados

Pontificia Universidad Católica del Ecuador Sede Santo Domingo

De mi consideración,

Por medio del presente informe en calidad de director del Trabajo de Titulación del Grado de Ingenieria en Tecnologías de la Información titulado: Sistema IoT con Machine Learning para el control de estacionamiento vehicular en el Comercial Miñaca del Cantón Santo Domingo, realizado por los estudiantes: Cruz Zambrano Patricio Ivan con cédula de ciudadanía 1350176101 y Carrascal Cedeño Byron David con cédula de ciudadanía 1758605925, previo a la obtención del título de Ingeniero en Tecnologías de la Información, informo que el presente Trabajo de Titulación escrito se encuentra finalizado conforme a la guía y al formato de la Sede vigente.

Además, certifico haber verificado la originalidad y autenticidad del trabajo de titulación por medio del programa anti plagio Turnitin, en respuesta a la normativa institucional vigente.

Santo Domingo, 20/02/2024.

Atentamente,

Mg. Willian Javier Ocampo Pazos

Profesor Titular Auxiliar II

RESUMEN

El control de estacionamiento vehicular permite gestionar la disponibilidad de plazas existentes mediante sensores, la adopción de sistemas inteligentes con *IoT* en los estacionamientos, mantienen un registro correspondiente a la entrada y salida de vehículos, generando así datos que tienen valor para la toma de decisiones. En el Comercial Miñaca del cantón Santo Domingo, se evidenció con encuestas que existe un inadecuado control del estacionamiento vehicular. Por lo cual, se empleó un enfoque cuantitativo con diseño pre-experimental, por contexto de la investigación se aplicó el muestreo por conveniencia, obteniendo así datos de 80 clientes que, durante un mes, utilizaron los 4 primeros estacionamientos de 23 en total. Con el propósito de resolver el problema, se desarrolló una aplicación *web* con *React, Django y MySQL*, para la visualización del estacionamiento, se crearon sensores con un *ESP8266* y el *HC-SR04*, el cual detecta la presencia de un vehículo, se incorporó el algoritmo regresión logística de *Machine Learning* para el pronóstico de clientes y se desplegó una infraestructura local con *Ubuntu Server* y en la nube con *Digital Ocean*, todo este desarrollo se manejó con el marco de trabajo *Scrum*.

Palabras claves: Sistema *IoT*; *Machine Learning*; Estacionamiento Vehicular.

ABSTRACT

Vehicle parking control allows managing the availability of existing spaces through sensors, the adoption of intelligent systems with IoT in parking lots, allows maintaining a record corresponding to the entry and exit of vehicles, thus generating data that has value for decision making. In the Comercial Miñaca of the Santo Domingo canton, surveys showed that there is inadequate control of vehicle parking. Therefore, a quantitative approach with a pre-experimental design was used, due to the context of the research, convenience sampling was applied, thus obtaining data from 80 clients who, for a month, used the first 4 parking lots out of 23 in total. In order to solve the problem, a web application was developed with React, Django and MySQL, to display the parking lot, sensors were created with an ESP8266 and the HC-SR04, which detects the presence of a vehicle, the Machine Learning logistic regression algorithm for customer forecasting and a local infrastructure was deployed with Ubuntu Server and in the cloud with Digital Ocean, all this development was managed with the Scrum framework.

Keywords: IoT System; Machine Learning; Vehicle Parking.

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN

El amplio crecimiento del internet que existe al día de hoy, junto con el constante avance de las nuevas tecnologías en diversos ámbitos, ha transformado significativamente la manera de relacionarse con la información. Este progreso ha llevado a una mayor accesibilidad y eficiencia en la manipulación de información, alcanzando un nivel en el cual las personas de todo el mundo, han adoptado las nuevas tecnologías como herramientas que agilizan sus actividades. Es aquí, donde aparece la incorporación del Internet de las Cosas (*IoT*) y *Machine Learning* como soluciones orientadas a enfrentar las dificultades que surgen de esta nueva evolución tecnológica.

El Internet de las Cosas (*IoT*) está conformado por una red de objetos, como sensores, software y otras tecnologías, que abarcan desde el ámbito doméstico hasta el industrial. Estos dispositivos establecen conexión a través de internet con el objetivo de intercambiar información. Esta se analiza mediante modelos de aprendizaje automático (*Machine Learning*), los cuales generan resultados en beneficio de los usuarios finales.

1.1. Antecedentes

La amplia incorporación de las últimas tecnologías por parte de ciudades u organizaciones, han brindado la capacidad de ofrecer servicios, que agilicen la gestión de las zonas de aparcamiento, esto se logra gracias a la utilización del Internet de las Cosas y al empleo de algoritmos, correspondientes al aprendizaje automático.

En este contexto, se plantea el problema abordado por Arjona, Linares, Casanovas-Garcia y Vázquez (2020) en España, donde desarrollaron un sistema basado en *IoT*, que permitió a los conductores de las ciudades urbanas conocer el estado de los estacionamientos, gracias a sensores inalámbricos que ayudaron a detectar la presencia del vehículo en tiempo real. Esto sumado a la aplicación de métodos predictivos con *Machine Learning*, que facilitaron a los conductores a encontrar estacionamiento lo más

cerca posible de su destino. Como resultado, se mejoró el flujo vehicular, el tiempo de espera por espacios disponibles y la congestión en zonas demandadas.

En igual medida, se plantea otro problema similar abordado por Valeo, Gregoracci, Seijas y Etcheverry (2020) en Argentina, donde elaboraron un sistema automatizado de estacionamiento (*SAE*), para cubrir la necesidad de implementación de espacios de aparcamiento automatizados en la localidad de Mar del Plata. Dicha solución fue realizada con un detector magnético, dos cámaras *IP* que registran la patente de los vehículos, además de contar con una pantalla que indica los espacios asignados. Como medio de respaldo ante la asignación del espacio, se imprime un *ticket* que al final debe ser pagado manualmente al operario del estacionamiento. La adopción de este sistema compensa la falta de estacionamientos disponibles, como consecuencia del incremento en la cantidad de automóviles que transitan por las calles.

Por otra parte, se observa el caso de Mejía (2021), donde incorporó una propuesta de solución *IoT* con tecnología *LoRaWAN* en Quito - Ecuador, para mejorar el proceso de asignación de los parqueaderos del conjunto habitacional Parque Real. Esto se consiguió con la incorporación de sensores inalámbricos, para la detección vehicular y una pantalla en la entrada del parqueadero, la cual es controlada por la recepción para asignar un estacionamiento a los usuarios y controlar el tiempo de asignación. La implementación de este sistema tuvo un impacto significativo en el procedimiento de asignación, entre los 150 parqueaderos disponibles para visitas, optimizando las labores esenciales del personal de recepción.

Finalmente, en el estudio realizado por Enciso, Sarango, Valladares y Condolo (2019) en Loja - Ecuador, se desarrolló una aplicación móvil de aparcamiento inteligente con tecnología *IoT*, para contrarrestar la escasez de estacionamientos por el crecimiento automotriz. La propuesta para solventar este problema, consistió en el establecimiento de una red de sensores destinada a identificar el estado de disponibilidad del estacionamiento

y una aplicación móvil. El funcionamiento del prototipo fue exitoso y se recomendó que la implementación del mismo se realice para beneficiar a los conductores y al gobierno.

1.2. Planteamiento y delimitación del problema

De los antecedentes revisados, se pudo destacar que la adopción de sistemas inteligentes en los estacionamientos, aportaron valor en la mejora del flujo vehicular, tiempos de espera, gestión ante el incremento de vehículos y optimización en las labores del personal de recepción o guardianía. Tres de los artículos analizados llegaron a satisfacer las necesidades aplicando un sistema de estacionamiento con sensores como parte de la estructura de *IoT* y otro de los proyectos implementados en el Ecuador, integró el sistema de estacionamiento inteligente en una aplicación móvil. Bajo esta perspectiva, se analizaron datos estadísticos que reflejan el incremento vehicular.

El aumento en el número de automóviles a nivel mundial, hasta la carencia de sistemas de control en los establecimientos locales del país, ha generado una deficiente gestión de estacionamientos vehiculares, que ocasionan inconvenientes en el día a día de las personas. Por lo tanto, se considera la evaluación realizada por la Organización Internacional del Trabajo (2021), donde se constató un incremento en la producción global de vehículos automotores durante los años 2010 y 2017, alcanzando un total de 95.66 millones de unidades fabricadas. No obstante, a finales de 2018, en el Pacífico y Asia se posicionó como líder con el 55% de la producción mundial de vehículos, aproximadamente 52 millones de unidades (p. 7). Según los datos recolectados por la *European Parking Association* (2013), se determinó que hay un total de 47.124.388 plazas reguladas en Europa, incluyendo los países no pertenecientes a la *EPA*, de las cuales 30.167.672 no pertenecieron a la vía pública y 16.956.716 formaron parte de la vía pública. Cabe destacar que no se incluyó en este estudio las plazas de las casas privadas (p. 4).

Según la información que ha sido proporcionada por el INEC - Instituto Nacional de Estadística y Censos (2022a), en el Anuario de Estadística de Transporte del 2021, en

Ecuador el número de vehículos aumentó en 977.695 vehículos entre 2012 y 2021, alcanzando los 2.535.853, lo que conlleva a un aumento anual del 5.6%. Durante el año 2021, se registró una matriculación de 2.5 millones de vehículos motorizados, esto significa un incremento del 7.4% en relación al año previo (p. 6). De los datos definidos por la INEC (2022b) en las Estadísticas de Edificaciones del 2021, de los 13.7 millones de m^2 para construcciones potenciales a ejecutarse, se esperó que 17.451 m^2 de área se considerarán como parte de la construcción de parqueaderos (p. 6).

En esta misma perspectiva, conforme a los datos proporcionados por el INEC (2022c), en los Tabulados del Anuario de Estadísticas de Transporte del 2021, en Santo Domingo hasta el 2021 se contaba con 82.508 vehículos motorizados matriculados, de los cuales 9 eran del estado, 6.916 eran de alquiler, 75.581 eran particulares y 2 vehículos con otros usos no registrados (p. 1).

Por lo tanto, se llevó a cabo una visita al comercial Miñaca, con la finalidad de llevar a cabo un análisis y definir su necesidad de controlar la zona de estacionamiento, con el propósito de ofrecer una experiencia más satisfactoria a los clientes al llegar con sus vehículos. Tomando en cuenta estas consideraciones, se pudo observar una gestión inadecuada de las incidencias, lo cual genera insatisfacción entre los clientes. Además, no se está haciendo un uso adecuado del estacionamiento durante las horas pico, lo que provoca una reducción en la disponibilidad de espacios. Añadido a esto, el registro de entrada y salida se lleva a cabo de manera deficiente, lo que resulta en inconsistencias en la información relacionada con los vehículos. Asimismo, no se cuenta con herramientas para consultar la disponibilidad de espacios, lo que prolonga el tiempo de espera de los clientes.

1.3. Preguntas de investigación

1.3.1. Pregunta General

¿Cómo fortalecer el proceso de control de estacionamiento vehicular en el comercial Miñaca del cantón Santo Domingo?

1.3.2. Preguntas Específicas

- ¿Cuáles son las necesidades que tiene el control de estacionamiento vehicular?
- ¿Qué aspectos son primordiales para la creación de una herramienta para la aplicación de machine learning?
- ¿Qué solución TICS se puede aplicar para el fortalecimiento del control de estacionamiento vehicular con el apoyo de machine learning?

1.4. Justificación

La presente investigación apunta a esclarecer las razones por la cual es necesario ejercer un control sobre los estacionamientos vehiculares. Según la Asamblea Nacional de la República del Ecuador (2021), específicamente en el capítulo 4, artículo 214z, dentro de los objetivos del SPPAT, se menciona el desarrollo de áreas de estacionamiento y puntos de descanso a lo largo de la carretera (p. 87). Esto implica la mejora de la infraestructura vial para ofrecer áreas de estacionamiento seguras. Por lo consiguiente, se plantea la importancia de controles de estacionamientos vehiculares que garanticen la seguridad y la fluidez del tráfico.

En segunda instancia, también se sustenta en la Ordenanza Municipal establecido por el GAD Municipal de Santo Domingo (2020), dentro del capítulo 1, en el artículo 4, donde se establece lo siguiente: Fomentar y establecer las regulaciones de circulación, así como el estacionamiento indebido de vehículos que circulan en las calles del cantón Santo Domingo, con la finalidad de salvaguardar la seguridad vial y garantizar la conservación y el

uso adecuado de las vías. (p. 5). Por otro lado, en relación a la ordenanza, los establecimientos públicos como privados, tienen el derecho legal de implementar sistemas que cumplan con los requerimientos de las personas, dado a que existe una ausencia de control en los estacionamientos.

Además, conforme a lo establecido por el Instituto Ecuatoriano de Normalización (2016), en la Normativa Técnica Ecuatoriana 2248, se designa la responsabilidad de contar con señalizaciones que permitan identificar y ubicar cada espacio de estacionamiento. Esto incluye la implementación de señales específicas, para personas que cuentan con una discapacidad o movilidad reducida (pp. 14-15). De igual forma, es importante aplicar estas regulaciones de señalización a los sistemas tecnológicos, con el fin de hacer más accesible su uso y comprensión para los usuarios.

Asimismo, para este trabajo de titulación se toma en consideración 3 de 17 Objetivos de Desarrollo Sostenibles definidos en la Agenda 2030 por las Naciones Unidas (2019), donde se establece en el objetivo 9, la construcción de infraestructuras, incentivando a la innovación para liberar fuerzas económicas y competitivas para el desarrollo de la economía. Estos elementos realizan un papel importante al fomentar nuevas tecnologías facilitando el comercio internacional y la utilización de los recursos. (p. 43).

Además, Naciones Unidas (2019), en el objetivo 11, tiene como propósito el crear ciudades que sean inclusivas, seguras, capaces y sostenibles en su funcionamiento debido al creciente proceso de urbanización a nivel mundial. La meta para 2030 es impulsar la urbanización inclusiva y sostenible además de facilitar la utilización de sistemas seguros de transporte (p. 51). Por otro lado, Naciones Unidas (2019), en el objetivo 13, busca tomar acciones inmediatas para hacer frente al cambio climático sin importar su ubicación, ya que esto conlleva consecuencias perjudiciales para la economía y el bienestar de vida de las personas. Una solución para los objetivos de este propósito es, fortalecer la capacidad para ajustarse a los riesgos asociados al clima (p. 59).

Por otra parte, este trabajo de titulación considera los objetivos establecidos según la Secretaría Nacional de Planificación (2021), donde se menciona al Objetivo 9 del eje de seguridad integral, el cual busca salvaguardar la seguridad de los ciudadanos, manteniendo el orden en la sociedad y gestionando situaciones de riesgo de manera eficaz (p. 76). Bajo este eje, se considera la política 9.2, que busca fortalecer la seguridad de los sistemas de transporte terrestre, con el objetivo de promover ambientes seguros.

Además, la Secretaría Nacional de Planificación (2021), menciona el Objetivo 12 del eje de transición ecológica, estimulando los enfoques de desarrollo sostenibles a través de la aplicación de medidas para adaptarse y reducir el impacto del cambio climático (p. 87). Ante la definición de los objetivos, es importante conocer el área de aplicación para el sistema de estacionamiento inteligente y las necesidades oportunas que se deben cubrir, considerando la innovación en infraestructuras, seguridad en ciudades y el cuidado del medio ambiente.

1.5. Objetivos de investigación

1.5.1. Objetivo general

Implementar un sistema *IoT* con *Machine Learning* para el control de estacionamiento vehicular en el comercial Miñaca del cantón Santo Domingo.

1.5.2. Objetivos específicos

- Identificar las necesidades que tiene el control de estacionamiento vehicular para la obtención de los requerimientos.
- Determinar herramientas y tecnologías de desarrollo adecuados para la implementación del control de estacionamiento vehicular.
- Desarrollar el sistema *IoT* con *Machine Learning* para el control de estacionamiento vehicular en el comercial Miñaca del cantón Santo Domingo.

2. REVISIÓN DE LA LITERATURA

2.1. Fundamentos teóricos

Para los fundamentos teóricos, ser realizaron diagramas que definen la estructura del tema, según Hernández y Mendoza (2018), se inicia con un índice preliminar global o general, se va ajustando hasta que alcanza un nivel muy detallado y finalmente se integran las referencias correspondientes dentro del esquema (p. 89). La figura 1 define la variable Internet de las Cosas, la figura 2 se dedica a la explicación de Machine Learning y la figura 3 se enfoca en el Control de Estacionamiento Vehicular.

Figura 1. Panorama bibliográfico de la variable independiente Internet de las Cosas (IoT)

Fases	Apartado	Sub apartado
Internet de las Cosas (IoT)	*Procesamiento*	*Cloud Computing* *Fog Computing* *Edge Computing*
	Protocolos	*MQTT* *CoAP* *AMQP* *HTTP/REST*
	Dispositivos Físicos	*Sensores* *Actuadores* *Microcontroladores*
	Tecnologías de Acceso	*WPAN* *WLAN* *WAN*

Figura 2. Panorama bibliográfico de la variable independiente Machine Learning

Fases	Apartado	Sub apartado
Machine Learning	*Tipos de Aprendizaje*	*Aprendizaje Supervisado* *Aprendizaje no Supervisado* *Aprendizaje Semi-supervisado* *Aprendizaje por Refuerzo*
	Algoritmos	*Regresión Lineal* *Regresión Logística* *Arboles de Decisión* *KNN* *Support Vector Machine* *Gradient Boosting* *Redes Neuronales*
	Aprendizaje Profundo (Deep Learning)	

Figura 3. Panorama bibliográfico de la variable dependiente Control de Estacionamiento Vehicular

Fases	Apartado	Sub apartado
Control de Estacionamiento Vehicular	*Sistema Inteligente Centralizado*	*Sistema de información y guiado de aparcamientos (PGIS)* *Sistema de información sobre la ocupación de aparcamientos (COINS)* *Sistema de guiado basado en agentes (ABGS)*
	Estrategia de Gestión	*Cantidad de Estacionamientos Utilizados* *Eficiencia en el Uso del Espacio*
	Estrategias de Oferta	*Diferenciación entre aparcamiento y construcción* *Aprovechamiento de los recursos de aparcamiento*
	Automatización	*Comportamientos de aparcamiento de los Vehículos Autónomos* *Detección de plazas de aparcamiento libres* *Método de tarificación dinámica*

2.1.1. Internet de las Cosas (*IoT*)

Según lo que define Gacovski (2019), se reconoce al Internet de las Cosas como un enfoque que asigna sensores, actuadores y procesadores a los objetos, que ayuda a comunicarse entre sí, para ejecutar funciones y cumplir un propósito significativo. Las tecnologías de *IoT*, han logrado con el tiempo reducir significativamente la labor humana y aumentar la calidad de vida en muchas actividades. Los dispositivos *IoT* están conectados a una red que transmite datos y agrega la capacidad de realizar tareas complejas, con varias tecnologías que trabajan juntas, que a su vez requieren un alto grado de inteligencia. El Internet de las Cosas se está implementado en diferentes áreas como la educación, entretenimiento, vida social, conservación de la energía, domótica y sistemas de transporte. (pp. 91–93).

2.1.1.1. Procesamiento

2.1.1.1.1. Cloud Computing

Según D'Agostino (2019), la Computación en la Nube (*Cloud Computing*) proporciona la capacidad de recuperar recursos informáticos a través de Internet, como software, almacenamiento y capacidad de procesamiento. En lugar de tener la infraestructura con los recursos de manera local, estos se ofrecen y se acceden a través de la nube mediante servidores remotos. Este modelo permite, ahorro de costos y la asignación de recursos de forma flexible, a su vez se suministra el servicio mediante un

modelo de pago por uso y el uso dependerá de los servicios que se prestan (pp. 1–3). Se considera una solución más completa y con menos mantenimiento sobre los sistemas, que una infraestructura local. La aplicación del *Cloud Computing* en *IoT*, permite reducir los costes, la movilidad y la capacidad de acceso a nivel mundial.

2.1.1.1.2. Fog Computing

Según lo indicado por Hanes et al. (2017), el *Fog Computing* es una arquitectura informática distribuida, que forma parte de los servicios de borde de *IoT* y se encarga de distribuir la gestión de los datos por todo el sistema. Este procesamiento permite analizar la información con mínima latencia y mantiene los datos sensibles dentro de la red local. Los servicios del *Fog Computing* se realizan normalmente muy cerca de los dispositivos de *IoT*. La ventaja que tienen, es que los dispositivos que son nodos de niebla, tienen el conocimiento contextual de los sensores que se están gestionando, debido a su cercanía geográfica a dichos sensores (pp. 65–67).

2.1.1.1.3. Edge Computing

De acuerdo a lo mencionado por Hanes et al. (2017), el *Egde Computing* es un enfoque descentralizado, en el cual el concepto del procesamiento *IoT* ha llegado a los dispositivos de borde como los sensores o dispositivos controladores *IoT*. Los recursos limitados que poseen los dispositivos de borde han aumentado, obteniendo suficientes recursos como para realizar al menos análisis y filtrado de bajo nivel. Análisis que permiten tomar decisiones básicas con baja latencia. La capacidad que tiene *Edge Computing* es que analiza lo que sucede sobre el dispositivo local, mientras que el *Fog Computing* considera el conjunto de dispositivos interconectados (p. 68).

2.1.1.2. Protocolos

2.1.1.2.1. MQTT

Conforme a lo señalado por Nikolov (2020), el protocolo *MQTT* se considera un protocolo de publicación/suscripción para comunicación *Machine to Machine*, este protocolo fue desarrollado inicialmente por *IBM*, pero después se convirtió en un estándar abierto para uso público. *MQTT* funciona con el modelo *cliente/servidor*, donde los sensores o controladores trabajan como clientes que se conectan a un servidor, este se conoce como

el *broker* y dentro del mismo se definen los temas. El protocolo se basa en el uso de mensajes *TCP*, que son enviados al tema compartido por los sensores. Estos sensores tienen la característica de que pueden publicar mensajes en el tema, mientras que otros controladores que estén suscritos al mismo tema pueden recibir y escuchar los mensajes enviados por los demás publicadores. En la seguridad, el protocolo utiliza un método de autenticación con usuario y contraseña, además de garantizar la privacidad de la conexión, con certificados *SSL/TLS* para el cifrado (p. 1).

2.1.1.2.2. CoAP

Bansal y Priya (2021) mencionan, que el protocolo *CoAP* es un protocolo similar a *HTTP*, diseñado por el grupo de trabajo *CoRE*, que se utiliza para la comunicación entre dispositivos, este protocolo es ligero y trabaja con el modelo *resource/observe*, similar a solicitud/suscripción. *CoAP* fue diseñado principalmente para la interacción con *HTTP* y sistemas de Internet de las Cosas, basados en *RESTful*. Para conocer donde se publican los datos, se emplean Identificadores Universales de Recursos (*URI*). Los *URI* son controlados por el servidor, los editores publican datos con este identificador y los suscriptores son notificados de que se ha publicado nuevos datos, los datos son identificados con una etiqueta de entidad (*E-Tag*) para diferenciarlos con datas anteriores (pp. 555–556).

2.1.1.2.3. AMQP

De acuerdo con lo mencionado por Naik y Bapat (2020), *AMQP* es un protocolo ligero basado en *Machine to Machine*, desarrollado en *JPMorgan Chase* en Londres, se considera un protocolo estándar abierto y se utiliza para interoperabilidad funcional entre el cliente y el *middleware*. El protocolo se basa en componentes que enrutan los mensajes a través de un servidor *broker*, para publicar y consumir se requiere de un nombre de intercambio que se difunde para todos. El protocolo se diseñó para ofrecer algunas características, como la estandarización, la fiabilidad, la interoperabilidad y la seguridad entre dispositivos. *AMQP* se considera binario, con algunas características como la negociación, multicanal, portabilidad, eficiencia e intercambio de mensajes asincrónicos.

Siendo el fuerte del protocolo, la entrega de mensajes rápida y garantizada, esta capacidad ayuda a la distribución de mensajes en un entorno multi clientes (pp. 137–138).

2.1.1.2.4. HTTP/REST

Conforme a lo que menciona Iglesias-Urkia et al.(2018), el protocolo *HTTP* se considera sin estado, lo que significa que cada solicitud y respuesta entre el cliente y servidor es independiente de otras peticiones. El protocolo utiliza métodos como *GET*, *POST y PUT*, que contiene una *URL* y la información se puede transmitir en diferentes formatos, como puede ser JSON, XML o HTML (p. 1018). Además, según Perna et al. (2022), el protocolo *HTTP* ha evolucionado desde su versión 1.1 hasta la versión 3 que sustituye *TCP* por *QUIC* como protocolo de transporte, mejorando el rendimiento y la seguridad (p. 116).

2.1.1.3. Dispositivos Físicos

2.1.1.3.1. Sensores

En base a lo que menciona Nagaraj (2021), los sensores son dispositivos de características pequeñas que se despliegan en diferentes entornos para monitorear cualquier objeto de interés, cuando están desplegados se encuentran en un estado inactivo, esperando captar un evento o cambios en el evento. Los sensores dependiendo de su tipo, logran recopilar lecturas necesarias que son procesadas y difundidas a los controladores, a esto se le conoce como redes de sensores inalámbricos. Los sensores se pueden utilizar para gestionar las condiciones físicas del entorno como el tiempo real, la temperatura, humedad o sonido. Las aplicaciones de los sensores aportan vigilancia en los fallos de infraestructura, conservación de recursos naturales y aumento de la producción (p. 1).

2.1.1.3.2. Actuadores

Según Singh et al. (2019), los actuadores son una parte de la electrónica que se considera como las máquinas responsables de una acción, como un movimiento o rotación, estos actuadores necesitan una fuente de energía para funcionar y señales para controlar su funcionamiento. Las señales de control pueden ser neumáticas, humanas o eléctricas, cuando el actuador recibe alguna señal de control, la señal será convertida en la acción

para la cual fue hecho el actuador como un movimiento mecánico, el tipo de actuador dependerá de su uso en la aplicación (p. 81).

2.1.1.3.3. Microcontroladores

Los microcontroladores son considerados por Westcott y Westcott (2020), como ordenadores integrados autónomos que permiten ejecutar una serie de instrucciones definidas en el código, las acciones se ejecutan con entradas y este genera salidas que están personalizadas para el funcionamiento esperado (p. 215).

Westcott y Westcott (2020) mencionan, que los microcontroladores tienen los componentes típicos de un ordenador, pero en menor medida, como es una *CPU*, puertos serie de *E/S*, memoria volátil como la *ROM* de solo lectura y el *flash* para el almacenamiento de las instrucciones, además incluye convertidor de señales analógico-digital. Los microcontroladores se programan con lenguajes de alto nivel como *C, C++* o *Python* que es convertido a lenguaje de bajo nivel ensamblador que a su vez es convertido en lenguaje máquina. Una aplicación son los sistemas embebidos, que están diseñados para cumplir únicamente una funcionalidad concreta (pp. 217–219).

2.1.1.4. Tecnologías de Acceso

2.1.1.4.1. WPAN

Considerando la aportación de Collin y Perry (2019), las redes de área personal como la comunicación de corto alcance se han convertido en una de las soluciones de enlace de datos en las implementaciones de *IoT*, considerando que este tipo de conexiones están predominando las conexiones industriales, comerciales y de consumo. *WPAN* se puede basar en una pila tradicional *IP* o una comunicación no *IP*. La tecnología *WPAN* más utilizada en *IoT* es *Bluetooth*, por sus significativas funcionalidades, además que dependiendo de la versión pueden llegar a más de 200 metros de cobertura, existen otras soluciones como *Zigbee*, *Z-Wave* e *IEEE 802.15.4*. La tecnología de área personal ha ampliado su significado gracias a la importancia en las implementaciones de Internet de las Cosas (pp. 107–108).

2.1.1.4.2. WLAN

Según Perry (2020), menciona que la red local inalámbrica (*WLAN*) permite comunicar dispositivos como móviles, portátiles o tabletas de manera inalámbrica utilizando los mismos protocolos de las capas del modelo *OSI*, simplemente reemplazando las capas *MAC* y Física por las tecnologías inalámbricas y el protocolo *802.11*. La *WLAN* tiene sus inicios en 1999 con la *Wi-Fi Alliance* y el protocolo original admitía una velocidad de transmisión de 2 *Mbps*, ahora existen protocolos de alta velocidad como el *802.11ac*. En el mundo del Internet de las Cosas, el uso del protocolo está dirigido a sensores inteligentes y concentradores de pasarelas, el protocolo *802.11ah* se utiliza en dispositivos como sensores que tienen limitaciones de potencia (pp. 195–196).

2.1.1.4.3. WAN

De acuerdo de la contribución de Perry (2020), menciona que la comunicación a largo alcance se logra con las redes área amplia (*WAN*), estas comunicaciones permiten cubrir la transmisión de datos a kilómetros de distancia, de dispositivos móviles, sensores, actuadores, cámaras, dispositivos inteligentes integrados, vehículos y robots, en lugares remotos de difícil acceso. La aplicación de redes *WAN,* se logra gracias a las tecnologías celulares como *4G LTE* y *5G,* que suelen estar relacionados con una suscripción a una operadora telefónica, para la comunicación con las torres de celular y sus infraestructuras. Existen soluciones de radio de largo alcance como *LoRaWAN* y *Sigfox*, que permiten tener una infraestructura y equipos propios (p. 243).

2.1.2. Machine Learning

El aprendizaje automático (*Machine Learning*), es un subconjunto de la *IA* que consiste en entrenar algoritmos, para que estos hagan predicciones o tomen decisiones basadas en los datos. De acuerdo con lo que menciona Rahman (2020), en *Machine Learning* los datos son cruciales, porque los algoritmos aprenden de ejemplos y patrones en los datos. Se considera que, en lo correspondiente a la calidad y la cantidad de datos, existe una influencia significativa en el rendimiento de un modelo de *ML*. Para la ejecución, existen varias técnicas utilizadas en el *ML*, como lo son los árboles de decisión, la máquina

de vectores soporte, los K vecinos más cercanos y el *clustering*, para aplicar estos algoritmos, hay que considerar algunos retos asociados al *ML*, como el sesgo de los datos y los modelos, la interpretabilidad de los resultados y las consideraciones éticas (pp. 38–41).

2.1.2.1. Tipos de Aprendizaje

2.1.2.1.1. Aprendizaje Supervisado

Según Golrokh y Hojjat (2022), mencionan que el aprendizaje supervisado se caracteriza por el entrenamiento del algoritmo, utilizando un conjunto de datos en el que se asignan etiquetas. Se proporcionan datos de entrada como de salida, además el algoritmo aprende a asignar los datos de entrada como los resultados adecuados mediante el análisis de los datos de entrenamiento. Uno de los retos del aprendizaje supervisado es, la obtención de extensos volúmenes de datos con etiquetas de alta calidad puede ser un proceso desafiante y costoso. Sin embargo, existen algunos conjuntos de datos con acceso públicos, que están a disposición para realizar la etapa del entrenamiento. Este aprendizaje se encuentra habitualmente en uso de aplicaciones médicas para realizar tareas como el diagnóstico de enfermedades o la predicción de los resultados del paciente (p. 2).

2.1.2.1.2. Aprendizaje No Supervisado

Siguiendo la línea de pensamiento de Golrokh y Hojjat (2022), expresan que el aprendizaje no supervisado permite detectar patrones de datos no etiquetados, entre las técnicas más importantes de aprendizaje no supervisado, se considera el *clustering*, que consiste en agrupar objetos similares en un mismo grupo o *cluster*. Las tareas del aprendizaje no supervisado consisten en extraer automáticamente características de los datos y asignar muestras de datos similares a los mismos grupos, considerando una medida de similitud. El funcionamiento se basa en extraer mediante una técnica de agrupación y después pasarlo a un clasificador para clasificar los datos, de esta forma el aprendizaje no supervisado provee de la capacidad para mejorar la comprensión y diagnósticos sobre los datos y el tema (p. 5).

2.1.2.1.3. Aprendizaje Semi-Supervisado

Según Sarker (2021), el aprendizaje semi-supervisado funciona con datos etiquetados y sin etiquetar, este tipo de aprendizaje se encuentra entre el no supervisado y

el supervisado. Los datos que están etiquetados pueden ser escasos o caros de obtener en varios contextos del mundo real, y los datos no etiquetados son abundantes, por lo que el aprendizaje semi-supervisado se vuelve altamente beneficioso al emplear datos nuevos o desconocidos. El propósito clave de un modelo de aprendizaje semi-supervisado radica en perfeccionar la exactitud de las predicciones, en comparación con el uso exclusivo de datos etiquetados. Existen muchas áreas de aplicación, algunas son la detección de fraudes, el etiquetado de datos, la clasificación de textos, la visión por ordenador y el reconocimiento de voz (p. 160).

2.1.2.1.4. Aprendizaje por Refuerzo

Siguiendo la línea de pensamiento de Sarker (2021), el aprendizaje por refuerzo se clasifica como un algoritmo dentro del campo de *Machine Learing*, que habilita a los agentes de software y las máquinas realicen evaluaciones automatizadas del comportamiento óptimo en un contexto o entorno específico para mejorar su eficacia. Esta perspectiva de aprendizaje se basa en el sistema de recompensas o sanciones y el objetivo primordial es emplear los conocimientos adquiridos para adoptar acciones que maximicen las recompensas y reduzcan el riesgo. El aprendizaje por refuerzo es una poderosa herramienta para capacitar modelos de inteligencia artificial, que logran impulsar la automatización y optimizar la eficiencia de sistemas complejos (p. 160).

2.1.2.2. Algoritmos

2.1.2.2.1. Regresión Lineal

Conforme a lo mencionado por Susmita (2019), la regresión lineal es un algoritmo de *Machine Learning*, utilizado para modelar las variables y hacer predicciones. El algoritmo se enfoca en examinar la relación lineal entre las variables que se definen como independientes y dependientes. Las predicciones son comúnmente utilizadas para predecir los valores numéricos, como el precio de un objeto, casa o sobre el rendimiento de una empresa ya que su enfoque, es localizar la línea que logre ajustarse de mejor forma a los datos, lo que requiere reducir los errores entre los valores reales y las predicciones. Es importante tomar en consideración la regresión lineal que posee limitaciones como, que no

es apta para capturar relaciones no lineales entre las variables o para manipular datos categóricos (p. 36).

2.1.2.2.2. Regresión Logística

Según lo expresado por Balaji et al. (2021), la regresión logística se define como un algoritmo de *Machine Learning* de clasificación. Su funcionalidad principal es encontrar el modelo que mejor se adapte a la descripción entre las relaciones de las características dicotómicas y un conjunto de atributos autónomos como variables independientes. La regresión logística puede ser binaria o binomial, donde solo existen dos resultados posibles que representen la variable dependiente cuando el resultado sea notable, puede ser 1 o 0. En la regresión logística binomial se utilizan las variables independientes para poder predecir la probabilidad como resultado. Por otro lado, existe la regresión logística multinomial que puede manejar más de dos variables dependientes como resultado. Además, existe la regresión logística ordenada que maneja variables dependientes ordinales (pp. 5–6).

2.1.2.2.3. Árboles de Decisión

Uddin et al. (2019) mencionan, que el árbol de decisión es el algoritmo de *Machine Learning*, más antiguo y destacado, su infraestructura es similar a un árbol que clasifica los datos, en donde los nodos se reemplazan por variables de entrada o atributos. Los árboles poseen en la parte superior el nodo raíz y los nodos internos poseen múltiples niveles, según el resultado de la prueba el algoritmo se ramificó, es decir escogerá la rama que lo dirija hacia un nodo y consecuente el camino seguirá hasta llegar a un nodo hoja, ya que de esta manera ayudará en problemas complejos que tengas múltiples variables y relaciones no lineales, son fáciles de interpretar lo que le otorga una ventaja en campo como el de la medicina y la ingeniería (p. 3).

2.1.2.2.4. KNN

Susmita (2019), señala que *KNN* K Vecino más cercano por sus siglas en inglés (*K Nearest Neighbor),* es un algoritmo del área de clasificación que emplea un sistema de base de datos agrupando puntos de datos similares que acostumbran a estar cerca de otros en el

espacio. El proceso de implementación de *KNN*, conlleva a encontrar los puntos K más cercanos a un punto de datos desconocido y procederá a asignar una etiqueta en función a las que se encuentran cercanas. El algoritmo *KNN*, se presenta como algo fácil de implementar, pero posee algunas características limitadas con respecto al ruido de datos lo que afecta de forma negativa a su precisión y el rendimiento del modelo puede presentar fallas por la elección de un valor K (p. 38).

2.1.2.2.5. Support Vector Machine

Según lo expresado por Uddin et al. (2019), *Support Vector Machine* funciona mediante un modelo que separa los datos y los clasifica en dos o más categorías, este modelo crea un hiperplano para ubicar los datos en diferentes clases que son útiles para solucionar problemas complejos que involucren varias variables y relaciones no lineales, asimismo pueden manipular grandes cantidades de datos y no son susceptibles al ruido y a los valores atípicos. Estas funciones envían los datos a un espacio donde se pueden separar fácilmente minimizando los errores de clasificación. Los puntos de datas se representan como un punto en el espacio de n dimensiones, ya que el número de características es representado por N y el valor de cada una de las características se relaciona con el valor de una coordenada específica en ese espacio (p. 3).

2.1.2.2.6. Gradient Boosting

De acuerdo a Ho et al. (2021), el *Gradient Boosting* es un algoritmo de *Machine Learning* que ha ganado aceptación en los últimos años gracias a su eficacia en la clasificación de conjuntos de datos complejos. Este modelo fusiona múltiples modelos de aprendizaje débiles, para formar un modelo predictivo completo que sea capaz de mejorar iterativamente el modelo ajustado a los pesos de las instancias que se encuentran mal clasificadas en cada iteración. Los *Gradient Boosting* poseen una capacidad para manejar amplias colecciones de datos con elevadas dimensiones y un mayor grado de dificultad, además de eso, este método no necesita que los datos sigan una distribución normal, lo que lo hace más adaptable que otros métodos estadísticos (p. 56).

2.1.2.2.7. Redes Neuronales

Según lo mencionado por Campesato (2020), una red neuronal tiene muchos nodos interconectados, o neuronas, que procesan y transmiten la información. Existen dos tipos de redes neuronales, las redes de *Hopfield* y las redes basadas en la retro propagación. Las redes de *Hopfield* utilizan una función de energía que permite aproximar a soluciones a problemas *NP-completos*, mientras que, en las redes que hacen uso de la retro programación, se centra en las tareas como predicción de las medias del *Dow Jones* y la lectura de material impreso en sistemas de reconocimiento óptico de caracteres (pp. 13-14).

2.1.2.3. Aprendizaje Profundo (*Deep Learning*)

En base a lo mencionado por Manel et al. (2021), el aprendizaje profundo es un subcampo del *Machine Learning*, en el que implica que las redes neuronales artificiales se entrenan para adquirir un extenso conjunto de los datos. Estas redes están formadas por numerosas capas de nodos interconectados, lo que le da la capacidad de aprender representaciones más complejas sobre los datos de entrada, además del análisis de espacios de características intermedias en el diseño de redes profundas. El aprendizaje profundo tiene arquitecturas que permiten analizar imágenes, videos, procesamiento del lenguaje, audio, además de sistemas de recomendación (p. 167).

2.1.3. Control de Estacionamiento Vehicular

Según Elsonbaty y Shams (2020), el incremento del número de automóviles y la demanda de espacios de estacionamiento, han generado problemas con la congestión del tráfico y la seguridad vial. Existen distintas soluciones para abordar este tema, por esta razón, el implementar sistemas de gestión inteligente, consigue traer numeroso beneficios, entre ellos, la reducción en el tiempo de búsqueda y la disminución en el consumo de combustible. Estas soluciones traen consigo un nivel de satisfacción para el cliente, ya que, proporciona facilidad e información actualizada sobre la disponibilidad de estacionamientos en tiempo real (pp. 55–56).

2.1.3.1. Sistemas Inteligentes Centralizado

2.1.3.1.1. Sistema de información y guiado de aparcamientos (PGIS)

Al-Turjman y Malekloo (2019) menciona, que el sistema de información y guiado de aparcamientos de transporte público por sus siglas (*PGIS*), recopila información sobre los aparcamientos de manera dinámica, aplicando detectores de bucle y distintos sensores ultrasónicos, infrarrojos y de microondas. Mediante este sistema avanzado de guía, informa a los conductores en tiempo real sobre los estacionamientos disponibles. Con el fin de alcanzar el propósito de mitigar los desafíos de estacionamiento en zonas urbanas, y contribuir a la construcción de ciudades inteligentes sostenibles (p. 4).

2.1.3.1.2. Sistema de información sobre la ocupación de aparcamientos (COINS)

Siguiendo la línea de pensamiento de Al-Turjman y Malekloo (2019), el *COINS,* conocido como sistema de información referente a la ocupación de aparcamiento, se define, como una tecnología que brinda información precisa, sobre los estados de los espacios de estacionamiento, sin depender de sensores individuales. *COINS* se simuló en distintos entornos con diferentes parámetros y diversas condiciones climáticas, lo que añade un nivel más de complejidad al sistema. Sin embargo, su aplicación en un aparcamiento multinivel puede no ser tan efectiva como otros sistemas debido a problemas de escalabilidad y cobertura (p. 5).

2.1.3.1.3. Sistema de guiado basado en agentes (ABGS)

Continuando con Al-Turjman y Malekloo (2019), menciona que el sistema de Guiado Basado en Agentes o más bien conocido como *ABGS*, se constituye como una tecnología que utiliza inteligencia artificial, para dirigir a los conductores a plazas de estacionamiento disponibles. Se basa en innovadoras soluciones de un sistema de agentes que se comunican entre sí, para proporcionar información precisa sobre los lugares de estacionamiento disponibles. Además, *ABGS* utiliza un protocolo denominado *MQTT,* para llevar a cabo la transmisión instantánea de datos, lo que permite una comunicación efectiva entre los elementos del sistema (p. 5).

2.1.3.2. Estrategias de Gestión

2.1.3.2.1. Cantidad de Estacionamientos Utilizados

Según menciona Richard (2015), el control de la demanda de viajes y las opciones de transporte, son instrumentos clave para fomentar un progreso urbano sostenible y disminuir la dependencia de los automóviles. Con el propósito de cumplir con estos objetivos, se implementa una concientización para reducir el uso del automóvil, como tarifas de estacionamiento más altas o limitaciones en el acceso a determinadas zonas (p. 31).

Además, Richard (2015) menciona, que es importante brindar incentivos para utilizar otros tipos de transportes. Cada uso de estas alternativas, reduce la necesidad de ocupar plazas de estacionamiento y a su vez contribuyen a la disminución de liberación de gases, que llegan a afectar la calidad del aire (p. 32).

2.1.3.2.2. Eficiencia en el Uso del Espacio

Siguiendo la línea de pensamiento de Richard (2015), al hacer énfasis en la eficiencia de uso de espacios, nos referimos a las asignaciones de áreas específicas, para un cierto grupo de vehículos, donde la atribución de espacios se enfoca en determinadas categorías de vehículos, que permitan fomentar el uso de otros métodos de transporte, ofreciendo alternativas. Mejorando la ocupación de estacionamientos para vehículos convencionales como bicicletas y vehículos eléctricos (pp. 32–33).

Además, Richard (2015) menciona, que la rotación de espacios promueve la gestión de tiempo de uso de estacionamiento, de forma que los espacios más favorables sean ocupados rápidamente, lo que trae una mejora en la productividad y en la experiencia de los clientes (p. 34).

2.1.3.3. Estrategias de Oferta

2.1.3.3.1. Diferenciación entre aparcamiento y construcción

Según Tang (2023), se considera el diseño actual del desarrollo regional, para realizar la subdivisión de las áreas de estacionamiento y se optimiza las zonas, para logra la diferenciación de las mismas. Conforme avanza la urbanización y se produce la transformación del modelo de desarrollo económico, se observan divisiones funcionales en las ciudades. Las distintas actividades desempeñadas en las áreas urbanas, zonas

periféricas, lugares de desarrollo, áreas de transporte y lugares turísticos, ocasionan diferentes necesidades de estacionamiento en las diferentes regiones. Por lo tanto, al establecer los estándares, resulta necesario implementar el control por zonas, elaborar políticas basadas en el flujo de tráfico y ejercer un control del suministro de espacios de estacionamiento (p. 162).

2.1.3.3.2. Aprovechamiento de los recursos de aparcamiento

Siguiendo la línea de pensamiento de Tang (2023), se toman en consideración diversas estrategias para el aprovechamiento de los recursos de aparcamiento, como primera instancia se aconseja la construcción de aparcamientos tridimensionales en donde permitan la optimización de espacios. Como objetivo, se plantea ciertas medidas con la finalidad de utilizar los recursos en los alrededores de las distintas comunidades, donde, se prioriza zonas altas con demanda de estacionamientos, la transformación de áreas verdes, o el uso de terrenos dispersos (p. 162).

2.1.3.4. Automatización

2.1.3.4.1. Mecanismo de aparcamiento dinámico

Considerando lo propuesto por Sayarshad (2023), el mecanismo de estacionamiento dinámico, se presenta como la solución al problema que existe en área urbanas con referencia al tráfico, existe una pérdida de tiempo al buscar lugares de estacionamiento, que conlleva a un aumento en el tiempo de viaje, la emisión de contaminación al aire y al consumo de combustible. Con el fin de enfrentar esta situación, se plantean alternativas que solucionan el mecanismo de estacionamiento dinámico, se toma en consideración las preferencias del conductor, donde compiten por espacios de estacionamiento públicos y privados (p. 2).

Los modelos descritos por Sayarshad (2023), tienen en cuenta información acerca de la demanda de precios y la capacidad de los estacionamientos. También, se investigan enfoques apoyados en agentes y tecnologías, para mejorar la asignación de espacios de estacionamiento, basándose en el bienestar social (p. 2).

2.1.3.4.2. Comportamientos de aparcamiento de los Vehículos Autónomos

De acuerdo con lo mencionado por Sayarshad (2023), la demanda de estacionamientos para usuarios de vehículos autónomos (*AV*), es determinada por varios casos, como los precios de estacionamiento, la geografía, el nivel de movilidad y el costo del combustible. Es importante realizar un estudio detallado entre la demanda de estacionamiento y la ubicación en áreas urbanas. Se debe realizar una planificación urbana efectiva, demostrando el impacto en las estrategias de estacionamiento público, que reduzcan los tiempos de búsqueda y precios de estacionamiento (p. 3).

2.1.3.4.3. Detección de plazas de aparcamiento libres

En función de las aportaciones de Babic et al. (2019), la detección de plazas de estacionamiento disponibles, se maneja como un problema importante en las ciudades inteligentes. Gracias al uso de las tecnologías, se plantearon propuestas, en donde se aprovechan los sensores de los teléfonos inteligentes, para recolectar la información de disponibilidad de estacionamientos en tiempo real. Estas implementaciones de sistemas inteligentes, conforman un requisito fundamental, para lograr que las ciudades sean más eficientes, percatandose de proponer diversas soluciones particulares, como las cantidades de vehículos, el impacto ambiental y el bienestar de la sociedad (p. 1080).

2.1.3.4.4. Método de tarificación dinámica

Manteniendo lo mencionado por Babic et al. (2019), la estrategia de gestión sobre el estacionamiento, tiene como objetivo maximizar los ingresos y reducir los tiempos asociados a la búsqueda de los lugares de estacionamiento. Se busca como objetivo la estrategia en la oferta y demanda, aplicando un costo del estacionamiento, en consideración al cambio de las demandas y el número de plazas de estacionamiento que se encuentran disponibles (p. 1077).

2.2. Predicción científica

H0: El sistema *IoT* con *Machine Learning* no incide significativamente en el control de estacionamiento vehicular en el comercial Miñaca del cantón Santo Domingo.

H1: El sistema *IoT* con *Machine Learning* incide significativamente en el control de estacionamiento vehicular en el comercial Miñaca del cantón Santo Domingo.

3. METODOLOGÍA

3.1. Enfoque, diseño y tipo de investigación

Con el fin de llevar a cabo este trabajo de titulación, se optó por utilizar un enfoque cuantitativo, dado que se emplean tanto métodos para la recopilación como análisis de datos. Según Hernández y Mendoza (2018), este enfoque es adecuado cuando se desea calcular las magnitudes o incidencias de los fenómenos y validar las hipótesis (p. 6).

También, se consideró el diseño preexperimental porque tiene un grado de control mínimo, según lo expuesto por Hernández y Mendoza (2018), a un grupo se le somete a una evaluación inicial antes de aplicar el tratamiento experimental, luego se implementa el tratamiento y finalmente se lleva a cabo una segunda evaluación posterior en el mismo grupo (p. 163).

Igualmente, este proyecto de titulación se orientó hacia la investigación aplicada, cuyo objetivo es demostrar el control y gestión en la utilización de espacios de estacionamiento. Con esa consideración, se adopta lo estipulado por Baena (2017), donde menciona que la investigación aplicada tiene como propósito analizar un problema con el fin de tomar medidas o acciones al respecto (p. 18). Donde se enfoca en las oportunidades específicas para implementar las teorías generales y se dedica a abordar las necesidades que aparecen dentro de la sociedad y entre las personas.

Finalmente, se integra la investigación de campo, donde se utilizan técnicas de investigación con el propósito de recopilar datos, según lo señalado por Baena (2017), en donde su objetivo es recopilar y registrar metódicamente los datos relacionados con el tema seleccionado (p. 70).

3.2. Unidades de análisis

En este trabajo de titulación, se emplearon los datos proporcionados por el Comercial Miñaca, que cuenta con una población aproximada de 1500 vehículos al mes,

indistintamente de la cantidad de clientes (conductores o pasajeros), 2 colaboradores (guardias) y 1 administrativo. Como se contempla en la tabla 1, los tipos de vehículos seleccionados y la respectiva cantidad al mes, que pueden ocupar cualquiera de las 23 plazas de estacionamiento que dispone el Comercial Miñaca.

Tabla 1. Clientes del parqueadero del Comercial Miñaca

Categoría	Tipo de Vehículo	Cantidad/mes	Colaboradores (Guardias)	Administrativos	Total
1	Motocicleta	150	2	1	
2	Automóvil	750			
3	Camioneta	450			
4	Furgoneta	150			
Total		1500	2	1	1503

Fuente: Datos proporcionados por el Gerente del Comercial Miñaca

Con la finalidad de definir la muestra, se empleó un enfoque no probabilístico, específicamente el muestreo por conveniencia, seleccionando un conjunto no aleatorio de la población objetivo, obteniendo así una representación del grupo de estudio.

En base a lo mencionado por Hernández y Mendoza (2018), la elección del muestreo no probabilístico brinda a los investigadores la flexibilidad de elegir la muestra de acuerdo a las características y el contexto en el que se encuentra la investigación (p. 200). Se optó por este tipo de muestreo debido al tiempo limitado para presentar los resultados necesarios, de otra manera se habría optado por el muestreo aleatorio simple.

Como resultado, teniendo en cuenta tanto la población, el límite de tiempo y que las encuestas se aplicaron únicamente a los conductores que utilizaron las primeras 4 plazas del estacionamiento, y en las cuales se implementó los sensores para este estudio, de un total de 23 plazas, se optó por realizar 80 encuestas.

3.3. Técnicas e instrumentos de investigación

Las técnicas aplicadas a utilizar son las encuestas de opinión y la entrevista, según los recursos en línea de Hernández y Mendoza (2018), la encuesta se clasifica como un método cuantitativo que utiliza preguntas para recopilar información de un grupo específico

de individuos (p. 180). Asimismo, la entrevista cualitativa es definida por Hernández y Mendoza (2018), como un encuentro destinado a la conversación y la comunicación de información entre una persona (el entrevistador) y una o más personas (los entrevistados) (p. 449), que es utilizada para crear el diagrama de procesos (*BPMN*).

Finalmente, se optó por utilizar los cuestionarios, ya que se toma en cuenta como herramienta para la obtención de datos al realizar entrevistas y encuestas, de acuerdo a lo expresado por Hernández y Mendoza, (2018) que citó a Bourke, Kirby y Doran (2016), un cuestionario implica una serie de interrogantes acerca de una o más variables que se desean evaluar (p. 250).

Por lo tanto, es necesario resaltar que estas técnicas, se utilizan para recolectar información sobre las respuestas de la población, según Arévalo (2020), el contenido de los grupos focales puede ser examinado mediante un análisis temático, mientras que los datos de las encuestas pueden ser analizados mediante métodos estadísticos (p. 150). Se entiende por sesiones de grupos focales como una entrevista, dirigida por una persona entrenada y calificada, que se basa en un cuestionario previamente establecido.

3.4. Técnicas de análisis de datos

En base a la hipótesis, se utilizó un enfoque estadístico descriptivo, que permite analizar los datos recolectados y presentar un resumen de su comportamiento, se aplicó la herramienta Google Forms, para recopilar los datos mediante encuestas y Microsoft Excel, para realizar la tabulación de los datos y la generación de gráficos. Se representa los resultados de las encuestas en formato de porcentaje, facilitando así el proceso de análisis e interpretación.

Además, según Hernández y Mendoza (2018), con los datos codificados, corregidos y transferidos a una matriz, se puede emplear un análisis, utilizando un programa computacional (p. 312). Por lo tanto, se utilizó el *software IBM SPSS*, para realizar el análisis de las encuestas, mediante regresión logística binaria.

3.5. Operalización de las variables

Tabla 2. Internet de las Cosas – Variable Independiente

Conceptualizaciones	*Dimensión*	Indicadores	Preguntas	Herramienta
El *Internet de las Cosas* se considera un paradigma que equipa a los objetos con sensores, actuadores y procesadores, que permiten comunicarse entre sí para ejecutar funciones y cumplir un propósito significativo. Las tecnologías de *IoT* han logrado con el tiempo reducir significativamente el esfuerzo humano. Los dispositivos *IoT* están conectados a una red que transmite datos y agrega la capacidad de realizar tareas complejas, con varias tecnologías que trabajan juntas, que a su vez requieren un alto grado de inteligencia. El *Internet de las Cosas* se está implementado en diferentes áreas como lo es la educación, entretenimiento, vida social, conservación de la energía, domótica y sistemas de transporte (Gacovski, 2019, pp. 91–93).	Procesamiento	*Cloud Computing*	¿Qué tipo de teléfono móvil posee actualmente?	Encuesta a los Clientes
		Fog Computing	¿Con qué frecuencia utiliza el navegador web de su teléfono móvil?	Encuesta a los Clientes
		Edge Computing	¿Cuáles son las dificultades en la congestión o insuficiencia de plazas de estacionamiento vehicular, cuando la demanda es alta?	Entrevista al Gerente
	Protocolos	*MQTT*	¿Con qué frecuencia utiliza estacionamientos vehiculares inteligentes?	Encuesta a los Clientes
		CoAP		
		AMQP	¿Qué tan importante es para usted la facilidad de encontrar un estacionamiento vehicular?	Encuesta a los Clientes
		HTTP/REST	¿Cuál es su opinión acerca de las plataformas en línea que aportan valor a las decisiones de los locales comerciales?	Entrevista al Gerente
	Dispositivos Físicos	Sensores	¿Con qué frecuencia utiliza aplicaciones web para visualizar la disponibilidad de estacionamientos vehiculares antes de llegar a su destino?	Encuesta a los Clientes
		Actuadores	¿Cómo ayudaría el sistema de registro de información vehicular a mejorar la gestión del estacionamiento, para que sea más eficiente y beneficiosa?	Entrevista al Gerente
		Microcontroladores	¿Cuál es su perspectiva acerca del beneficio de recopilar información de los clientes mediante el uso de estacionamientos inteligentes?	Entrevista al Gerente
	Tecnologías de Acceso	*WPAN*	¿Qué medio utiliza para conectarse a internet desde su teléfono móvil?	Encuesta a los Clientes
		WLAN		
		WAN	¿Qué tan complicado es para usted tener acceso a Internet todos los días en su teléfono móvil?	Encuesta a los Clientes

Tabla 3. Machine Learning – Variable Independiente

Conceptualizaciones	*Dimensión*	Indicadores	Preguntas	Herramienta
El *Machine Learning* (*ML*) se considera un subconjunto de la *IA*, que consiste en entrenar algoritmos para que estos hagan predicciones o tomen decisiones basadas en los datos, para comprender el *ML*, se requiere conocimientos técnicos previos en áreas como las matemáticas, estadística e informática. En el *Machine Learning* los datos son cruciales, porque los algoritmos aprenden de ejemplos y patrones en los datos. Se considera que la calidad y la cantidad de datos, pueden influir significativamente en el rendimiento de un modelo de *ML* (Was, 2020, pp. 38–57).	Tipos de Aprendizaje	Aprendizaje Supervisado	¿Cómo analiza el desempeño y la capacidad actual de su parqueadero, en términos de cumplir con las necesidades de los clientes?	Entrevista al Gerente
		Aprendizaje No Supervisado	¿Cuál es la tendencia en términos de ocupación durante ciertas horas del día o días de la semana que aumenten la cantidad de vehículos?	Entrevista al Gerente
		Aprendizaje Semi-supervisado		
		Aprendizaje por Refuerzo		
	Algoritmos	Regresión Lineal	¿Cómo se evalúa en la actualidad la eficiencia del estacionamiento vehicular en su local comercial, en términos de la gestión de plazas disponibles?	Encuesta a los Clientes
		Regresión Logística		
		Arboles de Decisión	¿Qué relevancia tiene para usted incorporar sensores para recopilar datos, que impulsen mejoras en la gestión del estacionamiento vehicular?	Entrevista al Gerente
		KNN		
		Support Vector Machine		
		Gradient Boosting		
		Redes Neuronales		

Tabla 4. Control de Estacionamiento Vehicular – Variable Dependiente

Conceptualizaciones	*Dimensión*	Indicadores	Preguntas	Herramienta
El incremento del número de automóviles y la demanda de espacios de estacionamiento, han generado problemas con la congestión del tráfico y la seguridad vial, existen distintas soluciones para abordar este tema, por esta razón el implementar sistemas de gestión inteligente, consigue traer numeroso beneficios, entre ellos, la reducción del tiempo de búsqueda y la disminución de consumo de	Sistemas Inteligentes Centralizado	Sistema de información y guiado de aparcamientos (*PGIS*)	¿Con qué frecuencia ha experimentado problemas para encontrar estacionamiento vehicular en un local comercial?	Encuesta a los Clientes
		Sistema de información sobre la ocupación de aparcamientos (*COINS*)	¿Qué tan importante es para usted una aplicación que informe el estado del estacionamiento vehicular para reducir los tiempos de espera?	Encuesta a los Clientes
			¿Está de acuerdo con el método actual para encontrar estacionamiento vehicular accesible para personas con discapacidad?	Encuesta a los Clientes
		Sistema de guiado basado en agentes (*ABGS*)	¿Considera que un sistema de estacionamiento vehicular inteligente podría contribuir a la identificación más eficiente de plazas accesibles para personas con discapacidad?	Entrevista al Gerente
				Entrevista a los clientes

combustible, estas soluciones traen consigo un nivel de satisfacción para el cliente, ya que, brinda comodidad e información de los estacionamientos (Elsonbaty & Shams, 2020, pp. 55–56).			¿Cuáles son las medidas adoptadas para disponer de plazas de estacionamiento accesibles para personas con discapacidad?	
	Estrategias de Gestión	Cantidad de Estacionamientos Utilizados	¿Estás de acuerdo en visualizar los espacios disponibles del estacionamiento vehicular desde tu teléfono móvil?	Encuesta a los Clientes
		Eficiencia en el Uso del Espacio	¿Cuál sería el beneficio de utilizar la información del estacionamiento inteligente, para analizarla con inteligencia artificial y así obtener pronósticos?	Entrevista al Gerente
	Estrategias de Oferta	Diferenciación entre aparcamiento y construcción	¿Qué tan importante es para usted un sistema que le permita ver la disponibilidad de espacios en un estacionamiento vehicular antes de llegar a su destino?	Encuesta a los Clientes
		Aprovechamiento de los recursos de aparcamiento	¿Qué impacto tendría el despliegue de un sistema de estacionamiento vehicular, para reducir la congestión y maximizar el uso de plazas disponibles?	Entrevista al Gerente
			¿Cuál es la capacidad diaria del estacionamiento del local comercial, en términos de cantidad de vehículos que puede albergar?	Entrevista al Gerente
	Automatización	Mecanismo de aparcamiento dinámico	¿Qué tipo de automotor utiliza como medio de transporte diario?	Encuesta para Cliente
		Comportamientos de aparcamiento de los Vehículos Autónomos	¿Qué te parece el servicio de estacionamiento vehicular que ofrece el local comercial?	Encuesta para Cliente
		Detección de plazas de aparcamiento libres	¿Qué tan complicado es para usted encontrar espacios disponibles en un estacionamiento vehicular?	Encuesta para Cliente
		Método de tarificación dinámica	¿Cuáles son los procesos existentes para administrar el estacionamiento vehicular en su local comercial?	Entrevista al Gerente
			¿Cuál es el procedimiento actual que los vehículos deben seguir para estacionarse en el estacionamiento de su local comercial?	Entrevista al Gerente
			¿Cuál es el método actual de cobro en el estacionamiento del local comercial y cómo impacta en la experiencia de los usuarios?	Entrevista al Gerente

4. RESULTADOS

4.1. Resultado del primer objetivo específico

La validación de los instrumentos definidos para la encuesta dirigida a los clientes y la entrevista al gerente, se encuentran en el anexo III, que fue previamente analizada y evaluada por los docentes presentes en la tabla denominada “Expertos en evaluación de instrumentos para recopilación de datos” que se encuentra en el anexo IV.

4.1.1. Resultados de la entrevista dirigida al gerente del Comercial Miñaca del Cantón Santo Domingo

Pregunta 1: ¿Cómo analiza el desempeño y la capacidad actual de su parqueadero, en términos de satisfacer las necesidades de los clientes?

Respuesta: El desempeño actual en el parqueadero se lleva a cabo mediante métodos tradicionales, pero estamos trabajando en la implementación de tecnología, justamente en colaboración con ustedes, para ofrecer un servicio al cliente de mayor calidad.

Pregunta 2: ¿Cómo se evalúa en la actualidad la eficiencia del estacionamiento vehicular en su local comercial, en términos de gestión de espacios?

Respuesta: La eficiencia es constante, sin embargo, las plazas disponibles son limitadas, y contamos con una única entrada y salida que ralentiza un poco el proceso.

Pregunta 3: ¿Cuál es la capacidad diaria del estacionamiento del local comercial, en términos de cantidad de vehículos que puede albergar?

Respuesta: La capacidad diaria es de 50 vehículos aproximadamente en total.

Pregunta 4: ¿Cuáles son los procesos existentes para administrar el estacionamiento vehicular en su local comercial?

Respuesta: El proceso se realiza mediante un sistema manual que registra y contabiliza los vehículos a través de sus placas.

Pregunta 5: ¿Cuál es el procedimiento actual que los vehículos deben seguir para estacionarse en el estacionamiento de su local comercial?

Respuesta: El procedimiento se lleva a cabo emitiendo un ticket que se entrega al cliente. En el caso de que su vehículo esté estacionado delante de otro, se solicita la entrega de las llaves para facilitar el movimiento del automóvil cuando el vehículo ubicado detrás necesite salir.

Pregunta 6: ¿Cuál es el método actual de cobro en el estacionamiento del local comercial y cómo impacta en la experiencia de los usuarios?

Respuesta: El método de pago principal es en efectivo, ya que la mayoría de los usuarios tienen la capacidad de pagar de esta manera. Sin embargo, existen excepciones, ya que algunos usuarios preguntan si es posible realizar el pago a través de transferencia.

Pregunta 7: ¿Cuál es la tendencia en términos de ocupación durante ciertas horas del día o días de la semana que aumenten la cantidad de vehículos?

Respuesta: La ocupación generalmente se produce durante las horas pico, coincidiendo con las entradas y salidas laborales de las personas.

Pregunta 8: ¿Cuáles son las dificultades en la congestión o insuficiencia de espacios de estacionamiento vehicular, cuando la demanda es alta?

Respuesta: En momentos de alta ocupación, es posible que los clientes deban esperar hasta que haya un espacio disponible.

Pregunta 9: ¿Cuál es la utilidad que tienen las plataformas en línea para almacenar y procesar datos en su empresa?

Respuesta: Actualmente, contamos con una plaza accesible para personas con discapacidad en el parqueadero. Además, no hay gradas en el área, ya que todo es plano, facilitando así un acceso directo a las aceras.

Pregunta 10: ¿Cómo ayudaría el sistema de registro de información vehicular a mejorar la gestión del estacionamiento, para que sea más eficiente y beneficiosa?

Respuesta: El beneficio sería del 100%, ya que se proporciona información completa a la clientela, permitiéndoles conocer con precisión los espacios disponibles y utilizarlos de manera correspondiente.

Pregunta 11: ¿Qué relevancia tiene para usted incorporar sensores para recopilar datos, que impulsen mejoras en la gestión del estacionamiento vehicular?

Respuesta: La implementación de sensores nos proporciona una mayor eficiencia y eficacia, permitiendo que el parqueadero funcione de manera más rápida, reduciendo tiempos y ofreciendo un servicio más satisfactorio para el cliente.

Pregunta 12: ¿Cuál sería el beneficio al utilizar la información del parqueadero inteligente, para analizarla con inteligencia artificial y así obtener pronósticos sobre los clientes?

Respuesta: La implementación del sistema sería beneficiosa para todos los clientes, ya que la tendencia debería ser descongestionar los puntos críticos y mejorar la circulación para facilitar el desplazamiento de todos.

Pregunta 13: ¿Qué impacto tendría la implementación de un sistema de estacionamiento vehicular, para reducir la congestión y maximizar el uso de los espacios disponibles?

Respuesta: Sí, considero que esto proporcionará mayores beneficios para todos los clientes, especialmente para las personas con discapacidades.

Pregunta 14: ¿Cuál es su opinión acerca de cómo los estacionamientos vehiculares inteligentes pueden ser beneficiosos al recopilar información de los clientes?

Respuesta: Cada registro se convierte en un historial que proporciona información detallada, lo cual puede ser beneficioso para el cliente. En algún momento, este historial puede servir como punto de referencia en relación con una gestión anterior realizada.

Pregunta 15: ¿Considera que un sistema de estacionamiento vehicular inteligente podría contribuir a la identificación más eficiente de plazas accesibles para personas con discapacidad?

Respuesta: El aporte es ciento por ciento beneficioso, ya que a través del comercio electrónico en línea se ofrecen mayores beneficios para los clientes. Esto puede resultar útil para realizar transacciones, reservar turnos y optimizar el uso del tiempo de manera más eficiente.

Pregunta 16: ¿Cuál sería la relevancia que otorgaría la disponibilidad de plazas de estacionamiento accesibles, para personas con discapacidad, que utilizan el sistema de estacionamiento vehicular?

Respuesta: El beneficio de utilizar la información y trabajar con pronósticos mediante inteligencia artificial es ideal. Esto permitiría informar a los clientes y a nosotros mismos, lo que les permitiría tomar decisiones más informadas y oportunas.

4.1.1.1. Análisis e Interpretación de la entrevista al gerente del Comercial Miñaca del Cantón Santo Domingo

En base a la entrevista realizada al gerente del Comercial Miñaca, se observó que existen limitaciones en la disponibilidad de plazas de estacionamiento, ya que actualmente utilizan un proceso tradicional de registro con tickets, además de afrontar desafíos al posicionar vehículos uno frente al otro, debido al espacio limitado y a una sola entrada y salida, obteniendo como resultado, una alta ocupación durante las horas pico, destacando la necesidad de mejoras. Por otro lado, el estacionamiento cuenta con una única plaza para personas con discapacidad.

El gerente mostró una actitud positiva hacia la implementación de tecnologías, como sensores para visualizar espacios disponibles, además de recopilar información utilizada en el análisis y proceso de toma de decisiones empresariales, mediante inteligencia artificial. Estos elementos sugieren una disposición en adoptar soluciones innovadoras, para mejorar la satisfacción de los clientes.

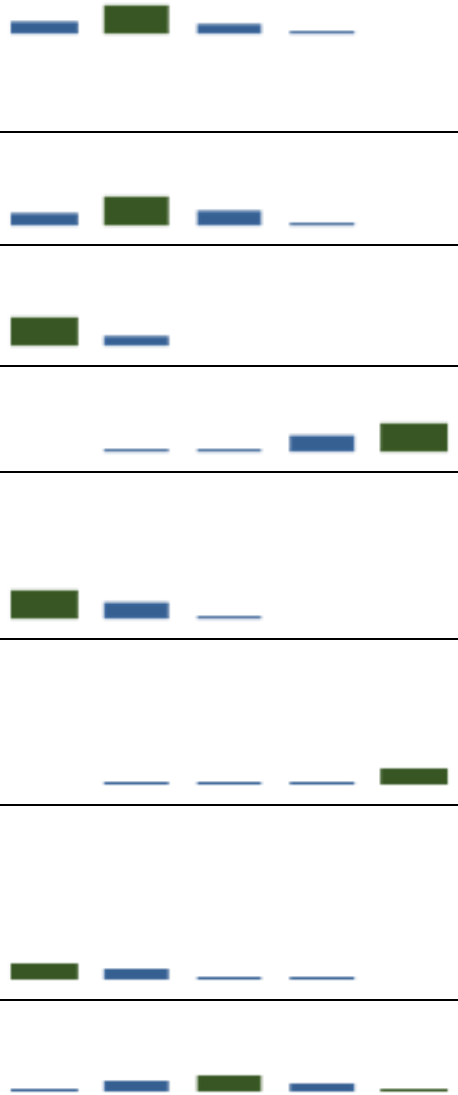
4.1.2. Resultados de encuesta dirigida a los clientes del Comercial Miñaca del Cantón Santo Domingo

En la tabla 5, se exhiben los resultados porcentuales, derivados de las encuestas efectuadas durante la fase de *pre test*, presentando las interrogantes con sus escalas asignadas y los porcentajes respectivos.

Tabla 5. Datos derivados de la encuesta del pre test

N°	***Preguntas***	***Escala y %***					***Figuras***
1	¿Qué tipo de teléfono móvil posee actualmente?	***Teléfono Inteligente***	***Teléfono Básico***	***Teléfono Convencional***	***Ninguno***		
		98,80%	1,20%	0,00%	0,00%		
2	¿Qué medio utiliza para conectarse a internet desde su teléfono móvil?	***Internet Prepago (Recargas)***	***Internet Pospago (Plan)***	***Internet mediante WiFi***	***Ninguno***		
		32,50%	53,80%	13,70%	0,00%		
3	¿Con qué frecuencia utiliza el navegador web de su teléfono móvil?	***Muy Frecuentemente***	***Frecuentemente***	***Ocasionalmente***	***Raramente***	***Nunca***	
		38,80%	41,30%	16,20%	3,70%	0,00%	
4	¿Estás de acuerdo en visualizar los espacios disponibles del estacionamiento vehicular desde tu teléfono móvil?	***Totalmente de acuerdo***	***De acuerdo***	***Neutral***	***Poco de acuerdo***	***Nada de acuerdo***	
		67,50%	30,00%	1,20%	0,00%	1,20%	
5	¿Qué tan complicado es para usted tener acceso a Internet todos los días en su teléfono móvil?	***Muy complicado***	***Complicado***	***Neutral***	***Poco complicado***	***Nada complicado***	
		0,00%	2,50%	22,50%	42,50%	32,50%	
6	¿Está de acuerdo con el método actual para encontrar estacionamiento vehicular accesible para personas con discapacidad?	***Totalmente de acuerdo***	***De acuerdo***	***Neutral***	***Poco de acuerdo***	***Nada de acuerdo***	
		3,80%	16,30%	26,30%	45,00%	8,80%	
7		***Muy Frecuentemente***	***Frecuentemente***	***Ocasionalmente***	***Raramente***	***Nunca***	

	¿Con qué frecuencia ha experimentado problemas para encontrar estacionamiento vehicular en un local comercial?	23,80%	55,00%	20,00%	1,20%	0,00%
8	¿Qué tan complicado es para usted encontrar espacios disponibles en un estacionamiento vehicular?	***Muy complicado***	***Complicado***	***Neutral***	***Poco complicado***	***Nada complicado***
		22,50%	51,20%	25,00%	1,20%	0,00%
9	¿Qué tan importante es para usted la facilidad de encontrar un estacionamiento vehicular?	***Muy Importante***	***Importante***	***Neutral***	***De poca importancia***	***Sin importancia***
		76,30%	23,80%	0,00%	0,00%	0,00%
10	¿Con qué frecuencia utiliza estacionamientos vehiculares inteligentes?	***Muy Frecuentemente***	***Frecuentemente***	***Ocasionalmente***	***Raramente***	***Nunca***
		0,00%	5,00%	5,00%	33,80%	56,30%
11	¿Qué tan importante es para usted una aplicación que informe el estado del estacionamiento vehicular para reducir los tiempos de espera?	***Muy Importante***	***Importante***	***Neutral***	***De poca importancia***	***Sin importancia***
		60,00%	36,20%	3,70%	0,00%	0,00%
12	¿Con qué frecuencia utiliza aplicaciones web para visualizar la disponibilidad de estacionamientos vehiculares antes de llegar a su destino?	***Muy Frecuentemente***	***Frecuentemente***	***Ocasionalmente***	***Raramente***	***Nunca***
		0,00%	2,50%	2,50%	6,30%	88,80%
13	¿Qué tan importante es para usted un sistema que le permita ver la disponibilidad de espacios en un estacionamiento vehicular antes de llegar a su destino?	***Muy Importante***	***Importante***	***Neutral***	***De poca importancia***	***Sin importancia***
		58,80%	36,20%	3,70%	1,20%	0,00%
14	¿Qué te parece el servicio de estacionamiento vehicular que ofrece el local comercial?	***Excelente***	***Buena***	***Regular***	***Mala***	***Pésima***
		2,50%	30,00%	45,00%	21,30%	1,20%
15		***Motocicleta***	***Automóvil***	***Camioneta***	***Furgoneta***	



	¿Qué tipo de automotor utiliza como medio de transporte diario?	10,00%	50,00%	27,50%	2,50%		
16	¿En qué rango de edad te encuentras?	***18-25***	***26-30***	***30-40***	***40 o más***		
		27,50%	33,80%	21,30%	17,50%		
17	¿Cómo te identificas en cuanto a género?	***Femenino***	***Masculino***				
		38,80%	61,30%				
18	¿Presentas alguna discapacidad?	***Auditiva***	***Visual***	***Física***	***Otro***	***Ninguna***	
		0,00%	0,00%	3,80%	2,50%	93,80%	

4.1.2.1. Análisis e Interpretación de los resultados de las encuestas dirigidas a los clientes del Comercial Miñaca del Cantón Santo Domingo

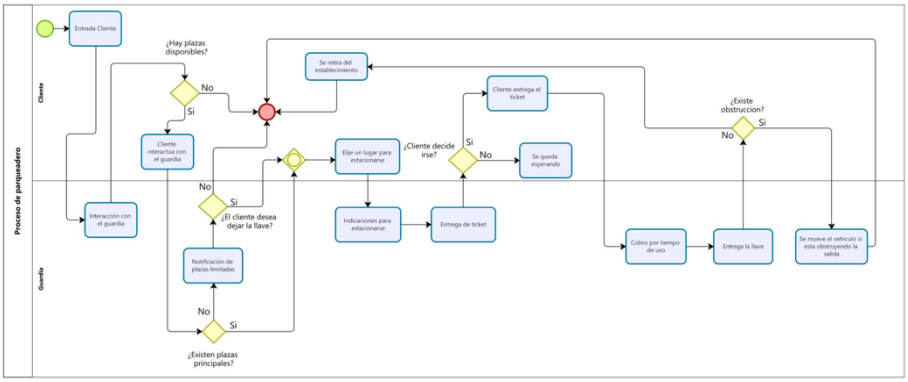
Mediante las encuestas que se realizaron a los 80 clientes del comercial Miñaca, se obtuvieron resultados pertinentes al *pre test*, donde se destacó que la mayor parte de los clientes están totalmente de acuerdo, en conocer el estado de las plazas de estacionamiento, y de la misma forma están a favor de visualizar los espacios disponibles desde sus teléfonos inteligentes. Además, se consideró importante el despliegue de una aplicación que le otorgue la información sobre el estado del estacionamiento, satisfaciendo así los tiempos de espera.

Por otro lado, los clientes reflejaron en la encuesta, la importancia de tener a la mano un sistema que les permita verificar la disponibilidad y cualquier dato informativo sobre el estado de las plazas, de la misma manera, se evidenció la aceptación por parte de los clientes, en la incorporación de sistemas inteligentes en los estacionamientos vehiculares. Estos factores fueron cuidadosamente considerados en la formulación de las preguntas.

4.1.2.2. Gráfico de flujo de actividades en el proceso de estacionamiento de vehículos del Comercial Miñaca del Cantón Santo Domingo

Se identificó el flujo de actividades que realiza el estacionamiento, desde la entrada de los vehículos, la interacción con el guardia y la salida de los vehículos. Con la herramienta de *BPMN* de *Bizagi,* se pudo determinar en la figura 4, el accionar de cada involucrado en el proceso.

Figura 4. Gráfico de flujo de actividades BPMN (Bizagi, 2023)



4.2. Resultado del segundo objetivo específico.

4.2.1. Tecnologías y Herramientas

4.2.1.1. Framework Frontend

Actualmente existen muchos *frameworks* y bibliotecas del lado del *frontend*, que permiten crear interfaces de usuario dinámicas y además amigables con el uso. Con la finalidad de elegir el *framework* que más se acople al desarrollo, se realizó un análisis comparativo entre los tres *frameworks* de desarrollo web más conocidos.

Tabla 6. Comparativa Frameworks Frontend

Característica	***Frameworks***		
	React [a]	***Vue.js*** [b]	***Angular*** [c]
Desarrollador	Meta	Evan You	Google
Lenguaje de Programación	JavaScript	JavaScript	JavaScript, TypeScript
Manipulación del DOM	Virtual DOM	Virtual DOM	DOM Real
Sintaxis de Plantilla	JSX (JavaScript XML)	HTML con directivas	HTML con directivas
Gestión de Estado	Context API	Pinia	RxJS
Organización	Componentes	Componentes	Componentes
Sistema de Enrutamiento	React Router	Vue Router	Angular Router
Estilos de Componentes	CSS-in-JS	CSS Localizado, Módulos CSS	Estilos de Componentes
Tools	Create React App	Vue CLI	Angular CLI
Gestor de Paquetes	NPM, Yarn	NPM, Yarn, pnpm, bun	NPM

Nota: Obtenido de: [a]Meta Open Source (2023), [b]You (2023), [c]Google(2023a)

En la tabla 6 se visualiza las diferentes características de cada *framework* y biblioteca. Se decidió utilizar la biblioteca de *React*, gracias a su capacidad para trabajar con componentes, poder incorporar código *HTML* dentro de archivos *JavaScript*, incorporar estilos *CSS* en el mismo archivo y por su rapidez en el proceso de desarrollo.

4.2.1.2. Framework Backend

En el proceso de desarrollo del *backend*, que implica la definición de la lógica de negocio y la organización de las funcionalidades que permiten a la aplicación alcanzar sus objetivos de desarrollo, se optó por utilizar un *framework* que agilice el proceso de desarrollo y que permita crear un sistema robusto.

Tabla 7. Comparativa Frameworks Backend

Característica	***Frameworks***		
	Django [a]	***Next.js*** [b]	***Laravel*** [c]
Lenguaje Principal	Python	JavaScript	PHP
Tipo de Framework	Framework web en Python	Framework basado en React	Framework web en PHP
Sistema de Enrutamiento	Incorporado	Sistema basado en archivos	Incorporado
Soporte de Bases de Datos	Django ORM	N/A	Eloquent ORM
Motor de Plantillas	Lenguaje de Plantillas de Django	JSX	Motor de Plantillas Blade
Autenticación	Sistema de autenticación incorporado	Autenticación personalizable	Sistema de autenticación incorporado
API	Django REST Framework	API Routes	API Laravel

Nota: Obtenido de: [a] Django(2023), [b] Vercel (2023) ,[c] Laravel (2023)

Entre los distintos *frameworks* de *backend* presentados en la comparativa que se visualiza en la tabla 7, se optó por realizar el desarrollo con Django, ya que es un *framework* completo que incorpora autenticación de usuarios y un ORM para el manejo de los modelos sobre la base de datos relacional. Además de que se utiliza Python, lo que apoya en la integración de funcionalidades de *Machine Learning*.

4.2.1.3. Placas de Desarrollo

Para el desarrollo y programación de los sensores que permite detectar la presencia de un vehículo, se utilizó una placa de desarrollo con un microcontrolador que cumpla las necesidades requeridas para el proyecto, en la tabla 8, se analiza una comparación de los distintos microcontroladores existentes en el mercado para el desarrollo de prototipos.

Tabla 8. Comparativa Placas de Desarrollo

Característica	***Placas de Desarrollo***		
	ESP8266EX [a]	***Arduino MKR1000*** [b]	***Raspberry Pi Pico*** [c]
Microcontrolador	ESP8266EX	SAMD21 Cortex-M0+	RP2040
Velocidad del reloj	160 MHz	48 MHz	133 MHz

Memoria	4 MB de flash, 50 KB de RAM	256 KB de flash, 32 KB de SRAM	2 MB de flash
Pines GPIO	17	8 digitales	26
Entradas Analógicas	1	7	3
Tecnología Inalámbrica	Wi-Fi	Wi-Fi	N/A
Herramientas	N/A	Arduino IDE, CLI, Web Editor	Python, C/C++
Interfaz de Comunicación	N/A	Micro USB	Micro USB
Voltaje de Operación Normal	2.5 – 3.6V	5 - 5.5V	1.8 - 3.3V

Nota: Obtenido de: [a] Espressif Systems(2023), [b] Arduino(2023), [c] Raspberry Pi (2023)

Para el diseño de los sensores de detección vehicular, se eligió la placa de desarrollo *NodeMCU* con el *ESP8266*. Se basó en la búsqueda de una opción que se ajuste al presupuesto del prototipo, ofreciendo un equilibrio entre costo y rendimiento. El microcontrolador satisface las necesidades técnicas que permitió el óptimo funcionamiento de los sensores. Siendo esta opción la más óptima en rendimiento a comparación con las otras alternativas. Además de tener la posibilidad de programar con *MicroPython*, facilitando así el proceso de desarrollo a quienes ya están familiarizados con el lenguaje de programación.

4.2.1.4. Sensores

Para la detección de obstáculos, se utilizó un sensor de proximidad en conjunto con el microcontrolador, que juntos formarían un sensor de detección vehicular, para la elección del sensor, se realizó una tabla comparativa entre un sensor de proximidad ultrasónico y uno infrarrojo.

Tabla 9. Comparativa Sensores

Característica	***Sensores***	
	Sensor Ultrasónico [a]	***Sensor Óptico*** [b]
Modulo	HC-SR04	FC-51
Funcionamiento	Medición mediante ultrasonido	Transmisión y Recepción IR
Tipo de Salida	Pulso	IR
Rango de medición	2cm a 400cm	2cm a 30cm

Precisión	Precisión en la detección de objetos	Sensibilidad ajustable
Polvo	Inmune	N/A
Voltaje	5V	3V - 6V
Corriente	15 mA	23 mA

Nota: Obtenido de: [b] AG Electrónica(2019), [a] Morgan (2014)

En base a la información comparativa mostrada en la tabla 9, se optó por utilizar un sensor ultrasónico debido a la capacidad que tiene en la detección de obstáculos a mayor distancia, en comparación con el sensor óptico, además de ser inmune al polvo, agua y al tipo o color del material que está detectando, siendo ideal para la detección de vehículos en estacionamientos que tengan techo o estén en el exterior.

4.2.1.5. Base de Datos

Con el fin de desarrollar el sistema *IoT,* se realizó un análisis de las bases de datos relacionales y no relacionales, y poder identificar cual cubre las necesidades del proyecto.

Tabla 10. Comparativa Tipos de Base de Datos

Característica	***Tipos***	
	SQL [a]	***NoSQL*** [b]
Modelo de Datos	Relacional	Documento, Clave-Valor, Gráfico
Esquema	Esquema fijo	Esquema dinámico
Lenguaje de Consulta	SQL	Funciones que varían según el SGBD
Transacciones	Si	No
Normalización	Datos normalizados para consistencia	Desnormalizados para rendimiento
Observaciones: El ORM de Django tiene mayor soporte con SQL, mientras que con NoSQL tiene un soporte limitado.		

Nota: Obtenido de: [a] Kline y Kline (2001), [b] Sadalage y Fowler (2013)

Sobre la comparativa visualizada en la tabla 10, se eligió *SQL*, porque tiene alta compatibilidad con el *ORM* del *framework Django*, ya que trabaja con datos estructurados en un modelo relacional, maneja un esquema de datos fijo y soporta transacciones.

Tabla 11. Comparativa SGBD

Característica	**SGBD**		
	MySQL [a]	**SQL Server** [b]	**PostgreSQL** [c]
Versión	8.0	2022	16

ACID	Sí	Sí	Sí
Tipos de Datos	Numérico, Caracteres, Temporales	Numéricos, Caracteres, Fecha, Binario	Numeric, Monetary, Character, Binary, Boolean, Date/Time, Geometric, Network Address
Motores de Almacenamiento	InnoDB	N/A	N/A
Agrupamiento	Cluster	Database Files y Filesgroups	N/A
Copia de Seguridad	Mysqldump	SQL Server backups	SQL dump, File system level backup, Continuous archiving
Disponibilidad Alta	MySQL ClusterSet, ReplicaSet, Group Replication, Router, Shell	Disponibilidad en Grupos	Load Balancing, Redundancia
Seguridad	Encriptación, Validación de Contraseña	Protección de fila, columna, Encriptación de archivos	User Authentication, Access Control
Transacciones	Sí	Sí	Sí
Replicación	Sí	Sí	Sí
Particionamiento	Sí	Sí	Sí
Soporte JSON	Sí	Sí	Sí
Soporte XML	Sí	Sí	Sí

Nota: Obtenido de: [a] Oracle (2023), [b] Microsoft(2023c), [c] The PostgreSQL Global Development Group (2023)

En base a la información comparativa de la tabla 11, se optó por utilizar el sistema gestor de base de datos *MySQL*, porque la versión 8.0 es de código abierto y tiene una gran comunidad de soporte. Además, tiene alta compatibilidad con el *ORM* del *framework Django,* ya que cumple con los tipos de datos necesarios para el desarrollo de la aplicación. Por otro lado, soporta transacciones, replicación y particionamiento, además de formatos JSON y XML, que permite utilizar la base de datos en otros sistemas.

4.2.1.6. Algoritmos de Machine Learning

La incorporación de *Machine Learning* en el desarrollo del sistema de *IoT* requiere que se elija un algoritmo, en este caso como el resultado de la funcionalidad de predicción es de dos clases, se procedió a realizar un análisis comparativo entre algoritmos de clasificación.

Tabla 12. Comparativa Algoritmos de Machine Learning

Característica	***Algoritmos***	
	Regresión Logística [a]	***Árboles de Decisión*** [b]
Aprendizaje	Supervisado	Supervisado
Tipo	Clasificación	Clasificación y Regresión
Funcionamiento	Se utiliza una matriz de coeficiente.	Uso de conjunto de reglas de decisión
Linealidad	Clasificador log-lineal	N/A
Valores Categóricos	Siempre se espera objectos Categóricos	No Soporta
Escalabilidad	Generalmente escala bien con un gran número de características	Limitación con un número mínimo de muestras.
Facilidad de Implementación	Relativamente fácil	Relativamente fácil
Rendimiento	Regularización mejora la estabilidad numérica	El costo de usar el árbol es logarítmico
Caso Binario	Si	No
Caso Multinomial	Si	No

Nota: Obtenido de: [a] scikit-learn developers(2023b), [b] scikit-learn developers (2023a)

Con la comparativa de la tabla 12, se optó por el algoritmo de regresión logística, ya que el modelo es supervisado, de clasificación y es relativamente fácil de implementar. Además, es eficaz en cumplir el propósito de obtener predicciones de dos clases, teniendo un número limitado de datos de entrada, que se puede escalar cuando existan más datos en el futuro.

4.2.1.7. Cloud Computing

Para llevar a cabo el despliegue a producción del *frontend*, *backend* y base de datos, fue ideal seleccionar un proveedor de computación en la nube. Este brindó todos los servicios necesarios y además se ajustó al presupuesto económico de un proyecto.

Tabla 13. Comparativa Cloud Computing

Criterio	***Proveedores***		
	Digital Ocean [a]	***GCP*** [b]	***Microsoft Azure*** [c]

Tipo de Servicios	PaaS	IaaS/PaaS	IaaS/PaaS/SaaS
Servicios de VM	Droplets	Compute Engine, App Engine	Máquinas Virtuales
Almacenamiento	Volumen Block, Space Object	Cloud Storage	Blob Storage, Disk Storage
Redes	DNS, IP Reservadas, Load Balancers, Cloud Firewall, IPv6, VPC	VPC, Balanceo de Carga	Red Virtual, Balanceador
Bases de Datos	Administradas, PostgreSQL, MySQL, Redis, MongoDB, Kafka	Cloud SQL	Azure SQL
Contenedores	Si	Kubernetes, Docker, Conteiners on Copute Engine	Azure Kubernetes Service
Monitoreo y Gestión	Si	Si	Azure Monitor
Seguridad	Si, soluciones de tercero	Security Center, IAM	Microsoft Defender

Nota: Obtenido de: [a] DigitalOcean(2023), [b] Google (2023b), [c] Microsoft (2023a)

En base a la información comparativa entre los distintos proveedores de computación en la nube que se redacta en la tabla 13, se optó por utilizar *Digital Ocean*. Esta elección se fundamenta en que *Digital Ocean* ofrece servicios *PaaS*, lo que facilita el despliegue de aplicaciones y bases de datos de manera sencilla, sin la necesidad de administrar la infraestructura y la configuración de los servidores. Asimismo, los servicios tienen precios que se ajustan al presupuesto de un proyecto.

4.2.1.8. Edge Computing

En la infraestructura local que trabaja como *Edge Computing*, fue necesario configurar un servidor que se encargue de la recolección de información de los sensores vehiculares, esta información es procesada y enviada al *Cloud Computing*, para conseguir esto se necesitó configurar un servidor con un sistema operativo.

Tabla 14. Comparativa Edge Computing

Característica	***Sistemas Operativos***		
	Ubuntu Server [a]	***CentOS*** [b]	***Windows Server*** [c]
Distribución Base	Debían	RHEL	Windows
Gestión de Paquetes	APT	YUM	Instalador de Windows

Formato de Paquete	deb	rpm	msi, misx, exe
Ciclo de Lanzamiento	Regular LTS cada 2 años	N/A	N/A
Entorno Gráfico	N/A	Si	Si
Seguridad	AppArmor	SELinux, iptables	Windows Defender, BitLocker
Sistema de Iniciación	systemd	N/A	N/A
CLI	Bash	N/A	PowerShell

Nota: Obtenido de: [a] Canonical (2023), [b] The CentOS Project (2023), [c] Microsoft (2023d)

A través de esta comparativa, se ha concluido que la aplicación de *Ubuntu Server* es ideal para la implementación de las herramientas necesarias en el servidor local del *Edge Computing*, utilizado para la recolección de datos de los sensores. Esto se debe a la facilidad de uso que ofrece *Ubuntu Server* y a su gestor de paquetes, que incluye todas las herramientas necesarias como *Node-RED* y el *broker* Mosquitto *MQTT*. Además, *Ubuntu Server* cuenta con soporte oficial de larga duración y una extensa documentación.

4.2.2. Arquitecturas

4.2.2.1. Arquitectura de Desarrollo

La arquitectura en el desarrollo de una aplicación web es importante, ya que permite sostener la estructura de los diferentes módulos de la aplicación, de manera lógica y segura, para esto se desarrolló un análisis comparativo que ayudó a considerar la mejor opción entre las arquitecturas.

Tabla 15. Comparativa Arquitectura de Desarrollo

Aspectos	***Arquitecturas***	
	Arquitectura Modelo-Vista-Plantilla (MVT) [a]	***Arquitectura de n Capas*** [b]
Capas	Modelo	Presentación
	Vista	Negocio
	Plantilla	Persistencia
	N/A	Base de Datos
Funcionalidades por capa	Modelo: Estructura y manipula los datos	Capa de Presentación: Se encarga de gestionar toda la interfaz y la comunicación con el navegador.
	Vista: Encapsula la lógica para procesar solicitudes de usuarios.	Capa de Negocio: Su responsabilidad es, dictar reglas de negocio.

	Plantilla: Define cómo se presenta la información.	Capa de Persistencia: Ejecutar la lógica lógica empresarial con los datos
	N/A	Capa de Base de Datos: Capa sólo para recuperar o guardar información de la base de datos

Nota: Obtenido de: [a] Django (2023), [b] Richards (2022)

Con la comparativa de la tabla 15, se escogió el *MVT* como arquitectura de desarrollo, ya que es ideal para trabajar con el *framework* de *backend Django* que se utilizó para la lógica de negocio del sistema, considerando que las capas de modelo y vista se ejecutaron en el *backend* y la capa de plantilla se desplegó como el *frontend*.

4.2.2.2. Arquitectura de IoT

En el desarrollo del sistema *IoT*, se tomaron en cuenta distintas arquitecturas para implementar el sistema de forma lógica y organizada. Para lograrlo, se elaboró una tabla comparativa que ayudó en la elección de la mejor solución.

Tabla 16. Comparativa Arquitectura de IoT

Característica	***Arquitecturas***	
	IoT de Tres Capas [a]	***Modelo de Referencia IoT de Cisco*** [b]
Descripción General	Consta de las capas de Percepción, Red y Aplicación.	Introducida por Cisco como un modelo de referencia para IoT.
Capas	Percepción	Física
	Red	Conectividad
	Aplicación	Edge Computing
	N/A	Acumulación de Datos
	N/A	Abstracción de Datos
	N/A	Aplicación
	N/A	Capa de Colaboración y Procesos
Conectividad	La capa de Red se encarga de la conectividad entre dispositivos inteligentes.	Conectividad centrada en la capa física.
Procesamiento de Datos	La capa de Aplicación se encarga del procesamiento y manejo de información.	Análisis y procesamiento avanzado en la Capa de Edge Computing.
Seguridad	N/A	Seguridad por niveles que se aplique en los puntos de transición entre niveles.
Escalabilidad	N/A	Descomponer el problema de IoT en partes más pequeñas.

Flexibilidad	La capa de Red es capaz de incluir funcionalidades como la computación en la nube	Las piezas puedan ser suministradas por distintos distribuidores.
Tecnologías	Ethernet, Wifi, Wifi-Max, Zigbee, BLE	MQTT, CoAP, IPv6, 6LoWPAN.

Nota: Obtenido de: [a] Tonato y Sinche (2022), [b] Hanes et al. (2017)

Con la comparativa de la tabla 16, se optó por utilizar el modelo de referencia de *IoT* de *Cisco*. Esta arquitectura permite segmentar lógicamente cada funcionalidad del sistema de estacionamiento inteligente en 7 capas, además, de ser escalable en la cantidad de dispositivos físicos, en este caso en la cantidad de sensores de vehículos.

4.3. Resultado del tercer objetivo: Sistema *IoT* con *Machine Learning.*

4.3.1. Nomenclatura y Logotipo

Se decidió usar la nomenclatura "*ParkIoT*", como acrónimo para el sistema de gestión de estacionamientos del cual nace por dos palabras esenciales: "*Park*" que se origina por la palabra *parking* e "*IoT*" que representa el intercambio de datos con otros dispositivos a través del Internet de las Cosas, el logotipo se observa en la figura 5.

Figura 5. Logo del Sistema generado en la aplicación Canva (Canva, 2023)

4.3.2. Marco de Trabajo Scrum

Para la construcción del sistema que integra Internet de las Cosas (*IoT*) y *Machine Learning*, al control de estacionamiento vehicular, se optó por utilizar el marco de trabajo ágil *Scrum*. Según Schwaber y Sutherlan (2020), este facilita a los equipos en el proceso de gestión y estructuración del trabajo a través de principios y prácticas, considerándose como un enfoque ágil que facilita a los individuos, equipos y entidades en el proceso de generar valor mediante la creación de soluciones adaptables para desafíos complejos (p. 3). Todo

esto se lleva a cabo con el propósito de lograr de manera más eficiente los objetivos previamente establecidos.

4.3.3. Sprint 1

4.3.3.1. Sprint 1 – Planificación *(Sprint Planning)*

En base a lo mencionado por Schwaber y Sutherland (2020), la fase inicial en el procedimiento de desarrollo consiste en la planificación, considerando al *sprint* como un evento con duración de un mes o menos y que cada *sprint* empieza una vez termine el anterior (pp. 8-9). Por ello, se considera que esta etapa es esencial para conseguir el éxito en la implementación del sistema. En relación con el marco de trabajo *Scrum*, se llevaron a cabo tres *sprints*, fundamentados en las necesidades y prioridades que tenía el cliente.

4.3.3.1.1. Roles

En *Scrum*, tal como se indica en la tabla 17, se definieron roles particulares para el desempeño del marco de trabajo. Según Palacio (2022), el *product owner* se le denomina al propietario del producto favoreciendo el cumplimiento de los objetivos. En segunda instancia, el *scrum master* se asegura que los miembros del equipo se adecuen al marco de trabajo de manera efectiva. En última instancia, es tarea del equipo de desarrollo encargarse de la creación y finalización del producto terminado (pp. 34-36).

Tabla 17. Distribución de Roles

Persona	***Roles***	***Área***
Lic. Arturo Miñaca	*Dueño el Producto (Product Owner)*	Gerente Comercial Miñaca
Mg. Willian Ocampo	*Scrum Master*	Docente de la PUCESD
Patricio Cruz	Equipo de Desarrollo y Testeo	Desarrollador *Frontend* y *Backend*
David Carrascal	Equipo de Desarrollo y Testeo	Diseñador *Web* y *Tester Web*

4.3.3.1.2. Modelo Vista Template

Se implementó la arquitectura MVT, según el análisis realizado, en el procedimiento de desarrollo del sistema de control de estacionamiento. En base a lo mencionado por Vidal-Silva et al. (2021), el modelo representa la lógica y los datos de la aplicación, responsable de gestionar la base de datos. La vista es el enlace que existe entre el modelo

y el *template*, que decide la información que se muestra. Finalmente, el *template* expone la presentación visual de los datos, a su vez, facilita que los usuarios interactúen con el sistema (p. 88).

4.3.3.1.3. Modelo de Referencia de IoT de Cisco

Para la infraestructura de *IoT*, se implementó el modelo de capas de referencias de *IoT* de *Cisco* que consta de 7 capas, según lo definido por Hanes (2017), la capa 1 representa los sensores, actuadores y controladores físicos; la capa 2 representa la conectividad, como puntos de acceso o puertas de enlace que trabajen con el modelo *TPC/IP*; la capa 3 representa el *Egde Computing*, como los servidores o centro de datos locales que se encargan de pre procesar la información (pp. 35-37).

Por otro lado, desde el punto de vista en la nube, de acuerdo con Hanes (2017), la capa 4 representa la acumulación de datos, como el almacenamiento de la información en la bases de datos; la capa 5 es la abstracción de datos; la capa 6 es la aplicación y se trata de utilizar los datos generados por los sensores; la última capa es la capa 7 de colaboración y proceso, que trata sobre como consumir y compartir la información de las aplicaciones, por ejemplo para realizar tomas de decisiones en el negocio en base a la información generada (p. 38).

4.3.3.1.4. Parametrización

Para el entendimiento entre el equipo de desarrollo, se definió una parametrización de los diferentes objetos que se manipulan en el código fuente, esto permitió tener un código más limpio y legible, por lo tanto, la estructura de cada elemento se visualiza en la tabla 18.

Tabla 18. Parametrización

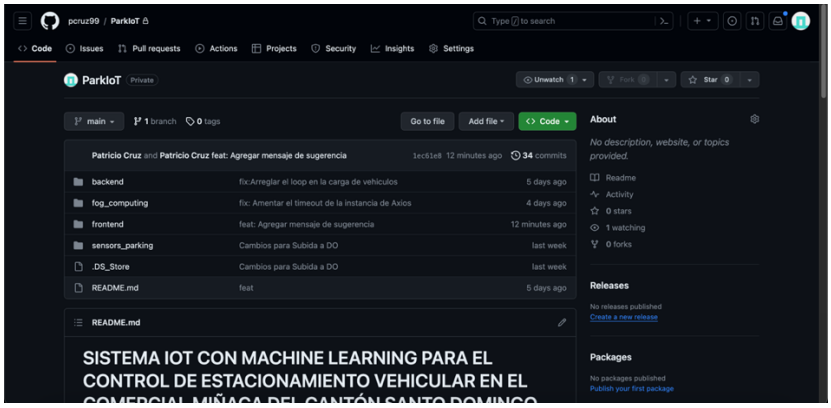
Template	***Modelo***	***Vista***
Paginas	**Métodos**	**Clases**
Show*Nombre*	create_user()	*Nombre*View
Componentes	create_superuser()	*Nombre*ViewSet
Base*Nombre*	create_qrcode()	**Funciones**
Hooks		*nombre*_view
Action*Nombre*		**Serializadores**
		*Nombre*Serializer

Idioma: español **Variables** *nombre* **Constantes** *NOMBRE*	**Idioma:** español **Variables** *nombre* **Constantes** *NOMBRE*	**Idioma:** español **Variables** *nombre* **Constantes** *NOMBRE*

4.3.3.1.5. Control de Versiones

Para controlar y tener una buena gestión de versiones, se utilizó la plataforma *GitHub*, la cual permitió colaborar de manera simultánea en el desarrollo, facilitando la manipulación del repositorio y sus diferentes versiones de forma remota. Del mismo modo, brinda la posibilidad de compartir y revisar el trabajo de otros de manera productiva, fomentando la colaboración en una plataforma segura, tal como se ilustra en la figura 6.

Figura 6. Plataforma GitHub para la gestión de versiones (Microsoft, 2023b)



4.3.3.1.6. Product Backlog

En torno a lo mencionado por Schwaber y Sutherland (2020), la elaboración del *product backlog* facilita la organización de las funcionalidades del sistema, en donde todas las tareas deben listarse para que el equipo de desarrollo tenga una idea general de lo que se va a realizar (p. 11). En consecuencia, se realiza una reunión con el *product owner* dado que él está familiarizado con los procesos de su negocio, como se destaca en la tabla 19, se detalla las historias de usuarios con la estimación respecto a la complejidad y en la tabla 20 se definen las historias técnicas.

Tabla 19. Backlog del Producto (Product Backlog)

N°	***Historia de Usuarios***	***Estimación***	***Prioridad de Negocio***	***Riesgo en Desarrollo***
1	Inicio de Sesión *(Login)*	3	100	ALTO
2	Registro de Cuentas	5	95	ALTO
3	Generar Código *QR*	3	90	MEDIO
4	Registro de Vehículos	5	85	ALTO
5	Visualizar Estacionamientos	5	80	ALTO
6	Pronóstico de Clientes	13	75	ALTO
7	Reportes de E/S Vehicular	2	70	MEDIO
8	Registro Automático de E/S Vehicular	8	65	ALTO
9	Registro Manual de E/S Vehicular	8	60	ALTO

Tabla 20. Backlog Técnico del Producto (Product Backlog Técnico)

N°	***Historia Técnica***	***Estimación***	***Prioridad de Negocio***	***Riesgo en Desarrollo***
1	Interoperabilidad. Rendimiento - Detección Vehicular	8	100	ALTO
2	Operabilidad. Escalabilidad - Despliegue en Producción	5	95	ALTO

4.3.3.1.7. Estimación

En cuanto a la valoración, se empleó la métrica de puntos de historia, apoyándose en la experiencia del grupo de desarrollo y la técnica de "*planning poker* con *fibonacci*", especificando la numeración especial por cada elemento que compone el *product backlog*, permitiendo determinar la complejidad de la historia.

En base a lo mencionado por Cohn (2005), este proceso considera la utilización de una baraja de cartas, en donde, cada miembro encargado del desarrollo elige un número que representa la complejidad de la historia conforme a su experiencia, si el equipo concuerda, se asignará el correspondiente valor, de lo contrario, se procederá a realizar un acuerdo (p. 56). En la tabla 21 se observa la planificación del trabajo.

Tabla 21. Calendario de Trabajo

Calendario de Trabajo			
Horas	***Días***	***Semanas***	***Meses***
4	5	4	1

4.3.3.1.8. Velocidad de Desarrollo

De acuerdo a los prerrequisitos del dueño del producto (*product owner)*, se consideran las valoraciones asignadas a las historias de usuario. En el *sprint* inicial, la historia uno tiene una estimación de 3 puntos, la historia dos se valora en 5 puntos, la historia tres en 3 puntos, la historia cuatro en 5 puntos y la historia cinco en 5 puntos, lo que suma en total 21 puntos de historia. Esto se traduce en un equivalente de 20 días de trabajo para el *sprint* 1.

4.3.3.1.9. Escenarios de Prueba

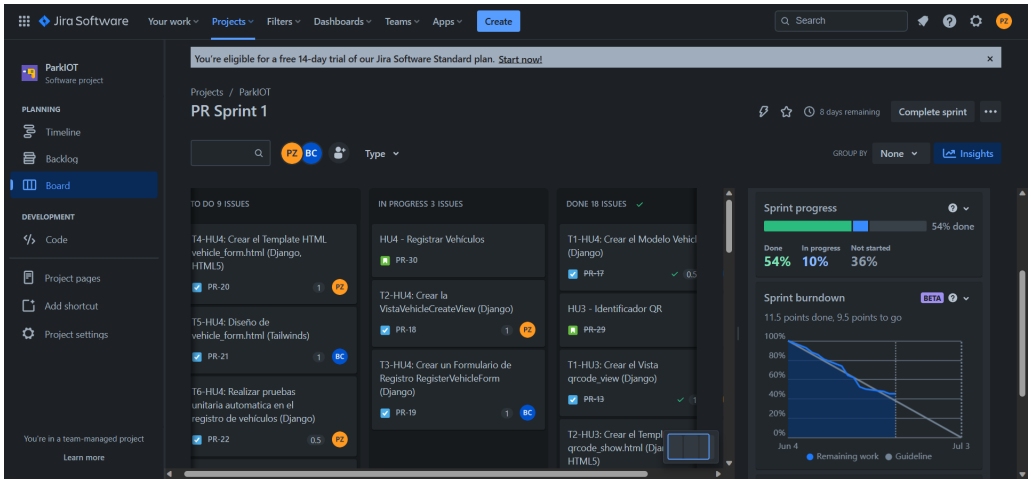
Una vez que se finalizan las historias de usuario, se comprueba que el sistema cumplió con las funcionalidades requeridas, necesitando los escenarios de prueba. Por lo tanto, Cohn (2004) afirma que, las pruebas de aceptación evidencian el cumplimiento de las historias de usuario, mismas que integran la metodología y diseño de *software* facilitando la interacción entre los desarrolladores y el usuario (p. 12).

4.3.3.1.10. Gestión de Incidencias

La gestión de incidencias y el registro de los avances de cada tarea, correspondientes a cada historia de usuario, se evidencia en el registro de tareas realizadas en la herramienta colaborativa *Jira*, esta herramienta permite gestionar y organizar las tareas dentro de un tablero *kanban*, como se visualiza en la figura 7.

La estructura de la herramienta consta de la sección “*to do*”, donde se establecen todas las tareas e historias que se van a realizar. La siguiente es la sección “*progress*”, ahí se define todas las que se están realizando en ese momento por los desarrolladores. La última sección es “*done*”, en esta zona se enlistan todas las tareas que ya han sido culminadas, a su vez el progreso se ve reflejado en el gráfico de trabajo pendiente.

Figura 7. Gestión de Incidencias Jira (Atlassian, 2023)



4.3.3.1.11. Sprint Backlog

Para el primer *sprint* se seleccionaron las 5 primeras historias de usuarios con mayor prioridad, de las cuales se acumularon un total de 21 puntos de estimación, como se visualiza en la tabla 22. Según Schwaber y Sutherland (2020), la utilidad del *sprint backlog* es estructurar las tareas, que los desarrolladores deben realizar, además de definir a los responsables y la estimación de las tareas sobre la estimación general de la historia de usuario (p. 13).

Tabla 22. Sprint Backlog 1

Sprint Backlog							
Objetivo: Elaborar el producto mínimo viable con las funciones de inicio de sesión *login*, registro de clientes, generar código *QR*, registro de vehículos y visualización de vehículos.							
Sprint	***Historia***	***Est***	***Categoría***	***Tarea***	***Est***	***Responsable***	***Estado***
1	HU1 - Inicio de Sesión (*Login*)	3	Desarrollo	Crear la URL a *LoginViewSet* (*Django*).	0.5	Patricio Cruz	Completo
			Desarrollo	Crear la página *AuthLogin.jsx* (*React, HTML5, JavaScript*)	1	David Carrascal	Completo
			Diseño	Estilizar el *AuthLogin.jsx* (*MUI, CSS*)	1	David Carrascal	Completo
			Pruebas	Realizar prueba unitaria automática al *Login* (*Django*)	0.5	Patricio Cruz	Completo
	HU2 - Registro de Cuentas	5	Desarrollo	Crear el Modelo *User* (*Django*)	1	Patricio Cruz	Completo
			Desarrollo	Crear la Vista *RegisterViewSet* (*Django*).	1	Patricio Cruz	Completo

		Desarrollo	Crear un Serializador de Registro *RegisterSerializer* (*Django*).	0.5	David Carrascal	Completo
		Desarrollo	Crear la página de Registro *AuthRegister.jsx* (*React, HTML5, JavaScript*)	1	Patricio Cruz	Completo
		Diseño	Estilizar el AuthRegister.jsx (*MUI, CSS*)	1	David Carrascal	Completo
		Pruebas	Realizar pruebas unitarias automática en el Registro de Clientes (*Django*)	0.5	David Carrascal	Completo
HU3 - Identificador QR	3	Desarrollo	Crear el Vista *UserQrcodeView* y el serializador *UserQrcodeSerializer*. (*Django*)	1	Patricio Cruz	Completo
		Desarrollo	Crear la página *QrcodeShow.jsx*. (*React, HTML5, JavaScript*)	1	David Carrascal	Completo
		Diseño	Estilizar el *QrcodeShow.jsx*. (*MUI, CSS*)	0.5	David Carrascal	Completo
		Desarrollo	Crear el Script *create_qrcode.py* para generar código QR. (*Django*)	0.5	Patricio Cruz	Completo
HU4 - Registrar Vehículos	5	Desarrollo	Crear el Modelo *Vehicle* (*Django*)	0.5	David Carrascal	En Proceso
		Desarrollo	Crear la Vista *VehicleViewList* y *VehicleViewDetail*. (*Django*)	1	Patricio Cruz	En Proceso
		Desarrollo	Crear un Serializador de Registro *VehicleSerializer* (*Django*)	1	David Carrascal	En Proceso
		Desarrollo	Crear la página *VehicleShow.jsx* y *VehicleCreate.jsx*. (*React, HTML5, JavaScript*)	1	Patricio Cruz	En Proceso
		Diseño	Estilizar *VehicleShow.jsx* y *VehicleCreate.jsx*. (*MUI, CSS*)	1	David Carrascal	En Proceso
		Pruebas	Realizar pruebas unitarias automática en el Registro de Vehículos (*Django*)	0.5	Patricio Cruz	En Proceso
HU5 - Visualizar	5	Desarrollo	Crear el Modelo *Space* y serializador *SpaceSerializer*. (*Django*)	0.5	David Carrascal	En Proceso

Estacionamientos	Desarrollo	Crear las Vistas *SpaceViewList* y *SpaceViewDetail*. (*Django*)	1	Patricio Cruz	En Proceso
	Desarrollo	Crear la página *ParkingShow.jsx* y componente *ParkingCard.jsx*. (*React, HTML5, JavaScript*)	1	David Carrascal	En Proceso
	Desarrollo	Estilizar el *ParkingShow.jsx* y *ParkingCard.jsx*. (*MUI, CSS*)	2	Patricio Cruz	En Proceso
	Pruebas	Realizar las pruebas sobre la visualización de estacionamientos.	0.5	David Carrascal	En Proceso

4.3.3.2. Sprint 1 - Reuniones diarias (*Daily Scrum*)

Por medio de la planeación realizada en el *sprint* 1, se llevaron a cabo reuniones diarias, cuya duración máxima fue de 15 minutos por día, en el que se registraron las actividades realizadas, los obstáculos encontrados para la culminación de las tareas y la definición de próximas tareas. Además, se empleó la aplicación *Jira* para dar seguimiento de las distintas tareas de ingeniería que están asociadas a las correspondientes historias de usuario, incluyendo sus estimaciones.

4.3.3.2.1. Historia de Usuario 1: Inicio de Sesión (Login)

La información detallada en la historia se puede hallar en el anexo V. Para el desarrollo de la historia 1, se utilizó un *template* gratuito de *MUI* con licencia del *MIT*, esta plantilla únicamente incorpora la funcionalidad de inicio de sesión, por lo cual se creó las páginas *AuthLogin.jsx* y *Login3.jsx,* para que se conecte con el *backend* y utilice el correo electrónico como parte de las credenciales para iniciar sesión, como se aprecia en la figura 8 vista desde la computadora y la figura 9 vista desde el teléfono.

Figura 8. Interfaz de Login3.jsx, vista de escritorio

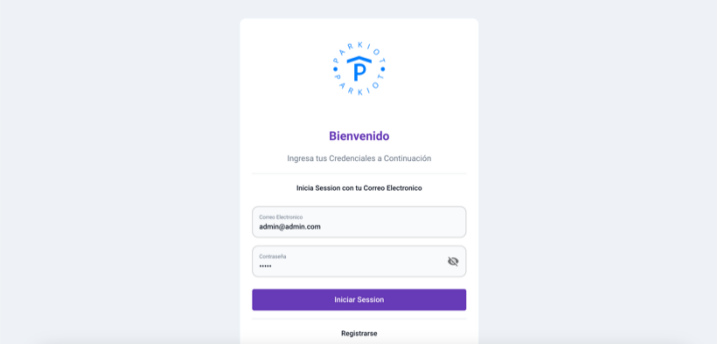
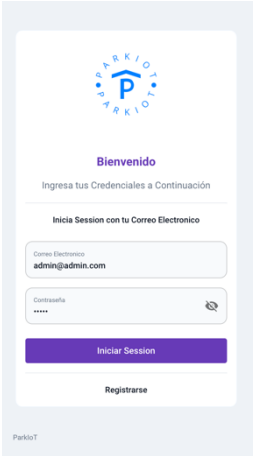


Figura 9. Interfaz de Login3.jsx, vista móvil



La pantalla para iniciar sesión fue realizada utilizando componentes de *MUI* y la biblioteca *React*, como se visualiza en la figura 10**.** Además, se empleó la librería de *Formik* para manipular los formularios, validarlos y manejar los mensajes de error, y para conectar con el *backend* se utilizó la librería de *Axios*, estas aplicaciones se indican en la figura 11.

Figura 10. Código de Login3.jsx

```
import { Link } from 'react-router-dom';

// material-ui
import { useTheme } from '@mui/material/styles';
import { Divider, Grid, Stack, Typography, useMediaQuery } from '@mui/material';

// project imports
import AuthWrapper1 from '../AuthWrapper1';
import AuthCardWrapper from '../AuthCardWrapper';
import AuthLogin from '../auth-forms/AuthLogin';
import Logo from 'ui-component/Logo';
import AuthFooter from 'ui-component/cards/AuthFooter';

// assets

// ================================|| AUTH3 - LOGIN ||================================ //

const Login = () => {
  const theme = useTheme();
  const matchDownSM = useMediaQuery(theme.breakpoints.down('md'));

  return (
    <AuthWrapper1>
      <Grid container direction="column" justifyContent="flex-end" sx={{ minHeight: '100vh' }}>
        <Grid item xs={12}>
          <Grid container justifyContent="center" alignItems="center" sx={{ minHeight: 'calc(100vh - 68px)' }}>
            <Grid item sx={{ m: { xs: 1, sm: 3 }, mb: 0 }}>
              <AuthCardWrapper>
                <Grid container spacing={2} alignItems="center" justifyContent="center">
                  <Grid item sx={{ mb: 3 }}>
                    <Link to="#">
                      <Logo width={150}/>
                    </Link>
```

Figura 11. Código de AuthLogin.jsx

En el *backend* se utilizó *Django*, con un servidor previamente configurado para que utilice autenticación con *JSON Web Tokens*, este servidor tiene licencia *MIT* para el uso libre, a partir de esa base se creó la vista *LoginViewSet,* que se muestra en la figura 12 y se modificó el modelo *User*, para que se adapte a las necesidades de la aplicación, como se muestra en la figura 13**.**

Figura 12. Código de la Vista LoginViewSet

Figura 13. Código del Modelo User

4.3.3.2.2. Historia de Usuario 2: Registro de Clientes

La información detallada en la historia se puede hallar en el anexo V. Para el desarrollo de la historia 2, se siguió utilizando el modelo para el registro de usuarios que tiene el *template* de *MUI*, pero se adaptó a las necesidades de la aplicación, creando así las páginas *AuthRegister.jsx* y *Register3.jsx*, para que se conecten al *backend* y realicen el proceso de registro con las funciones creadas, a la página de registro se le agregó los campos de nombres y cédula como se muestra en la figura 14, vista desde la computadora y la figura 15 vista desde el teléfono.

Figura 14. Interfaz de Register3.jsx, vista de escritorio

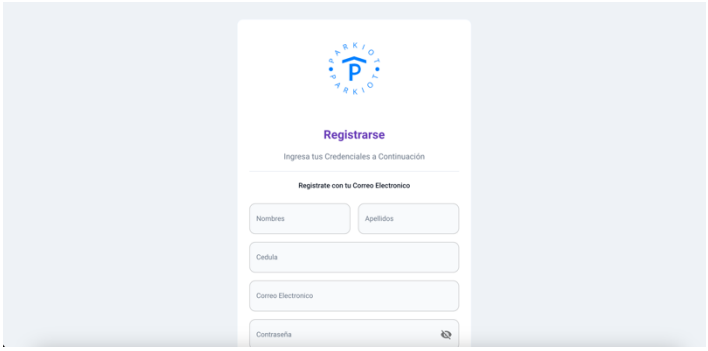


Figura 15. Interfaz de Register3.jsx, vista móvil

Se utilizó librerías de *React* y componentes de *MUI* para crear la página *Register3.jsx,* como se muestra en la figura 16, dentro de esta página se contiene *AuthRegister.jsx* que implementa la librería de *Formik* para manejar formulario, validarlos y manejar errores, además, *Axios* sirve para conectar con el *backend*, como se visualiza en la figura 17.

Figura 16. Código de Register3.jsx

```jsx
import { Link } from 'react-router-dom';

// material-ui
import { useTheme } from '@mui/material/styles';
import { Divider, Grid, Stack, Typography, useMediaQuery } from '@mui/material';

// project imports
import AuthWrapper1 from '../AuthWrapper1';
import AuthCardWrapper from '../AuthCardWrapper';
import Logo from 'ui-component/Logo';
import AuthRegister from '../auth-forms/AuthRegister';
import AuthFooter from 'ui-component/cards/AuthFooter';

// assets

// ================================|| AUTH3 - REGISTER ||================================ //

const Register = () => {
  const theme = useTheme();
  const matchDownSM = useMediaQuery(theme.breakpoints.down('md'));

  return (
    <AuthWrapper1>
      <Grid container direction="column" justifyContent="flex-end" sx={{ minHeight: '100vh' }}>
        <Grid item xs={12}>
          <Grid container justifyContent="center" alignItems="center" sx={{ minHeight: 'calc(100vh - 68px)' }}>
            <Grid item sx={{ m: { xs: 1, sm: 3 }, mb: 0 }}>
              <AuthCardWrapper>
                <Grid container spacing={2} alignItems="center" justifyContent="center">
                  <Grid item sx={{ mb: 3 }}>
                    <Link to="#">
                      <Logo width={150}/>
                    </Link>
```

Figura 17. Código de AuthRegister.jsx

```jsx
    </Grid>

    <Formik
      initialValues={{
        first_name: '',
        last_name: '',
        email: '',
        password: '',
        submit: null
      }}
      validationSchema={Yup.object().shape({
        email: Yup.string().email('Must be a valid email').max(255).required('El correo es requerido'),
        password: Yup.string().max(255).required('La contrasena es requerida')
      })}
      onSubmit={async (values, { setErrors, setStatus, setSubmitting }) => {
        if (checked) {
          try {
            axios
              .post(`${configData.API_SERVER}/api/users/register`, {
                first_name: values.first_name,
                last_name: values.last_name,
                email: values.email,
                password: values.password,
                username: extraerUsuarioCorreo(values.email)
              })
              .then((response) => {
                if (response.data.success) {
                  if (scriptedRef.current) {
                    setStatus({ success: true });
                    setSubmitting(false);
                  }
                  navigate('/login');
                } else {
```

En el *backend* se creó la vista *RegisterViewSet*, que mediante la función *create*, permite crear instancias del objeto *User,* que a su vez se guarda en la base de datos *MySQL* mediante el *ORM* de *Django*, la lógica se puede apreciar en el código de la figura 18. Dentro del proceso de creación del usuario, se utilizó el serializador *RegisterSerializer* para validar los campos correspondientes al modelo *User* como se visualiza en la figura 19.

Figura 18. Código de la vista RegisterViewSet

```python
from rest_framework import viewsets, status
from rest_framework.response import Response
from rest_framework.permissions import AllowAny

from api.authentication.serializers import RegisterSerializer

Patricio Cruz, 3 weeks ago | 1 author (Patricio Cruz)
class RegisterViewSet(viewsets.ModelViewSet):
    http_method_names = ["post"]
    permission_classes = (AllowAny,)
    serializer_class = RegisterSerializer

    def create(self, request, *args, **kwargs):
        serializer = self.get_serializer(data=request.data)     Patricio Cruz, 3 weeks ago • Actualizacion Completa

        serializer.is_valid(raise_exception=True)
        user = serializer.save()

        return Response(
            {
                "success": True,
                "userID": user.id,
                "msg": "The user was successfully registered",
            },
            status=status.HTTP_201_CREATED,
        )
```

Figura 19. Código del Serializador RegisterSerializer

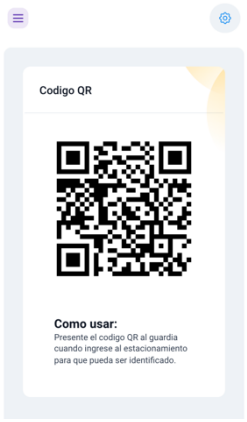
```python
from rest_framework import serializers
from rest_framework.exceptions import ValidationError
from django.core.exceptions import ObjectDoesNotExist
from api.user.models import User


class RegisterSerializer(serializers.ModelSerializer):
    password = serializers.CharField(min_length=4, max_length=128, write_only=True)
    username = serializers.CharField(max_length=255, required=True)
    email = serializers.EmailField(required=True)

    class Meta:
        model = User
        fields = ["id", "username", "password", "email", "is_active", "date", "first_name", "last_name"]

    def validate_username(self, value):
        try:
            User.objects.get(username=value)
        except ObjectDoesNotExist:
            return value
        raise ValidationError({"success": False, "msg": "Nombre de Usuario ya Existente."})

    def validate_email(self, value):
        try:
            User.objects.get(email=value)
        except ObjectDoesNotExist:
            return value
        raise ValidationError({"success": False, "msg": "Correo Electronico ya Existente"})

    def create(self, validated_data):
        return User.objects.create_user(**validated_data)
```

4.3.3.2.1. Historia de Usuario 3: Identificador QR

La información detallada en la historia se puede hallar en el anexo V. Para el desarrollo de la interfaz del Identificador *QR* se creó la página *QrcodeShow.jsx*, donde se decidió solo mostrar el código *QR* de manera que sea lo único visible para la interacción con el guardia al momento de escanear el código, debajo se definió una pequeña guía para que el usuario conozca la utilidad que tiene el código *QR*, como se visualiza en la figura 20 vista desde el computador y la figura 21 vista desde el teléfono.

Figura 20. Interfaz de QrcodeShow.jsx, vista de escritorio

Figura 21. Interfaz de QrcodeShow.jsx, vista móvil



Se utilizó las librerías de *React* y *MUI* para crear el componente *QrcodeCard.jsx* que presenta el código *QR*, ya que este código como imagen está comprimido en *base64* para que pueda ser almacenado, este es presentado con el componente *Box* que simula la etiqueta de imagen *HTML* convencional, como se muestra en la figura 22. Además, se creó el componente *QrcodeShow.jsx* que utiliza la librería de *Axios* para traer la información del *backend*, como se visualiza en la figura 23.

Figura 22. Código de QrcodeCard.jsx

```jsx
import React from 'react';
import { Box } from '@mui/material';
const QrcodeCard = ({ qrcode }) => {
  return (
    <>
      <Box
        component="img"
        sx={{
          height: {
            xs: 280,
            md: 400,
            xl: 450
          },
          width: {
            xs: 280,
            md: 400,
            xl: 450
          }
        }}
        src={`${qrcode}`}
        alt='Codigo QR con Informacion del Usuario'
      />
    </>
  );
};

export default QrcodeCard;
```

Figura 23. Código de QrcodeShow.jsx

```jsx
import { useEffect, useState } from 'react';
import { useSelector } from 'react-redux';
import axios from 'axios';
// import MainCard from 'ui-component/cards/MainCard';
import { Box, Typography } from '@mui/material';
// import { Typography } from '@mui/material'

import QrcodeCard from 'components/QrcodeCard';
import configData from '../../config';
import GeneralBack from 'components/GeneralBack';

const QrcodeShow = () => {
  const account = useSelector((state) => state.account);
  const [qrcode, setQrcode] = useState(null);

  const callApi = async () => {
    axios
      .get(`${configData.API_SERVER}/api/users/qrcode/${account.user?._id}/`, {
        headers: {
          Authorization: `${account.token}`
        }
      })
      .then((response) => {
        if (!response) {
          throw new Error({ msg: 'error' });
        }
        setQrcode(response.data.qrcode);
      })
      .catch((error) => {
        console.log(error);
        setQrcode(null);
```

En el *backend* se creó un *viewset UserQrcodeView* que consulta todos los usuarios en búsqueda del código *QR* correspondiente, esto se consiguió con el serializador por modelos *UserQrcodeSerializer,* que responde a todas las consultas con el identificador del usuario y su código *QR* en formato *base64*, como se visualiza en la figura 24. Además, se creó un *script* que utiliza la librería *qrcode* de *Python*, donde se genera un identificador único para el usuario utilizando como referencia el espacio de nombres *DNS,* que utiliza el

servidor y este *UUID* es embebido en una ruta *URL, y* a su vez es convertida en el código *QR* y pasado a formato *base64*, el código del *script* se aprecia en la figura 25.

Figura 24. Código del Serializador USerQrcodeSerializer

```
    vehicles = VehicleSerializer(many = True)
    Patricio Cruz, 3 weeks ago | 1 author (Patricio Cruz)
    class Meta:
        model = User
        fields = ["id", "username", "email", "date", "first_name", "last_name", "vehicles"]
        read_only_field = ["id"]

Patricio Cruz, 3 weeks ago | 1 author (Patricio Cruz)
class UserQrcodeSerializer(serializers.ModelSerializer):
    Patricio Cruz, 3 weeks ago | 1 author (Patricio Cruz)
    class Meta:
        model = User
        fields = ["id", "qrcode"]
        read_only_field = ["id", "qrcode"]
```

Figura 25. Código del Script create_qrcode.py

```
from uuid import uuid3, NAMESPACE_DNS
from io import BytesIO
import qrcode
from base64 import b64encode


def create_qrcode(username: str = "", url: str = "") -> list:
    try:
        # Creat Unique Identifier by the username and hash MD5
        uuid = uuid3(NAMESPACE_DNS, username)
        # Create QRCODE for the user and encode to base64
        qr = qrcode.QRCode(
            version=1,
            error_correction=qrcode.constants.ERROR_CORRECT_L,
            box_size=10,
            border=4,
        )
        qr.add_data(f'{url}/check/{uuid.hex}')
        qr.make(fit=True)
        img = qr.make_image(fill_color="black", back_color="white")

        buffered = BytesIO()
        img.save(buffered)
        img_str = b64encode(buffered.getvalue()).decode("utf-8")

        return [img_str, uuid]
    except Exception:
        print(Exception.args)
        return ["", ""]
```

4.3.3.2.2. Historia de Usuario 4: Registrar Vehículos

La información detallada en la historia se puede hallar en el anexo V. En el desarrollo de esta interfaz se utilizó la librería de *React* y componentes de *MUI* para crear la página *VehicleShow.jsx*, que se encarga de visualizar varias tarjetas que representa los vehículos con su información, como se visualiza en la figura 26 desde la computadora y la figura 27 desde el teléfono.

Figura 26. Interfaz de VehicleShow.jsx, vista de escritorio

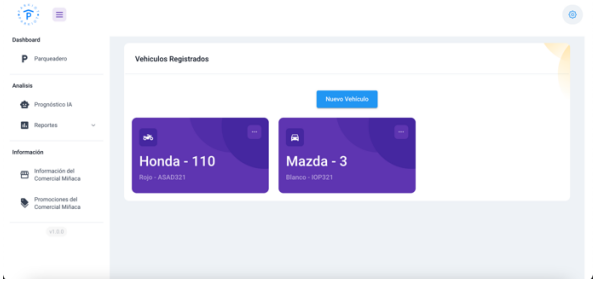
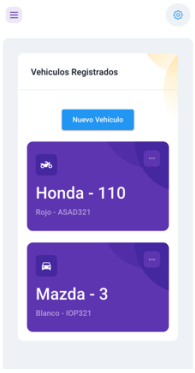


Figura 27. Interfaz de VehicleShow.jsx, vista móvil



En la misma sección del Registro de Vehículos, se creó la página *VehicleCreate.jsx,* que utiliza componentes de *MUI* y *Formik* para permitir crear un formulario e ingresar nuevos vehículos, como se muestra en la figura 28 desde la computadora y la figura 29 desde el teléfono.

Figura 28. Interfaz de VehicleCreate.jsx, vista de escritorio

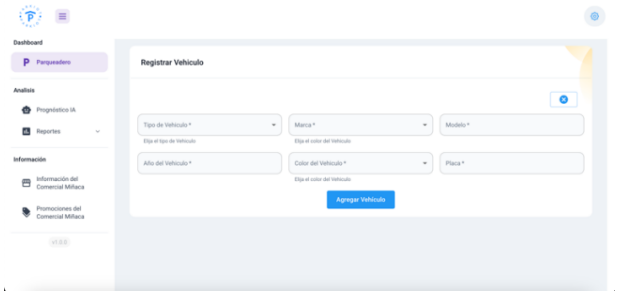
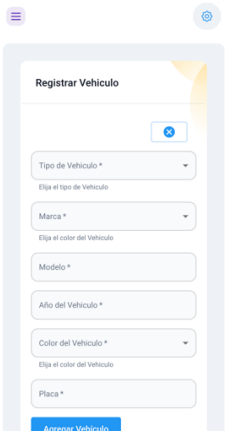


Figura 29. Interfaz de VehicleCreate.jsx, vista móvil



Para crear las páginas se utilizaron las librerías de *React* y *MUI*, primero se creó el componente *VehicleCard.jsx,* que define la información detallada de cada uno de los vehículos, esta tarjeta incluye la opción de actualizar los datos o eliminar el vehículo

vinculado al usuario, la estructura de la página se puede apreciar en el código de la figura 30, por lo tanto, se creó un componente base denominado *VehicleShow.jsx,* que muestra una tarjeta por vehículo relacionado al usuario, como se plasma en el código de la figura 31.

Figura 30. Código de VehicleCard.jsx

```
return (
  <>
    {isLoading ? (
      <SkeletonEarningCard />
    ) : (
      <CardWrapper border={false} content={false} ischecked={vehicleId === vehicle.id ? 'true' : 'false'}>
        <Box sx={{ p: 2.25 }}>
          <Grid container direction="column">
            <Grid item>
              <Grid container justifyContent="space-between">
                <Grid item>
                  <Avatar
                    variant="rounded"
                    sx={{
                      ...theme.typography.commonAvatar,
                      ...theme.typography.largeAvatar,
                      backgroundColor: theme.palette.secondary[800],
                      mt: 1
                    }}
                    onClick={handleCheckVehicle}
                  >
                    {vehicle.tipo === 'carro' ? (
                      <DirectionsCar stroke={1.5} size="1.3rem" sx={{ color: 'white' }} />
                    ) : (
                      <TwoWheeler stroke={1.5} size="1.3rem" sx={{ color: 'white' }} />
                    )}
                  </Avatar>
                </Grid>
                {!isForCheck && (
                  <Grid item>
                    <Avatar
                      variant="rounded"
```

Figura 31. Código de VehicleShow.jsx

```
//MUI
import { Grid, Box } from '@mui/material';
import { Button } from '@mui/material';

import AnimateButton from 'ui-component/extended/AnimateButton';

import configData from '../../config';
import VehicleCard from 'components/VehicleCard';
import GeneralBack from 'components/GeneralBack';

const VehicleShow = () => {
  const account = useSelector((state) => state.account);
  const { vehicles, slots } = useSelector((state) => state.vehicles);
  const dispatcher = useDispatch();
  const [disableAddButton, setDisableAddButton] = useState(false);
  const navigate = useNavigate();

  const handleAddVehicle = () => {
    if (slots < 3) {
      navigate('/vehicle/create');
    }
  };

  useEffect(() => {
    axios
      .get(`${configData.API_SERVER}/parking/vehicle/`, {
        headers: {
          Authorization: `${account.token}`
        }
      })
      .then((response) => {
        dispatcher({
          type: SET_VEHICLES,
```

Se creó la página *VehicleCreate.jsx* con la librería *Formik* para crear los formularios de registro de vehículos, validar los datos y manejar los errores, además de la librería *Axios* para enviar los datos al *backend*, como se visualiza en la figura 32.

Figura 32. Código de VehicleCreate.jsx

En el lado del *backend* se definió el modelo *Vehicle* con todos los campos que debe poseer cada instancia del objeto que se cree, como se está utilizando el *ORM* de *Django*, automáticamente se genera la estructura de información en la base de datos correspondientes al modelo que se muestra en la figura 33**.**

Figura 33. Código del Modelo Vehicle

Para llevar a cabo operaciones de *CRUD* sobre el modelo se crearon dos vistas *VehicleViewList,* que permite consultar una lista de todos los vehículos y crear nuevos vehículos como se muestra en la figura 34 y *VehicleViewDetail,* que permite actualizar y eliminar un vehículo en específico, como se visualiza en la figura 35.

Figura 34. Código de la vista VehicleViewList

Figura 35. Código de la vista VehicleViewDetail

```
class VehicleViewDetail(APIView):
    permission_classes = [IsAuthenticated,]

    def get_object(self, pk: int):
        try:
            return Vehicle.objects.get(pk=pk, owner=self.request.user)
        except Vehicle.DoesNotExist:
            return Http404

    def put(self, request, pk, format=None):
        vehicle = self.get_object(pk)



    def delete(self, request, pk, format=None):
        vehicle = self.get_object(pk)
        # vehicle.delete()
        vehicle.owner = None
        vehicle.save(update_fields=['owner'])
        return Response(status=HTTP_204_NO_CONTENT)
```

Para que la información sea enviada al *frontend*, esta debe ser convertida del formato *Python Data Object* a *JSON*, para esto se creó el serializador *VehicleSerializare*, que define los campos que se enviaran en la consulta y una función personalizada para la generación de objetos de la clase *Vehicle*, esta lógica se puede apreciar en el código de la figura 36.

Figura 36. Código del serializador VehicleSerializer

```
class VehicleSerializer(serializers.ModelSerializer):
    placa = serializers.CharField(
        validators=[
        UniqueValidator(
            # queryset=Vehicle.objects.all(),
            queryset=Vehicle.objects.filter(owner = not None),
            message="El Vehiculo ya se encuentra Registrado"
        )]
    )

    class Meta:
        model = Vehicle
        fields = ['id', 'brand', 'model', 'color',
                  'tipo', 'placa', 'year']
        read_only_field = ["id"]

    def create(self, validate_data):
        try:
            v = Vehicle.objects.get(placa = validate_data['placa'], owner = None)
            v.owner = validate_data['owner']
            v.save(update_fields=['owner'])
            return v
        except Vehicle.DoesNotExist:
            return Vehicle.objects.create(**validate_data)
```

4.3.3.2.3. Historia de Usuario 5: Visualizar Estacionamientos

La información detallada en la historia se puede hallar en el anexo V. Para desarrollar la interfaz se utilizó la librería de *React* y componentes de *MUI*. Para presentar la información de cada estacionamiento se crearon tarjetas que representen cada espacio, la información es el número de estacionamiento, la ubicación, el color determina si está libre u ocupado y además tiene un símbolo que representa si el estacionamiento es para un automóvil, motocicleta, camioneta y furgoneta, como se visualiza en la figura 37 vista desde el computador y la figura 38 vista desde el teléfono.

Figura 37. Interfaz de ParkingShow.jsx, vista de escritorio

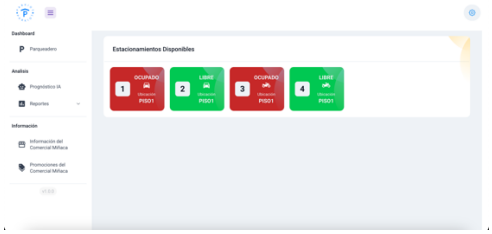


Figura 38. Interfaz de ParkingShow.jsx, vista móvil

Para la interfaz se creó el componente *ParkingCard.jsx,* con la librería de *React* y componentes de *MUI*, se creó una plantilla de tarjetas que representen cada estacionamiento como se muestra en la figura 39. Además, se utilizaron estas tarjetas para representar varias instancias de los estacionamientos en la página *ParkingShow.jsx*, dentro de esta página se utilizó el componente de *MUI Grid,* que permitió cuadrar los demás componentes en la página con un comportamiento responsivo. También, se utilizó la librería de *Axios* para traer la información de los estacionamientos desde el *backend* como se visualiza en la figura 40.

Figura 39. Código de ParkingCard.jsx

Figura 40. Código de ParkingShow.jsx

En el *backend,* se creó el modelo *Space* que define la estructura de información de cada estacionamiento, estos campos se convirtieron en columnas de la tabla *Space* que se crea automáticamente gracias al *ORM* de *Django*, el modelo se visualiza en la figura 41. Además, se creó un serializador por modelo denominado *SpaceSerializer*, este ayuda a convertir la información de los objetos en *Python data type* a *JSON* para ser enviado al *frontend*, como se observa en la figura 42.

Figura 41. Código del modelo Space

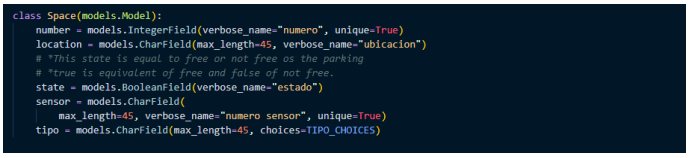


Figura 42. Codigo del serializador SpaceSerializer

Por último, se crearon las vistas *SpaceViewList* que permite obtener una lista de todos los registros de los espacios de estacionamiento y a su vez crearlos, y la vista *SpaceViewDetail* que permite actualizar el estado y la información de un estacionamiento en específico, estas vistas se pueden apreciar en la figura 43.

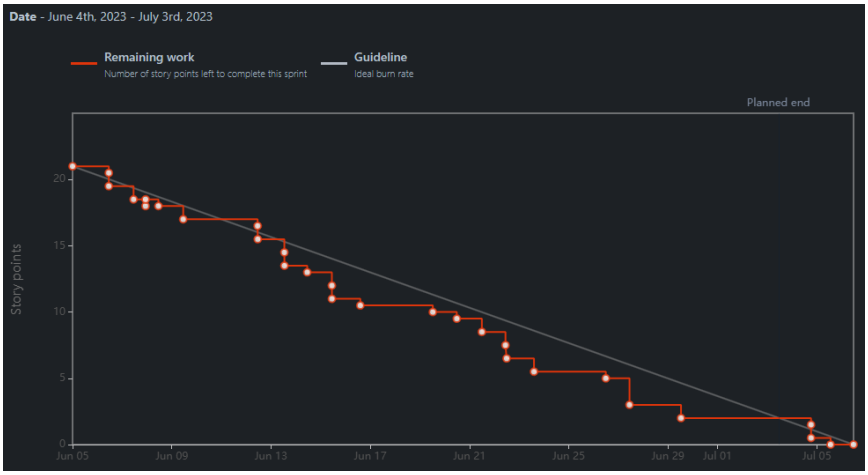
Figura 43. Código de las vistas SpaceViewList y SpaceViewDetail

```python
class SpaceViewList(mixins.ListModelMixin,
                    mixins.CreateModelMixin,
                    generics.GenericAPIView):
    queryset = Space.objects.all()
    serializer_class = SpaceSerializer
    permission_classes = [IsAuthenticated,]

    def get(self, request, *args, **kwargs):
        return self.list(request, *args, **kwargs)

    def post(self, request, *args, **kwargs):
        return self.create(request, *args, **kwargs)


class SpaceViewDetail(mixins.UpdateModelMixin,
                      mixins.RetrieveModelMixin,
                      generics.GenericAPIView):
    queryset = Space.objects.all()
    serializer_class = SpaceSerializer
    permission_classes = [AllowAny,]

    def get(self, request, *args, **kwargs):
        return self.retrieve(request, *args, **kwargs)

    def put(self, request, *args, **kwargs):
        self.partial_update(request, *args, **kwargs)
        return Response({"success": True}, status=HTTP_200_OK)
```

4.3.3.2.1. Sprint 1 - Gráfico de trabajo pendiente

Para el desarrollo del *sprint* 1, se utilizó el gráfico de trabajo pendiente como una guía para conocer las tareas por cumplir de las historias de usuario. Por lo tanto, se percibió la reducción de puntos de historia faltantes por completar en el sprint, como se puede visualizar en la figura 44, el gráfico de trabajo pendiente autogenerado por la herramienta de gestión de incidencias *Jira*.

Figura 44. Gráfico de trabajo pendiente del Sprint 1 generado por Jira



4.3.3.3. Sprint 1 - Revision (*Sprint Review*)

Culminado el desarrollo del primer *sprint*, se procedió a realizar la revisión entre los integrantes de desarrollo y el gerente del local comercial, se le indicaron los avances realizados sobre el sistema de estacionamiento y se verificó el progreso del desarrollo. La reunión duró alrededor de 2 horas, donde se revisaron las historias de usuarios con sus correspondientes escenarios de pruebas y las pruebas de aceptación que se encuentran en el anexo VII.

4.3.3.3.1. Pruebas unitarias (Unit Testing)

El propósito de las pruebas unitarias, es verificar el correcto funcionamiento de la estructura y lógica de la aplicación web. De esta forma se puede verificar, que se está obteniendo el comportamiento deseado y en caso de detectar problemas, se pueden realizar correcciones al código.

Durante la etapa de desarrollo de las historias de usuarios, se ejecutaron pruebas unitarias, con la técnica de caja blanca, por lo que se centra en el funcionamiento del código y estas pruebas son automatizadas. Para llevar a cabo estas pruebas unitarias se utilizó las funciones de *UnitTest* integradas en el *framework Django*, en la figura 45 se puede visualizar la evidencia de un caso de prueba unitaria de las funcionalidades de autenticación como: *login, register* y *logout*.

Figura 45. Prueba Unitaria del Login, Register y Logout

```
Patricio Cruz, last month | 1 author (Patricio Cruz)
from django.urls import reverse
from rest_framework.test import APITestCase
from rest_framework import status


Patricio Cruz, last month | 1 author (Patricio Cruz)
class AuthenticationTest(APITestCase):
    base_url_register = reverse("api:register-list")
    base_url_login = reverse("api:login-list")
    base_url_logout = reverse("api:logout-list")
    base_url_check_session = reverse("api:check-session-list")

    data_register = {"username": "test", "password": "pass", "email": "test@parkiot.com"}

    data_login = {"password": "12345678", "email": "teast@admin.com"}

    def test_register(self):
        response = self.client.post(
            f"{self.base_url_register}", data=self.data_register
        )
        self.assertEqual(response.status_code, status.HTTP_201_CREATED)
        response_data = response.json()
        self.assertEqual(response_data["success"], True)

    def test_login(self):
        response = self.client.post(f"{self.base_url_login}", data=self.data_login)
        self.assertEqual(response.status_code, status.HTTP_200_OK)
        response_data = response.json()
        self.assertEqual(response_data["success"], True)

    def test_logout(self):
        # Login to retrieve token
```

4.3.3.4. Sprint 1 - Retrospectiva (*Sprint Retrospective*)

Para obtener la retrospectiva del *sprint* 1, se formularon tres preguntas fundamentales con el fin de obtener una perspectiva clara, completa y concisa, como se puede apreciar en la tabla 23.

Tabla 23. Sprint 1 - Retrospectiva

Aspectos exitosos en el Sprint	***Mejoras que se podrían aplicar en el siguiente Sprint (recomendaciones para la mejora continua)***	***Errores en el Sprint***
Durante el desarrollo del *sprint* 1, se concluyó con éxito todas las tareas de ingeniería, completando así las operaciones de inicio de sesión (*login*), registro de clientes, generación de código *QR*, registro de vehículos y visualización de estacionamientos, correspondientes a cada historia de usuario. Además, se utilizó el *ORM* incorporado en *Django* para establecer conexión con la base de datos *MySQL* y almacenar la información de los modelos.	En la planificación del próximo *sprint*, se tomará en cuenta la documentación de *Django, React y Redux* como recursos de apoyo para abordar los problemas. Además, se definirán funciones individuales que ayudan a la detección de los problemas y corrección de los mismos de manera ágil.	Dentro del proceso de desarrollo del s*print* 1, se tuvo dificultades con el registro de vehículos. El error se detectó al momento de crear un nuevo vehículo, este proceso no vinculaba el vehículo con el usuario que tenía la sesión activa en el navegador *web*.

4.3.4. Sprint 2

4.3.4.1. Sprint 2 - Planificación (*Sprint Planning)*

En lo que respecta al *sprint planning* del *sprint* 2, se realizó un análisis de las historias de usuario 6 y 7, además de la incorporación de la historia técnica 1. El resultado global fueron 23 puntos de estimación, que son necesarios para completar el sprint, de igual forma se elaboró el *sprint backlog* que se visualiza en la tabla 24.

Tabla 24. Sprint Backlog 2

Sprint Backlog							
Objetivo: Elaborar el PMV con las funciones de detección vehicular, pronóstico de clientes y reportes de E/S vehicular							
Sprint	***Historia***	***Est***	***Categoría***	***Tarea***	***Est***	***Responsable***	***Estado***
2	HT1 - Interoperabilidad. Rendimiento - Detección Vehicular	8	Diseño	Diseñar las conexiones entre el sensor de proximidad y el microcontrolador. (*ESP8266, HC-SR04*)	1	David Carrascal	Completo
			Diseño	Diseñar la fuente de energía del sensor y el microcontrolador.	0,5	David Carrascal	Completo

			(*ESP8266, PowerBank*)			
		Desarrollo	*Falshear* el microcontrolador con *Micropython*. (*ESP8266, Micropython*)	1	Patricio Cruz	Completo
		Desarrollo	Programar el microcontrolador con el sensor y crear archivo *boot.py*. (*ESP8266, Micropython, HC-SR04*)	1	Patricio Cruz	Completo
		Desarrollo	Programar el microcontrolador como *publicador* crear archivo *main.py*. (*ESP8266, Micropython, MQTT*)	2	Patricio Cruz	Completo
		Desarrollo	Configurar un servidor con un *broker MQTT.* (*Ubuntu Server, Mosquitto, NodeRED*)	1	David Carrascal	Completo
		Desarrollo	Programar la aplicación *NodeRED* como *suscriptor.* (*NodeRED, MQTT*)	1	Patricio Cruz	Completo
		Pruebas	Realizar pruebas al funcionamiento del sensor. (*Django, ESP8266, MQTT*)	0,5	David Carrascal	Completo
HU6 - Pronóstico de Clientes	13	Análisis	Determinar las preguntas sobre los datos que se deben analizar.	2	David Carrascal	Completo
		Desarrollo	Recopilación de datos en la funcion *teach_model. (Python, Sklearn)*	1,5	David Carrascal	Completo
		Desarrollo	Limpieza de datos en la funcion *teach_mode*l. *(Python, Sklearn)*	1	Patricio Cruz	Completo
		Desarrollo	Manipulación de datos en la *funcion teach_model. (Python, Sklearn)*	1	Patricio Cruz	Completo
		Análisis	Determinar los datos que se van a incluir en el modelo de aprendizaje en la	1,5	David Carrascal	Completo

			funcion *teach_model*. (*Python, Sklearn*)			
		Análisis	Elegir el Algoritmo de Aprendizaje Automático en la funcion *teach_model*. (*Python, Sklearn*)	2	David Carrascal	Completo
		Desarrollo	Dividir los datos en conjunto de entrenamiento y prueba en la función *teach_model*. (*Python, Sklearn*)	1	Patricio Cruz	Completo
		Desarrollo	Entrenar y probar el modelo de aprendizaje automático en la funcion *teach_model*. (*Python, Sklearn*)	1	Patricio Cruz	Completo
		Pruebas	Puntuar el modelo de aprendizaje automatico en la función *teach_model*. (*Python, Sklearn*)	1	Patricio Cruz	Completo
		Diseño	Visualizar los datos generados por el modelo de aprendizaje automático en la función *prognosis_model. (Python, Sklearn*)	1	David Carrascal	Completo
HU7 - Reporte de E/S Vehicular	2	Desarrollo	Crear el Vista *RegisterTotalDayViewList* y *RegisterViewFiltered*. (*Django*)	0,5	Patricio Cruz	Completo
		Desarrollo	Crear la página *GeneralReportShow.jsx* y *ESRespotShow.jsx*. (*React, HTML5, JavaScript*)	0,5	Patricio Cruz	Completo
		Diseño	Estilizar el *GeneralReportShow.jsx* y *ESRespotShow.jsx*. (*MUI, CSS*)	0,5	David Carrascal	Completo

	Desarrollo	Crear un Serializador de *RegisterFilterSerializer*. (*Django*)	0,5	David Carrascal	Completo

4.3.4.2. Sprint 2 - Reuniones diarias (*Daily Scrum*)

Con la planificación del *sprint* 2, se detallaron los procedimientos necesarios que se debe cumplir, para la culminación deseada. Se aplicó la herramienta de gestión de incidencias *Jira,* para las reuniones diarias, de esta forma se gestionó el progreso del proyecto. Además, se mantuvo un seguimiento de las actividades de ingeniería, definidas para las historias de usuario e historia técnica, que fueron culminadas por cada desarrollador diariamente.

4.3.4.2.1. Historia técnica 1: Interoperabilidad. Rendimiento - Detección Vehicular

La información detallada en la historia técnica se puede hallar en el anexo VI. Abarca el desarrollo de una funcionalidad técnica y es consumida por una historia de usuario. En este caso, la interoperabilidad y el rendimiento necesario para la detección vehicular, es utilizada por la historia de usuario 5 “visualizar estacionamientos”.

En el desarrollo de los sensores aplicados para la detección vehicular, se utilizó los siguientes componentes: un sensor *HC-SR04*, un microcontrolador *NodeMCU ESP8266*, un regulador de energía de 12v, un cargador de 12v, 2 transistores que se encargan de controlar un *led* indicador de estado y un *buzzer* que indica de forma sonora la presencia de un vehículo.

Se diseñó la conexión entre el sensor de proximidad *HC-SR*04 y el microcontrolador *ESP8266*, en el que se utilizó un regulador de voltaje que alimenta con 3.3v al microcontrolador y 5v al sensor de proximidad, estas conexiones se pueden apreciar en la *protoboard* de la figura 46 y el diagrama de la figura 47. Además, estos componentes se ensamblaron en una tapa de *PBC* como se evidencia en la figura 48.

Figura 46. Protoboard con fuente de alimentación, sensor y microcontrolador

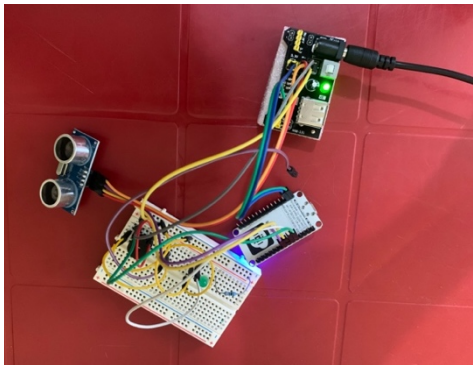


Figura 47. Diagrama Eléctrico del Sensor

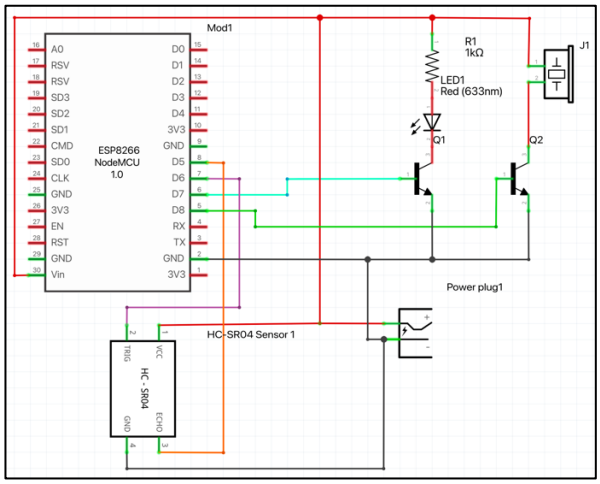
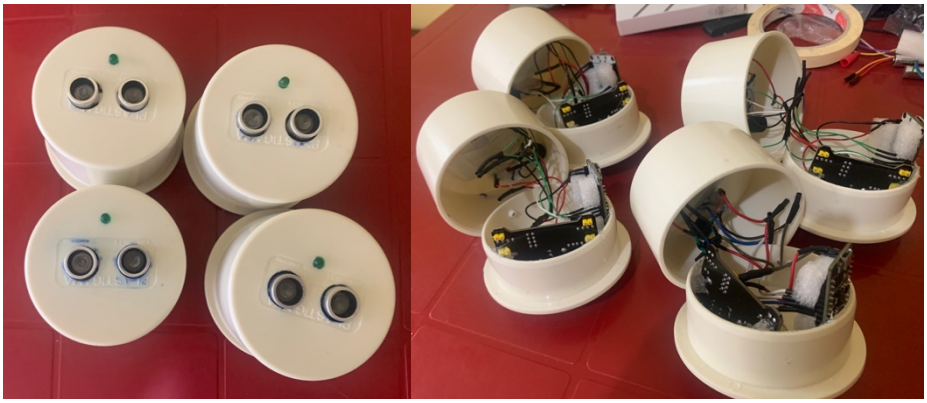
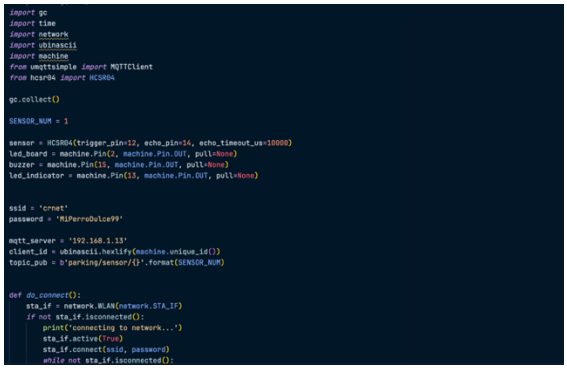


Figura 48. Sensor de proximidad y microcontrolador encapsulado en tapa PBC



Para el funcionamiento del microcontrolador con el sensor de proximidad, se *flasheo* el *ESP8266* con el *firemware* de *MicroPython*, de esta manera el microcontrolador puede ser programado con *Python*, por lo tanto, se procedió a configurar el archivo *boot.py,* que se encarga de *bootear* el microcontrolador, además de vincularse a la red inalámbrica *WiFi*, cómo se visualiza en el código de la figura 49.

Figura 49. Archivo de arranque boot.py



```
import gc
import time
import network
import ubinascii
import machine
from umqttsimple import MQTTClient
from hcsr04 import HCSR04

gc.collect()

SENSOR_NUM = 1

sensor = HCSR04(trigger_pin=12, echo_pin=14, echo_timeout_us=10000)
led_board = machine.Pin(2, machine.Pin.OUT, pull=None)
buzzer = machine.Pin(15, machine.Pin.OUT, pull=None)
led_indicator = machine.Pin(13, machine.Pin.OUT, pull=None)


ssid = 'crnet'
password = 'MiPerroDulce99'

mqtt_server = '192.168.1.13'
client_id = ubinascii.hexlify(machine.unique_id())
topic_pub = b'parking/sensor/{}'.format(SENSOR_NUM)


def do_connect():
    sta_if = network.WLAN(network.STA_IF)
    if not sta_if.isconnected():
        print('connecting to network...')
        sta_if.active(True)
        sta_if.connect(ssid, password)
        while not sta_if.isconnected():
```

Para la conexión del sensor, se utilizó la librería correspondiente al sensor *HC-SR04*, que permitió obtener los datos que lee el sensor, también se realizó la conexión al *broker MQTT* y se configuró el microcontrolador como un publicador. El proceso se puede apreciar en el código de la figura 50.

Figura 50. Archivo de ejecución main.py

```
last_message = 0
message_interval = 2

flag = True
MIN_DISTANCE = 200

while True:
    try:
        if (time.time() - last_message) > message_interval:
            distance = sensor.distance_cm()
            msg = '{'+'\"C_ID\":'+'\"' + \
                str(client_id)[2:10]+'\"'+',\"distance\":'+str(distance)+'}'
            client.publish(topic_pub, msg, qos=1)
            last_message = time.time()
        if distance < MIN_DISTANCE and flag:
            buzzer.on()
            led_indicator.on()
            time.sleep(1)
            buzzer.off()
            led_indicator.off()
            flag = False
        elif distance > MIN_DISTANCE:
            flag = True
        time.sleep_ms(3)
    except OSError as e:
        restart_and_reconnect()
```

Para obtener los datos generados por los sensores configurados como publicadores *MQTT*, estos se conectaron a un *broker MQTT*, por lo que se configuró un servidor con *Ubuntu Server,* que trabajó como *Fog Computing, el mismo se* visualiza en la figura 51. En el servidor se instaló el *broker* de código abierto *Mosquitto* y se utilizó la herramienta *NodeRED* como suscriptor del *bróker*, la aplicación se puede visualizar en el diagrama de funcionamiento de la figura 52. *NodeRED* se encargó de recabar los datos de los sensores, analizar la información con una función y enviarla mediante *HTTP REST* al servidor *backend Django* de la aplicación *web*.

Figura 51. Servidor en Ubuntu Server que trabaja como Fog Computing

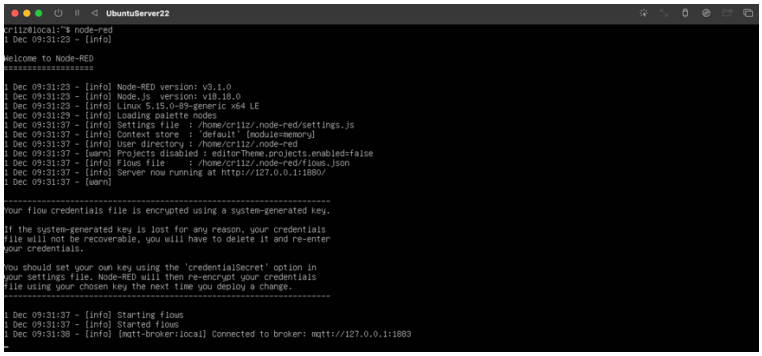
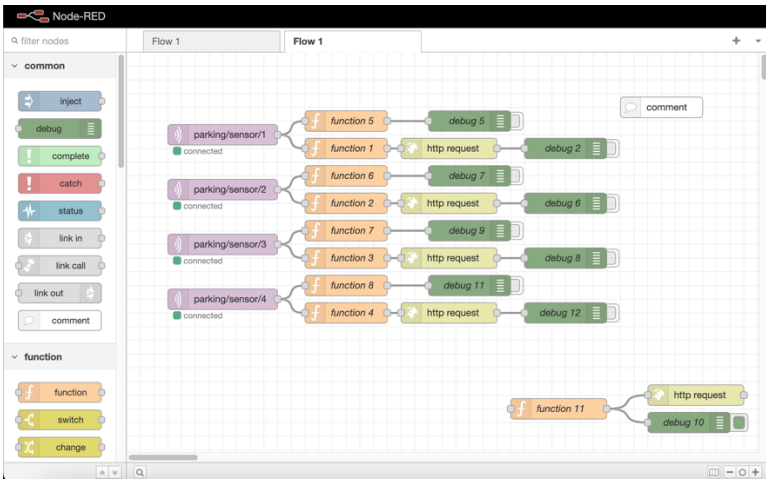


Figura 52. Diagrama Lógico de funciones de NodeRED como suscriptor del bróker



4.3.4.2.2. Historia de usuario 6: Pronóstico de clientes

La información detallada en la historia se puede hallar en el anexo V. El desarrollo de la interfaz gráfica empleó la biblioteca de *React* y los componentes de *MUI*. Se creó un botón para entrenar el modelo de *Machine Learning* y un texto de estado que indica si el modelo se encontraba entrenado o pendiente. Además, se definieron 2 campos de entrada de datos, uno correspondió a la fecha de pronóstico, con un límite de hasta 10 días y una lista de opciones que permitió seleccionar la franja horaria del pronóstico.

Una vez generado el pronóstico, la información se visualizó en un texto, como ALTA o BAJA, indicando la cantidad de vehículos prevista para ese día. Se puede observar la estructura de la interfaz en la figura 53 vista desde el computador, y la figura 54 vista desde el teléfono.

Figura 53. Interfaz de PrognosisShow.jsx, vista de escritorio

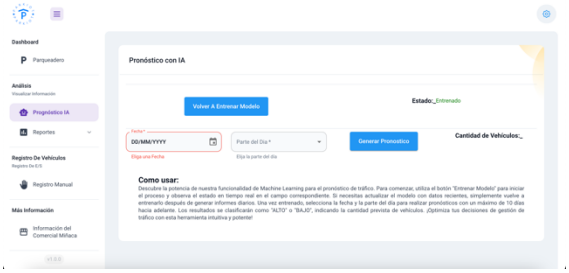


Figura 54. Interfaz de PrognosisShow.jsx, vista móvil

Para desarrollar los módulos de la interfaz gráfica, se creó la página *PrognosisShow.jsx*, donde se utilizaron los componentes de *MUI* para crear botones y campos de texto, además se utilizó los estados de *React*, para almacenar los valores de los campos. En la comunicación con el *backend*, se utilizó *Axios* para realizar peticiones *POST, que correspondió* al entrenamiento del modelo y se realizaron peticiones *GET*, para obtener la información del pronóstico. La estructura de la página se visualiza en el código de la figura 55.

Figura 55. Código de PrognosisShow.jsx

```
import BasicDatePicker from 'components/Mui/BasicDatepicker';

import AnimateButton from 'ui-component/extended/AnimateButton';

import GeneralBack from 'components/GeneralBack';
import MenuCard from 'components/MenuCard';

const POD = [
  { value: 'MAD', Label: 'MAD' },
  { value: 'MAN', Label: 'MAN' },
  { value: 'TAR', Label: 'TAR' },
  { value: 'NOC', Label: 'NOC' }
];

const PrognosisShow = () => {
  const [pickDate, setPickDate] = useState();
  console.log(pickDate);

  return (
    <>
      <GeneralBack title="Pronóstico con IA">
        <Grid container spacing={3}>
          <Grid item lg={6}>
            <AnimateButton>
              <Button fullWidth variant="contained" size="large">
                Entrenar Modelo
              </Button>
            </AnimateButton>
          </Grid>
          <Grid item lg={6}>
            <Box alignItems={['center']}>
              <Typography variant="h3">
```

En la fase de desarrollo del *backend*, se definieron tres modelos que ayudaron a generar información necesaria en el entrenamiento del modelo de *Machine Learning*. Estos

modelos fueron *Register* para el registro diario de entrada y salida de vehículos, *RegisterTotalDay* para la documentación de la información del día y *PromotionDay* para aportar información de las promociones disponibles en el local comercial. Estos modelos se pueden destacar en el código de la figura 56.

Figura 56. Código de los modelos Register, RegisterTotalDay y PromotionDay

```python
class Register(models.Model):
    time_entry = models.TimeField(auto_now_add=True)
    time_departure = models.TimeField(blank=True, null=True)
    date = models.DateField(auto_now_add=True)
    is_active = models.BooleanField(default=True)
    #TODO: Definir este campo para que acepte valores nulos
    user = models.ForeignKey(
        User, on_delete=models.CASCADE, verbose_name="usuario", related_name='users')
    vehicle = models.ForeignKey(
        Vehicle, on_delete=models.CASCADE, verbose_name="vehiculo", related_name='vehicles')
    #TODO: Si es necesrio, agregar la informacion del guardia que realizo el registro
    # guard = models.ForeignKey(User, on_delete=models.CASCADE)

class RegisterTotalDay(models.Model):
    date = models.DateField(auto_now_add=True)
    part_of_day = models.CharField(max_length=3)
    is_holiday = models.BooleanField(default=es_feriado(timezone.localtime(timezone.now()).date()))
    is_weekend = models.BooleanField(
        default=True if timezone.localtime(timezone.now()).weekday() >= 5 else False)

    is_promotionday = models.BooleanField(default=False)
    number_vehicles = models.IntegerField(verbose_name="cantidad de vehiculos")
    # temperature = models.IntegerField(default=0, verbose_name="temperatura de la ciudad")
    temperature = models.FloatField(default=0)

class PromotionDay(models.Model):
    name = models.CharField(max_length=45)
    date = models.DateField()
    description = models.CharField(max_length=100)

    def today_is_promotionday(self):
        return self.date == timezone.now().date()
```

Con la estructura de datos que se definió, se establecieron las bases con las que el modelo de *Machine Learning* se entrenó. Para llevar a cabo este proceso, se empleó la librería *sklearn* con el algoritmo de *Regresión Logística* y se utilizó *pandas* para crear el *dataframe* con el que se manipularon los datos.

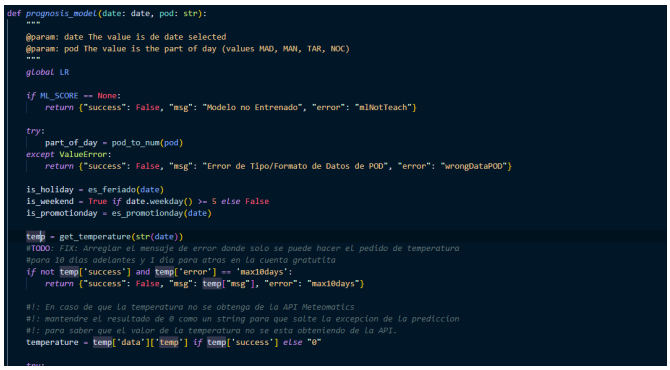
Para definir y entrenar el modelo de *Machine Learning,* se creó un *script* con la función *teach_model, d*onde se convirtió el *queryset* de *Django*, en un *dataframe*. Posteriormente, el *dataframe* se manipuló para que la información pudiera ser comprendida por la computadora, transformando los datos en números, dividiendo así la información en datos de entrada x, y datos de salida y, que a su vez se definieron en datos en entrenamiento y prueba. Con los datos, se entrenó el modelo con la función *fit* y se calificó la precisión del modelo con la función *score*, como se visualiza en el código de la figura 57.

Figura 57. Código de la función teach_model sel script ml.py

```python
def teach_model(queryset):
    global LR
    global ML_SCORE

    data = pd.DataFrame(list(queryset.values(
        'part_of_day', 'is_holiday', 'is_weekend', 'is_promotionday', 'number_vehicles', 'temperature')))

    data['is_holiday'] = label_encoder.fit_transform(data['is_holiday'])
    data['is_weekend'] = label_encoder.fit_transform(data['is_weekend'])
    data['is_promotionday'] = label_encoder.fit_transform(
        data['is_promotionday'])
    # data['part_of_day'] = label_encoder.fit_transform(data['part_of_day'])
    pod = []
    for i in data['part_of_day']:
        pod.append(pod_to_num(i))
    data['part_of_day'] = pod

    avg = data['number_vehicles'].mean()
    amount = []
    for i in data['number_vehicles']:
        # 0 es una cantidad baja y 1 es alta
        amount.append(1 if i >= avg else 0)
    data['amount'] = amount

    y = data['amount']
    data.drop(['number_vehicles', 'amount'], axis=1, inplace=True)
    X = data

    try:
        X_train, X_test, y_train, y_test = train_test_split(
            X, y, test_size=0.2, random_state=99)
    except ValueError:
```

Dentro del mismo archivo de *script* se creó la función *prognosis_model*, que se encargó de obtener los datos necesarios para efectuar un pronóstico. Los datos que se emplearon son la fecha y la franja horaria (*pod*). A partir de la fecha se extrajeron datos como la temperatura, si era feriado, fin de semana o día de promoción. Una vez obtenido los datos, se utilizó la función *predict* para enviar una matriz, con un solo registro que representa los datos de entrada para generar un solo dato de salida, resultando en la predicción. Esta lógica se puede visualizar en el código de la figura 58.

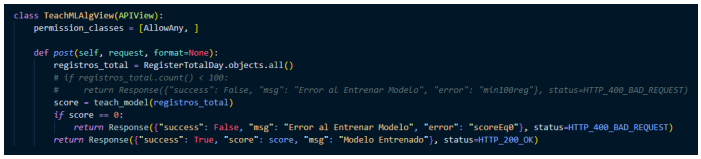
Figura 58. Código de la función prognosis_model del script ml.py



```python
def prognosis_model(date: date, pod: str):
    """
    @param: date The value is de date selected
    @param: pod The value is the part of day (values MAD, MAN, TAR, NOC)
    """
    global LR

    if ML_SCORE == None:
        return {"success": False, "msg": "Modelo no Entrenado", "error": "mlNotTeach"}

    try:
        part_of_day = pod_to_num(pod)
    except ValueError:
        return {"success": False, "msg": "Error de Tipo/Formato de Datos de POD", "error": "wrongDataPOD"}

    is_holiday = es_feriado(date)
    is_weekend = True if date.weekday() >= 5 else False
    is_promotionday = es_promotionday(date)

    temp = get_temperature(str(date))
    #TODO: FIX: Arreglar el mensaje de error donde solo se puede hacer el pedido de temperatura
    #para 10 dias adelantes y 1 dia para atras en la cuenta gratuita
    if not temp['success'] and temp['error'] == 'max10days':
        return {"success": False, "msg": temp["msg"], "error": "max10days"}

    #!: En caso de que la temperatura no se obtenga de la API Meteomatics
    #!: mantendre el resultado de 0 como un string para que salte la excepcion de la prediccion
    #!: para saber que el valor de la temperatura no se esta obteniendo de la API.
    temperature = temp['data']['temp'] if temp['success'] else "0"
```

Para poner a disposición las funciones de entrenamiento y pronóstico, se crearon dos vistas, que se llamaron mediante peticiones *HTTP REST*, la vista *TeachMLAlgView* responde a una solicitud *POST*, mediante el cual se interrogó a la base de datos, sobre los registros de *RegisterTotalDay*. Como resultado de la consulta, se obtuvo un *queryset* que se envió a la función *teach_model*, donde el puntaje fue enviado al *frontend* como confirmación de que se entrenó el modelo. La lógica de este procedimiento se visualiza en la figura 59.

Figura 59. Código de la vista TeachAlgView



La vista *PrognosisMLAlgView* respondía a una solicitud *GET*, donde se recibió como *query params* la fecha y la franja horaria *(pod),* se realizó una validación para asegurar que los datos recibidos sean correctos y luego se enviaron a la función *prognosis_model*. Si el pronóstico era exitoso, entonces se enviaba el resultado al *frontend*. Este proceso se encuentra representado en la figura 60.

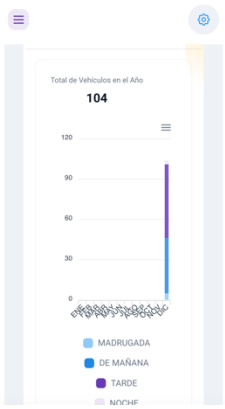
Figura 60. Código de la vista PrognosisMLAlgView

4.3.4.2.3. Historia de usuario 7: Reporte de E/S Vehicular

La información detallada en la historia se puede hallar en el anexo V. En el desarrollo de la interfaz, que representa la información de los reportes de la entrada y la salida de los vehículos, vista por el administrador. Se crearon dos páginas, una representa una gráfica y la otra una tabla. Para desarrollar la primera página, se utilizó un componente gráfico que incluye la plantilla de *MUI*, donde se representó la cantidad de vehículos por mes y por parte del día que se registraban, además de mostrar el total de vehículos en ese año. El grafico se visualiza en la figura 61 vista desde el computador y la figura 62 vista desde el teléfono

Figura 61. Interfaz de GeneralReportShow.jsx, vista de escritorio

Figura 62. Interfaz de GeneralReportShow.jsx, vista móvil



El desarrollo de la página *GeneralReportShow.jsx* fue realizada con un componente incorporado en la plantilla de *MUI*, que a su vez utilizó la librería *ApexChart* y *React-ApexChart* para generar gráficos dinámicos. Además, se utilizó la librería de *Axios* para realizar una petición *GET*, de esa manera se obtuvo del *backend,* los datos necesarios para reflejarse en el gráfico de barras, como se aplicó en el código de la figura 63.

Figura 63. Código de GeneralReportShow.jsx

```
const getDataAPI = async () => {
  await cax
    .get('/parking/registertotalday')
    .then((response) => {
      setRegisters(response.data);
    })
    .catch((error) => {
      console.log(error);
    });
};

useEffect(() => {
  getDataAPI();
}, []); // eslint-disable-line react-hooks/exhaustive-deps

useEffect(() => {
  if (registers) {
    setIsLoading(false);
  }
}, [registers]);

return (
  <>
    <GeneralBack title="Reporte General">
      <Box textAlign="center">
        <Grid container spacing={2} sx={{ alignItems: 'center', display: 'flex' }}>
          <Grid item xs={12} lg={12}>
            <TotalGrowthBarChart isLoading={isLoading} registers={registers} />
          </Grid>
        </Grid>
      </Box>
    </GeneralBack>
```

La otra página de reportes de entrada y salida, visualiza una tabla con información más detallada. Se implementó un campo de calendario y otro para ingresar la placa del vehículo, de esa forma se genera una consulta filtrada, como se visualiza en la figura 64 vista desde el computador y la figura 65 vista desde el teléfono.

Figura 64. Interfaz de ESReportShow.jsx, vista de escritorio

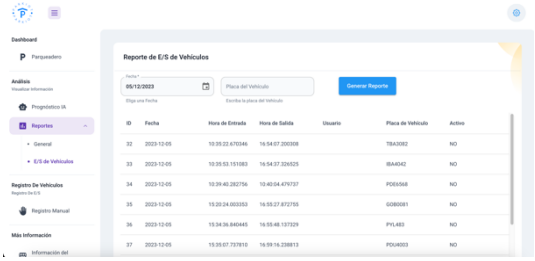
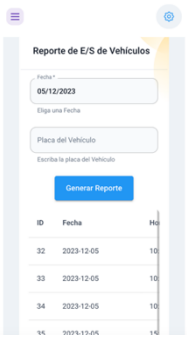


Figura 65. interfaz de ESReportShow.jsx, vista móvil



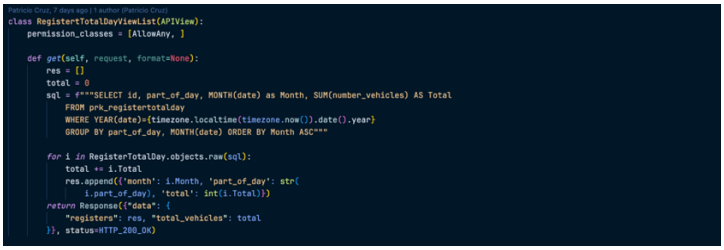
El desarrollo de la página *ESRespotShow.jsx,* se logró con la implementación de componentes de *MUI* para presentar la tabla y seleccionar la fecha del calendario. Además, se utilizó la librería de *Axios* para realizar una petición *GET* al *backend* y traer la información necesaria para mostrar en la tabla, la fecha y el número de placa se envíaron como parte de la ruta, siendo estos los *query params*. La lógica de este proceso se visualiza en el código de la figura 66.

Figura 66. Código de ESResportShow.jsx

En el *backend* se definieron dos vistas y un serializador que utilizaban los modelos de registro, previamente creados para el pronóstico con *Machine Learning*. Se creó la vista *RegisterTotalDayViewList* que responde a una solicitud *GET*, realizando una consulta *SQL*

a la base de datos y seleccionando la información requerida para el gráfico de barras, como se visualiza en el código de la figura 67.

Figura 67. Código de la vista RegisterTotalDayViewList



```
Patricio Cruz, 7 days ago | 1 author (Patricio Cruz)
class RegistertTotalDayViewList(APIView):
    permission_classes = [AllowAny, ]

    def get(self, request, format=None):
        res = []
        total = 0
        sql = f"""SELECT id, part_of_day, MONTH(date) as Month, SUM(number_vehicles) AS Total
            FROM prk_registertotalday
            WHERE YEAR(date)={timezone.localtime(timezone.now()).date().year}
            GROUP BY part_of_day, MONTH(date) ORDER BY Month ASC"""

        for i in RegisterTotalDay.objects.raw(sql):
            total += i.Total
            res.append({'month': i.Month, 'part_of_day': str(
                i.part_of_day), 'total': int(i.Total)})
        return Response({"data": {
            "registers": res, "total_vehicles": total
        }}, status=HTTP_200_OK)
```

Para generar la información de la tabla de reportes de entrada y salida, se creó la vista *RegisterViewFiltered* como se visualiza en la figura 68. La vista obtiene información como la fecha y el número de placa mediante *query params*, realizando la consulta correspondiente a la base de datos utilizando las funcionalidades del *ORM* de *Django*. Además, se utilizó el serializador RegisterFilterSerializer para convertir la consulta a la base de datos en formato *JSON,* que posteriormente se envía al *frontend*, el código del serializador se visualiza en la figura 69.

Figura 68. Código de RegisterViewFiltered

```
Patricio Cruz, 7 days ago | 1 author (Patricio Cruz)
class RegisterViewFiltered(APIView):
    permission_classes = [AllowAny, ]

    def get(self, request):
        try:
            # date = str(request.GET.get('year')) + '-' + \
            #     str(request.GET.get('month') + '-' + str(request.GET.get('day')))
            date = datetime.strptime(
                str(request.GET.get('year')) + '-' +
                str(request.GET.get('month') + '-' +
                    str(request.GET.get('day'))), "%Y-%m-%d"
            ).date()
        except ValueError:
            return Response(status=HTTP_400_BAD_REQUEST)
        except TypeError:
            return Response(status=HTTP_400_BAD_REQUEST)

        if request.GET.get('placa'):
            try:
                vehicle = Vehicle.objects.get(
                    placa=str(request.GET.get('placa')))
            except Vehicle.DoesNotExist:
                return Response(status=HTTP_400_BAD_REQUEST)
            registers = Register.objects.filter(date=date, vehicle=vehicle)
        else:
            registers = Register.objects.filter(date=date)

        r = RegisterFilteredSerializer(registers, many=True)      Patricio Cruz, 7 days ago • Avances
        if not r.data:
            return Response(status=HTTP_400_BAD_REQUEST)
        return Response(r.data, status=HTTP_200_OK)
```

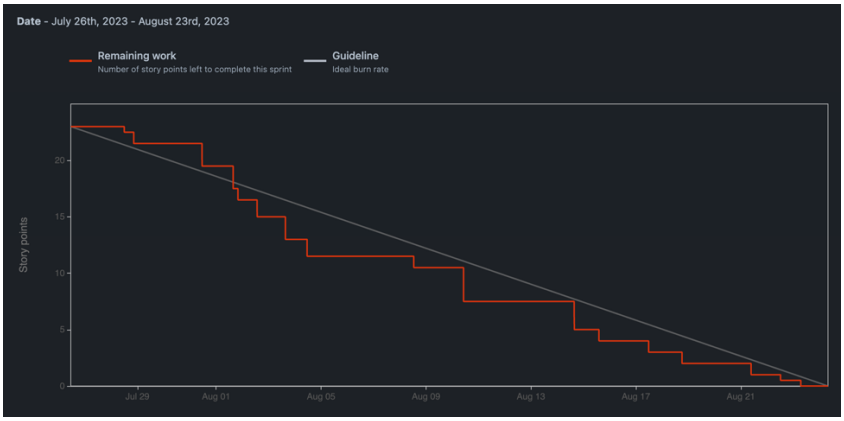
Figura 69. Código de RegisterFilterSerializer

```
Patricio Cruz, 7 days ago | 1 author (Patricio Cruz)
class RegisterFilteredSerializer(serializers.ModelSerializer):
    user = serializers.ReadOnlyField(source='user.email')
    vehicle = serializers.ReadOnlyField(source='vehicle.placa')

    Patricio Cruz, 7 days ago | 1 author (Patricio Cruz)
    class Meta:
        model = Register
        fields = '__all__'
```

4.3.4.2.1. Sprint 2 - Gráfico de trabajo pendiente

Las tareas de ingeniería faltantes, fueron evidenciadas en la figura 70, en el gráfico de trabajo pendiente, autogenerado por la herramienta de gestión de incidencias *Jira*, que representa si el desarrollo estuvo adelantado o atrasado,

Figura 70. Gráfico de trabajo pendiente del Sprint 2 generado por Jira



4.3.4.3. Sprint 2 - Revisión (Sprint Review)

El dueño del producto realizó la comprobación de los avances y mejoras en el sistema y la aplicación web. La reunión para verificar los avances duró alrededor de 2 horas aproximadamente y en ese tiempo se revisaron las historias de usuario con sus correspondientes escenarios de prueba, de igual forma se valoró las pruebas de aceptación que se pueden visualizar en el anexo VII.

4.3.4.4. Sprint 2 - Retrospectiva (Sprint Retrospective)

Se definieron tres preguntas esenciales para obtener la retrospectiva del *sprint* 2, estas preguntas fueron consultadas, una vez finalizada la revisión del *sprint*, los resultados se detallan en la tabla 25.

Tabla 25. Sprint 2 - Retrospectiva

Aspectos exitosos en el Sprint	***Mejoras que se podrían aplicar en el siguiente Sprint (recomendaciones para la mejora continua)***	***Errores en el Sprint***
Durante el desarrollo del *sprint* 2, se logró concluir con éxito todas las tareas de ingeniería, completando así las funcionalidades de detección vehicular, Pronóstico de Clientes y Reportes de E/S vehicular. Además, se terminaron de ensamblar los sensores para su correcto funcionamiento.	Una de las sugerencias para la planificación del próximo *sprint*, es que se analizará todo el ámbito de funcionalidades que se pueden incorporar en las historias de usuario para que no se necesiten modificar.	Dentro del proceso de desarrollo del *sprint* 2 se tuvo dificultades con el registro de la entrada y salida de vehículos, ya que no se contaba con una funcionalidad para realizar dicho registro.

4.3.5. Sprint 3

4.3.5.1. Sprint 3 - Planification (Sprint Planning)

En lo concerniente al s*print planning* del *sprint* 3, se realizó un análisis de las historias de usuario 8, 9 y la historia técnica 2. El resultado global fueron 21 puntos de estimación, que son necesarios para completar el *sprint*, de igual forma se elaboró el *sprint backlog* que se visualiza en la tabla 26.

Tabla 26. Sprint Backlog 3

Sprint Backlog							
Objetivo: Elaborar el PMV con las funciones de registro Automático y Manual de la E/S de Vehículo, además del despliegue en el cloud.							
Sprint	***Historia***	***Est***	***Categoría***	***Tarea***	***Est***	***Responsable***	***Estado***
3	HU8-Registro Automático de E/S Vehicular	8	Desarrollo	Crear la vista *CheckView.* (*Django*).	1	Patricio Cruz	Completo
			Desarrollo	Crear la vista *RegisterEntryView.* (*Django*)	2	Patricio Cruz	Completo
			Desarrollo	Crear la vista *RegisterDeparture View.* (*Django*)	1.5	Patricio Cruz	Completo
			Desarrollo	Crear el Serializaer *RegisterSerializaer*. (*Django*)	0.5	Patricio Cruz	Completo
			Desarrollo	Crear la página *CheckShow.jsx*.	1	David Carrascal	Completo

			(*React, HTML5, JavaScript*)			
		Diseño	Estilizar el *CheckShow.jsx.* (*MUI, CSS*)	2	David Carrascal	Completo
HU9-Registro Manual de E/S Vehicular	8	Desarrollo	Crear la vista *CheckManualView*. *(Django)*	2	Patricio Cruz	Completo
		Desarrollo	Modificar la vista *RegisterEntryView.* (*Django*)	2	Patricio Cruz	Completo
		Desarrollo	Crear la página *CheckManualShow .jsx*. (*React, HTML5, JavaScript*)	1	David Carrascal	Completo
		Diseño	Estilizar el *CheckManualShow .jsx.* (*MUI, CSS*)	2	David Carrascal	Completo
		Pruebas	Realizar pruebas unitarias el registro manual de E/S Vehicular (*Django*)	1	David Carrascal	Completo
HT2 - Despliegue en el Cloud	5	Implementación	Registrar el medio de pago	0.25	Patricio Cruz	Completo
		Implementación	Enlazar la cuenta de *DigitalOcean* con la cuenta de *Github*	0.25	David Carrascal	Completo
		Desarrollo	Modificación del archivo *package.json*	0.25	Patricio Cruz	Completo
		Implementación	Crear una aplicación *Node.js* con el repositorio de *frontend* (*GitHub*)	0.25	Patricio Cruz	Completo
		Implementación	Cambiar el script de *build* y *start* para que se ejecuten en producción	1	Patricio Cruz	Completo
		Implementación	Agregar variables de entorno a la aplicación de Node.js	0.25	David Carrascal	Completo
		Implementación	Crear una aplicación *Python* con el repositorio de *backend* (*GitHub*)	1	Patricio Cruz	Completo
		Implementación	Cambiar el script de *start* para que	0.25	Patricio Cruz	Completo

		se ejecute con el servidor *Gunicorn*			
	Implementación	Agregar variables de entorno a la aplicación de *Python*	0.25	David Carrascal	Completo
	Implementación	Crear una instancia de Base de Datos *MySQL*	0.5	Patricio Cruz	Completo
	Implementación	Crear una Base de Datos en la configuración de la instancia	0.25	David Carrascal	Completo
	Implementación	Crear un usuario para conectarse de forma segura	0.25	Patricio Cruz	Completo
	Implementación	Agregar la aplicación *Python* del *backend* y la base de datos al mismo grupo seguro de conexión.	0.25	David Carrascal	Completo

4.3.5.2. Sprint 3 - Reuniones diarias (Daily Scrum)

En el proceso de desarrollo del *sprint* 3, se ejecutaron reuniones diarias con una duración de 15 minutos, se utilizó la herramienta de gestión de incidencias Jira, para validar los avances diarios de cada desarrollador. Se mantuvo un seguimiento de la culminación de las tareas de ingeniería, establecidas previamente en las historias de usuario y la historia técnica.

4.3.5.2.1. Historia de Usuario 8: Registro Automático de E/S Vehicular

La información detallada en la historia se puede hallar en el anexo V. Para el diseño de la interfaz gráfica, se utilizó *React* y la librería de componentes de *MUI*. La interfaz gráfica definió información del cliente como el nombre, apellido, cédula, correo electrónico, una lista de los vehículos del cliente y dos botones de acción para registrar la entrada y la salida. Esta página se puede abrir en la computadora, como se aprecia en la figura 71, o en la figura 72 vista desde el teléfono móvil.

Figura 71. Interfaz CheckShow.jsx vista de escritorio

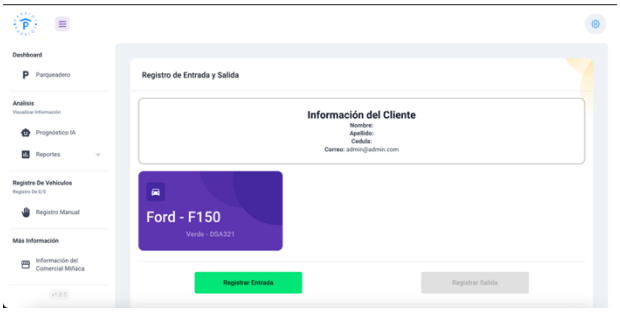
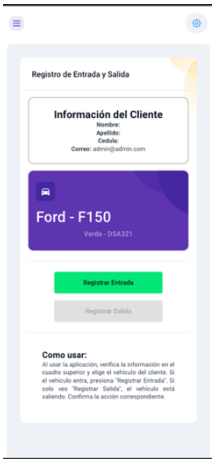


Figura 72. Interfaz CheckShow.jsx vista móvil



El desarrollo de la página *CheckShow.jsx* se logró mediante la aplicación de *React hooks*, la librería de componente *MUI* y la librería *Axios* para él envió de peticiones *HTTP* al *backend*. Se reutilizó el componente *VehicleCard.jsx* para mostrar la lista de vehículos que se pueden seleccionar, junto a las peticiones con *Axios,* se utilizó un componente de mensajes de error, que es visualizado dependiendo del estado de respuesta que se reciba del *backend*. El desarrollo se puede visualizar en el código de la figura 73.

Figura 73. Código del CheckShow.jsx

En el desarrollo del *backend,* se utilizó los modelos *Register* y *RegisterTotalDay* de la historia de pronóstico, para crear los registros de *E/S* de vehículos. Se elaboró primero la vista *CheckView*, que verificaba el estado de los registros para saber si un vehículo se encontraba o no en el estacionamiento, y se realizaban peticiones a la base de datos para verificar los estados de los registros, como se puede apreciar en el código de la figura 74.

Figura 74. Código de la vista CheckView

Para registrar la entrada del vehículo, se creó la vista *RegisterEntryView*, cuando un vehículo desea registrar su entrada, los datos enviados son: el usuario, la información del vehículo y la información del guardia. La información se valida y se crea un nuevo registro con la hora de entrada actual. En el proceso se verifica si existe un registro en la base de datos, que tenga el número de vehículos que haya entrado el día de hoy, si es así se suma un nuevo vehículo, caso contrario se crea un registro del modelo *RegisterTotalDa. La lógica de* este proceso se puede visualizar en el código de la figura 75.

Figura 75. Código de la vista RegisterEntryView

Para crear los registros y validar su información, se creó el serializador *RegisterSerializer,* que obtiene los datos del *frontend*, verifica que estén en el formato correcto y valida si existe o no un registro activo con ese vehículo ante de registrar una entrada, la funcionalidad se puede apreciar en la figura 76.

Figura 76. Código del serializador RegisterSerializer

Por último, para el registro de la salida se creó la vista *RegisterDepartureView*, que obtenía el registro actual con el código de registro, para establecer la hora de salida y el estado del registro como inactivo, de esta forma finaliza el ciclo de funcionamiento de la *E/S* de vehículos. El código de la funcionalidad de registro de salida se puede apreciar en la figura 77.

Figura 77. Código de la vista RegisterDepartureView

4.3.5.2.2. Historia de Usuario 9: Registro Manual de E/S Vehicular

La información detallada en la historia se puede hallar en el anexo V. Para el desarrollo de la interfaz gráfica se utilizó *React* y la librería de componentes de *MUI*. Se definió un cuadro de texto, para escribir el número de placa, esta placa se refleja en un texto mayor, para facilitar la lectura, además se agregó un botón que permite validar la placa con los registros actuales, verificando si se debe registrar la entrada o la salida. La interfaz está diseñada para que sea responsiva y se adapte al modo escritorio como se visualiza en la figura 78, o en el teléfono en la figura 79.

Figura 78. Interfaz CheckManualShow.jsx vista de escritorio

Figura 79. Interfaz CheckManualShow.jsx vista móvil

Para el desarrollo de la página *CheckManualShow.jsx,* se utilizó la librería de *React con* los componentes de *MUI,* para crea la estructura de la página. Se aplicó la librería de *Axios* para verificar si existen registros correspondientes a la placa ingresada y enviar peticiones al *backend*, generando un registro de entrada o salida. Esta lógica se aprecia en el código de la figura 80.

Figura 80. Código del CheckManualShow.jsx

```
import GeneralBack from 'components/GeneralBack';
import AnimateButton from 'ui-component/extended/AnimateButton';
import MessageCard from 'components/MessageCard.jsx';
import Spinner from 'components/Spinner.jsx';

const CheckManualShow = () => {
  const account = useSelector((state) => state.account);
  const dispatcher = useDispatch();
  const cax = caxios(account.token);

  const [open, setOpen] = useState(false);
  const [msg, setMsg] = useState(null);
  const [type, setType] = useState(null);

  const [placa, setPlaca] = useState('');
  const [register, setRegister] = useState(null);

  const [isLoaded, setIsLoaded] = useState(true);
  const [statusReq, setStatusReq] = useState(0);

  const handleCheckVehicle = async () => {
    if (placa) {
      setStatusReq(0);
      await cax
        .get(`/parking/check/manual/${placa}/`)
        .then((response) => {
          if (response.status === 200) {
            setRegister(response.data.register);
            setMsg('Vehiculo Existente Listo para Registrar');
            setType('success');
            setOpen(true);
          } else if (response.status === 204) {
```

En el *backend*, se creó la vista *CheckManualView*, la cual utiliza la placa del vehículo para verificar los registros de entrada/salida, si existe un vehículo dentro del estacionamiento. En el proceso, se verifica si el vehículo ya está registrado en el sistema. Si

encuentra un registro que tenga la placa del vehículo activa y se encuentre dentro del estacionamiento, retorna el registro para que solo se valide la salida. Esta lógica se visualiza en el código de la figura 81.

Figura 81. Código de la vista CheckManualView

```python
class CheckManualView(APIView):
    permission_classes = [IsAuthenticated, ]

    def get(self, request, placa, format=None):
        try:
            vehicle = Vehicle.objects.get(placa=placa)
        except Vehicle.DoesNotExist:
            return Response(status=HTTP_204_NO_CONTENT)
        register = Register.objects.filter(
            vehicle=vehicle, date=timezone.localtime(timezone.now()).date()).last()
        # Me valida si no existe un registro o si algun registro ya no esta activo
        # para no entregar informacion al serializador
        if register == None or not register.is_active:
            r_serilizer = None
        else:
            r_serilizer = RegisterSerializer(register)

        return Response({
            "register": r_serilizer.data if r_serilizer != None else None
        }, status=HTTP_200_OK)
```

Se adaptó la vista *RegisterEntryView*, para verificar si el proceso de registro es manual. Esto se logra al igualar a 0 el identificador del usuario y comprobar, cuando se obtenga un *ID* de usuario igual a 0, el proceso es manual. Sin embargo, el resultado es el mismo que en el proceso automático. Se puede apreciar la lógica del código en la figura 82. Para el proceso de registro de la salida de vehículos, se utilizó la misma vista del registro automático *RegisterDepartureView*.

Figura 82. Código de la vista RegisterEntryView

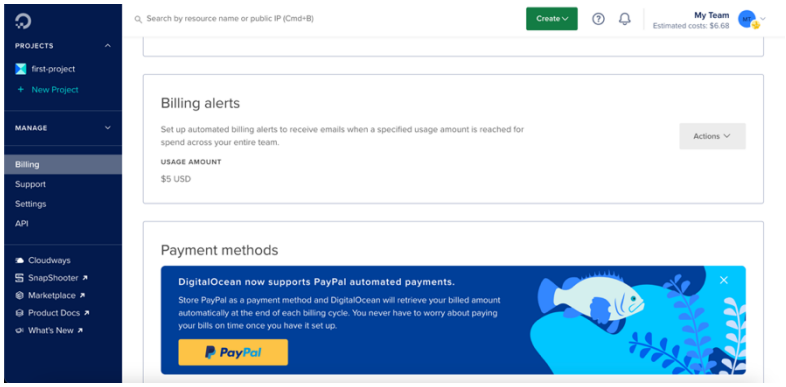
```python
# *Proceso de Registro Manual por parte del Guardia
if request.data['user'] == 0:
    try:
        vehicle = Vehicle.objects.get(placa=request.data['placa'])
        data = {"user": vehicle.owner.id if vehicle.owner != None else None, "vehicle": vehicle.id,
                "guard": request.data['guard']}
    except Vehicle.DoesNotExist:

        v = VehicleSerializer(data={"placa": request.data['placa']})
        if v.is_valid(raise_exception=True):
            vehicle = v.save(owner=None, register_manual=True)
        else:
            return Response(status=HTTP_400_BAD_REQUEST)
        data = {"user": None, "vehicle": vehicle.id,
                "guard": request.data['guard']}
else:
    # data = request.data
    data = {"user": request.data['user'], "vehicle": request.data['vehicle'],
            "guard": request.data['guard']}
```

4.3.5.2.3. Historia Técnica 2: Operabilidad. Escalabilidad - Despliegue en Producción

La información detallada en la historia técnica se encuentra en el anexo VI. Se utilizó la plataforma *Digital Ocean*, para cumplir con los requisitos no funcionales de operatividad y escalabilidad, en el despliegue a producción del Sistema *IoT* y la aplicación *web*. Por lo tanto, *se* creó una cuenta en la plataforma, se utilizó *PayPal* como medio de pago seguro

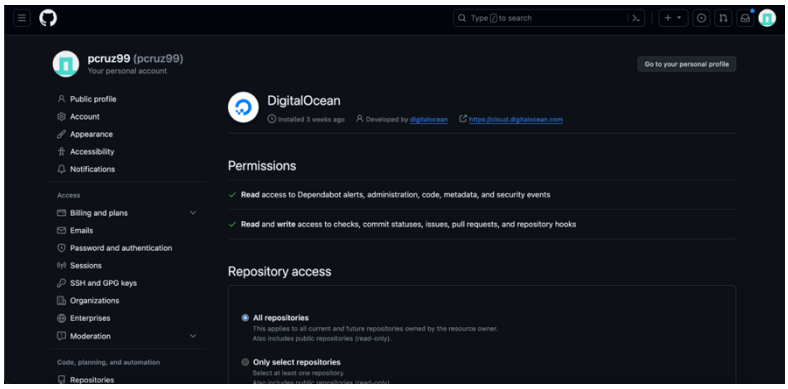
para poder utilizar los servicios necesarios de la plataforma. Este proceso se puede apreciar en la figura 83.

Figura 83. Sección de Facturación de DigitlOcean



Para poder desplegar aplicaciones con el código que se encuentra alojado en el repositorio de *GitHub*, se enlazó las cuentas y se dio permisos a *Digital Ocean* como se visualiza en la figura 84.

Figura 84. Enlace de cuentas entre GitHub y DigitlOcean



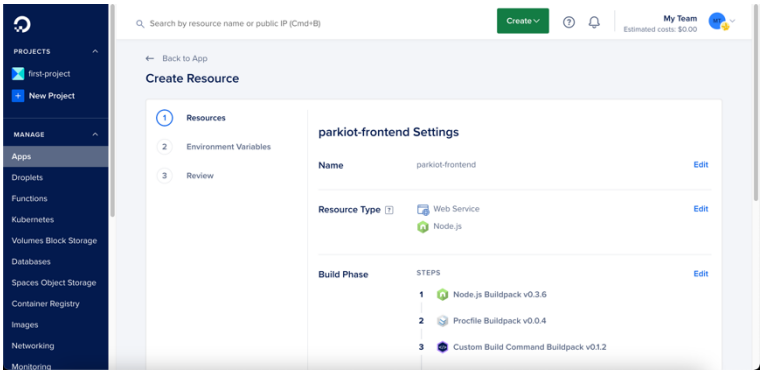
Con la cuenta de *GitHub* enlazada, se creó la aplicación en *Node.js* y *Python*, primero se definió el *script* para la construcción y el arranque de la aplicación de *Node.js*, en el archivo *package.json* como se observa en el código de la figura 85.

Figura 85. Código del Script package.json

```
},
Debug
"scripts": {
  "start": "node sammy.js; serve -s _static -l tcp://0.0.0.0:$PORT -n",
  "build": "rm -r _static 2>/dev/null; node sammy.js; react-scripts build && mv build _static",
  "test": "react-scripts test",
  "eject": "react-scripts eject"
},
"eslintConfig": {
```

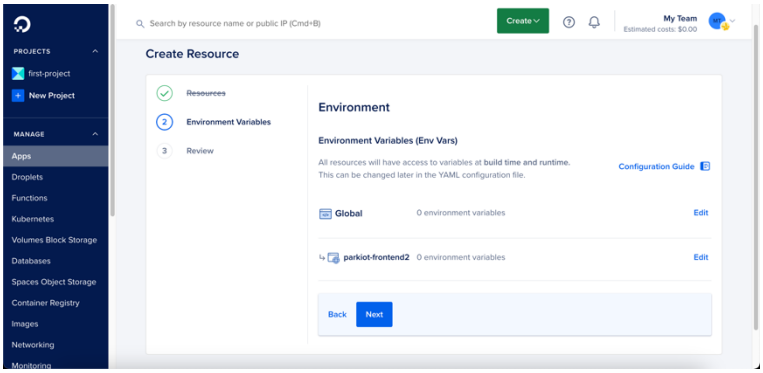
Posteriormente, se siguió los pasos de la plataforma para crear la aplicación de *frontend* con *Node.js*, como se aprecia en la captura de la figura 86.

Figura 86. Creación de la Aplicación Frontend en Node.js



Se definieron las variables de entorno, para la aplicación en *Node.js*, como se visualiza en la siguiente captura de la figura 87.

Figura 87. Definición de Variables de Entorno para Node.js

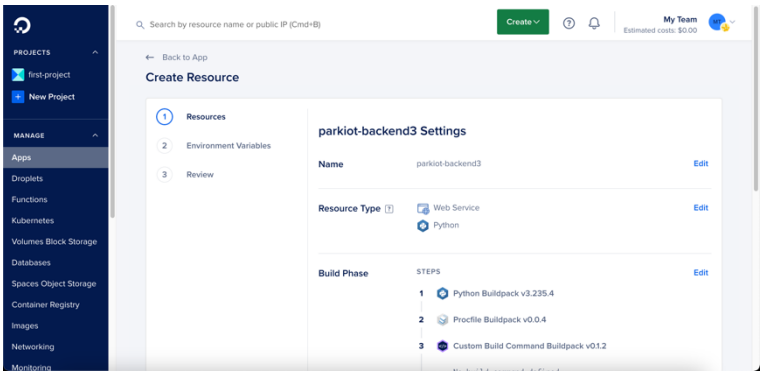


Finalmente, para la aplicación de *Node.js* en el *frontend*, se definió los *scripts* de construcción y ejecución, como se visualiza en la figura 88 del proceso de creación a la aplicación.

Figura 88. Scripts de ejecución en Node.js

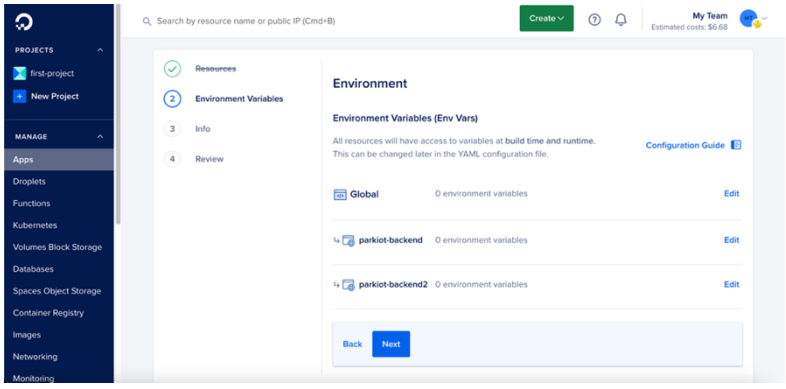
Por otro lado, se procedió a crear la aplicación correspondiente al *backend* con *Python*, por lo cual se siguió los pasos de la plataforma, como se aprecia en la captura de la figura 89.

Figura 89. Creación de la Aplicación Backend en Python



Se registró las variables de entorno necesarias en el apartado de configuración de la aplicación, para el correcto funcionamiento, este proceso se visualiza en la captura de la figura 90.

Figura 90. Sección para ingresar las variables de entorno



Por último, se definió el *script* de ejecución de la aplicación *backend* con *Python*, como se aprecia en la figura 91, se utilizó un servidor de *gunicorn* para ejecutar la aplicación.

Figura 91. Scripts de ejecución en Python

Se seleccionó el gestor de base de datos relacional *MySQL*, como se indica en la figura 92 y 93. Además, se determinó la región y se especificó las características físicas del servidor, así como la cantidad de almacenamiento.

Figura 92. Elección del SGBD y la región

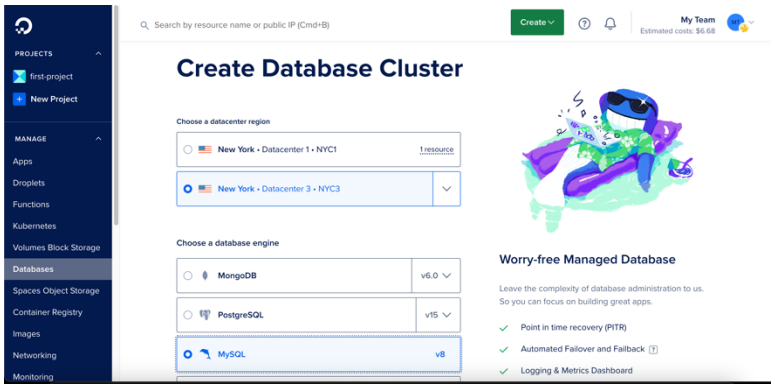
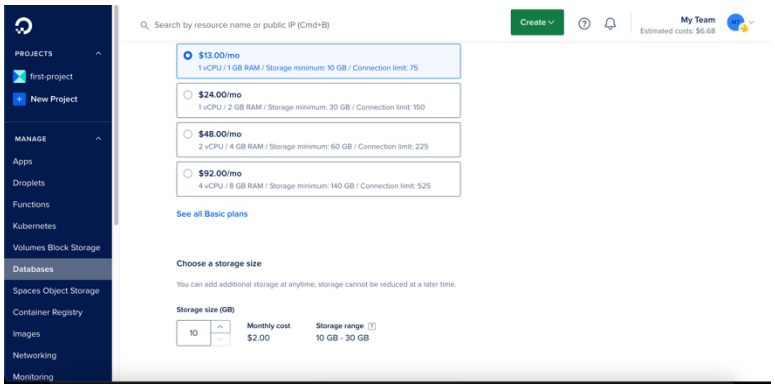
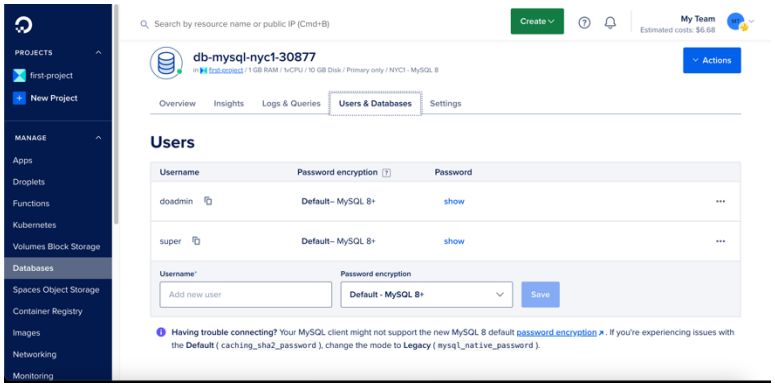


Figura 93. Elección de las características del servidor de base de datos



Se creó un usuario dedicado a la conexión a la aplicación llamado “*super*”, en la sección de la configuración que da acceso a la base de datos de manera segura desde el *backend*, la creación del usuario se visualiza en la figura 94.

Figura 94. Creación de usuario para la base de datos



Se creó la base de datos “*parkiot*”, que es accedida con el usuario “*super*” previamente creado, el proceso de creación se muestra en la figura 95 y 96. También, se realizó la asignación en el grupo seguro de conexiones, junto a la aplicación *backend*.

Figura 95. Creación de la base de datos para la aplicación

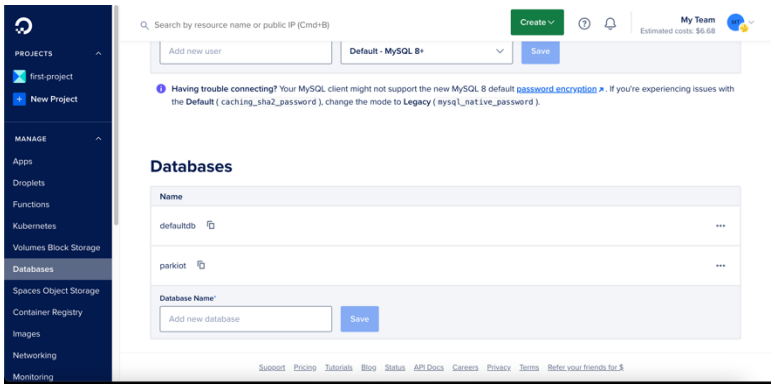
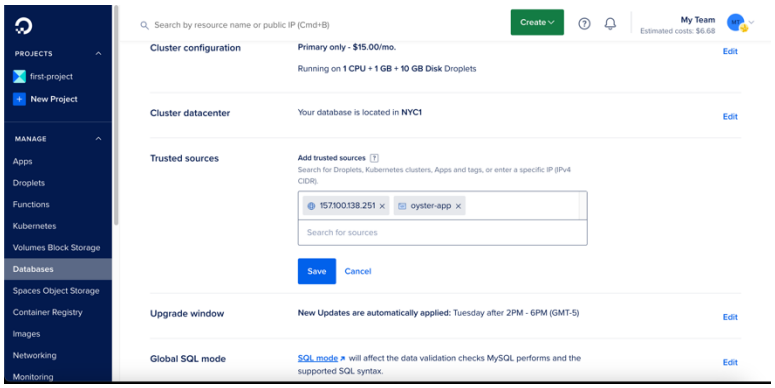


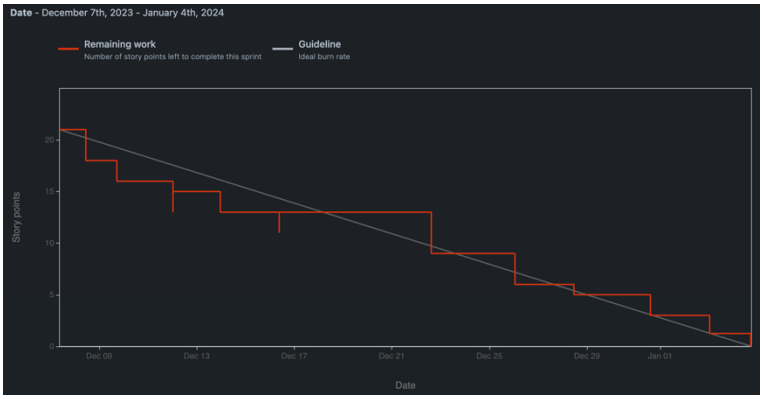
Figura 96. Grupo seguro de conexiones



4.3.5.2.1. Sprint 3 - Gráfico de trabajo pendiente

El gráfico de trabajo pendiente permitió representar el progreso, referente a las tareas de ingeniería faltantes, manifestando gráficamente si las actividades del desarrollo están adelantadas o atrasadas. Dada la figura 97, se visualiza el gráfico de trabajo pendiente correspondiente al *sprint* 3, que fue autogenerado mediante los reportes de la herramienta de gestión de incidencias *Jira*.

Figura 97. Gráfico de trabajo pendiente del Sprint 3 generado por Jira



4.3.5.3. Sprint 3 - Revisión (Sprint Review)

En la etapa de revisión, correspondiente al *sprint* 3, se llevó a cabo una reunión con el dueño del producto, se verificaron los avances y las mejoras en el sistema *IoT*. Además, se examinaron las historias de usuario en conjunto con sus escenarios de prueba. De la misma forma, se revisaron las pruebas de aceptación, las cuales se encuentran en el anexo VII.

Posteriormente, se consideró la creación de una maqueta con sensores, que representen el funcionamiento del estacionamiento vehicular, como referencia en el cumplimiento de las actividades del sistema *IoT* con *Machine Learning,* la propuesta se visualiza en la figura 98 y 99.

Figura 98. Vista Frontal de la maqueta con sensores del estacionamiento vehicular

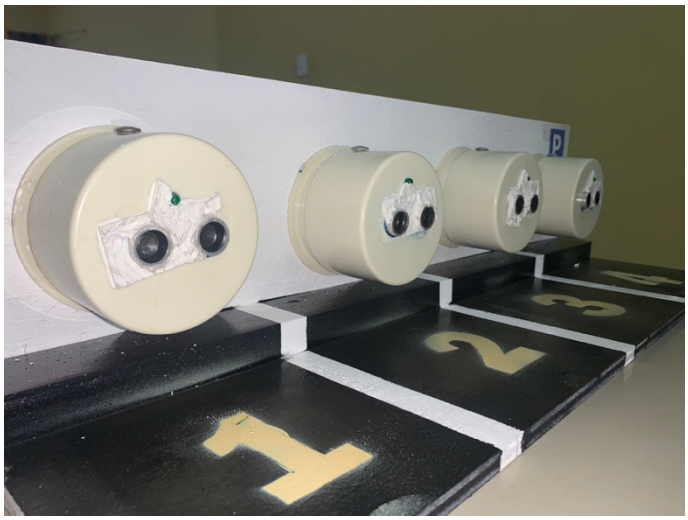
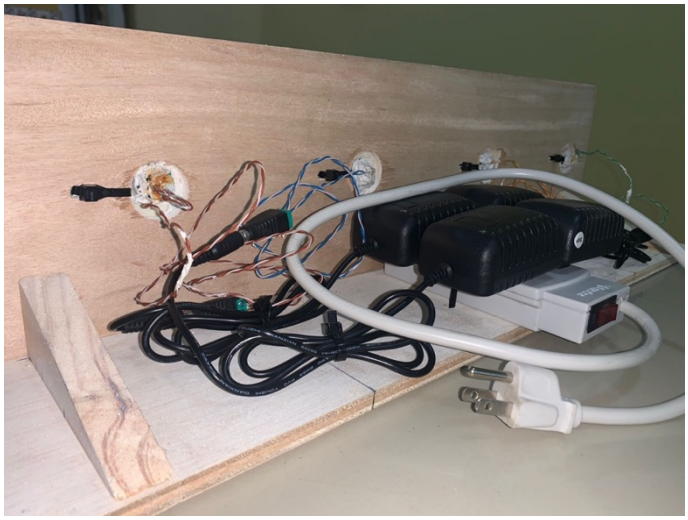


Figura 99. Vista Trasera de la maqueta con sensores del estacionamiento vehicular



4.3.5.4. Sprint 3 - Retrospectiva (Sprint Retrospective)

Dada la etapa de retrospectiva del *sprint* 3, se plantearon tres preguntas fundamentales, para obtener información relevante acerca del desarrollo del *sprint*, e identificar las mejoras que se derivan de los inconvenientes surgidos en el proceso. Las respuestas se reflejan en la Tabla 27.

Tabla 27. Sprint 3 - Retrospectiva

Aspectos exitosos en el Sprint	***Mejoras que se podrían aplicar en el siguiente Sprint (recomendaciones para la mejora continua)***	***Errores en el Sprint***
Durante el desarrollo del *sprint* 3, se logró concluir con éxito todas las tareas de ingeniería, completando así las funcionalidades de registro automático y registro manual de la entrada/salida de vehículos, además del despliegue en el *cloud* de los servicios.	Una sugerencia para la planificación del próximo *sprint*, sería realizar un análisis previo de las funcionalidades esenciales con el fin de evitar modificaciones en módulos previamente creados.	Dentro del proceso de desarrollo del *sprint* 3, se presentaron dificultades al verificar si un vehículo ya se encuentra dentro del parqueadero y que este proceso de validación funcione tanto en el registro manual como en el automático.

4.3.5.5. Implementación del Sistema *IoT* con *Machine Learning*

Completado el sprint 3, se procedió a realizar la implementacíon del sistema en el Comercial Miñaca del Cantón Santo Domingo, desplegando los sensores de detección vehicular en el estacionamiento, como se aprecia en las imágenes del anexo VIII y en el video del anexo IX.

Por otra parte, al finalizar la implementación del sistema, se publicó los datos de uso del estacionamiento vehicular, que se registraron entre el 4 de diciembre del 2023 y el 3 de enero del 2024. Estos datos fueron compartidos mediante la plataforma *Kaggle*, los cuales pueden ser accedidos, por medio del enlace que se encuentra en el anexo X.

4.4. Validación de la propuesta

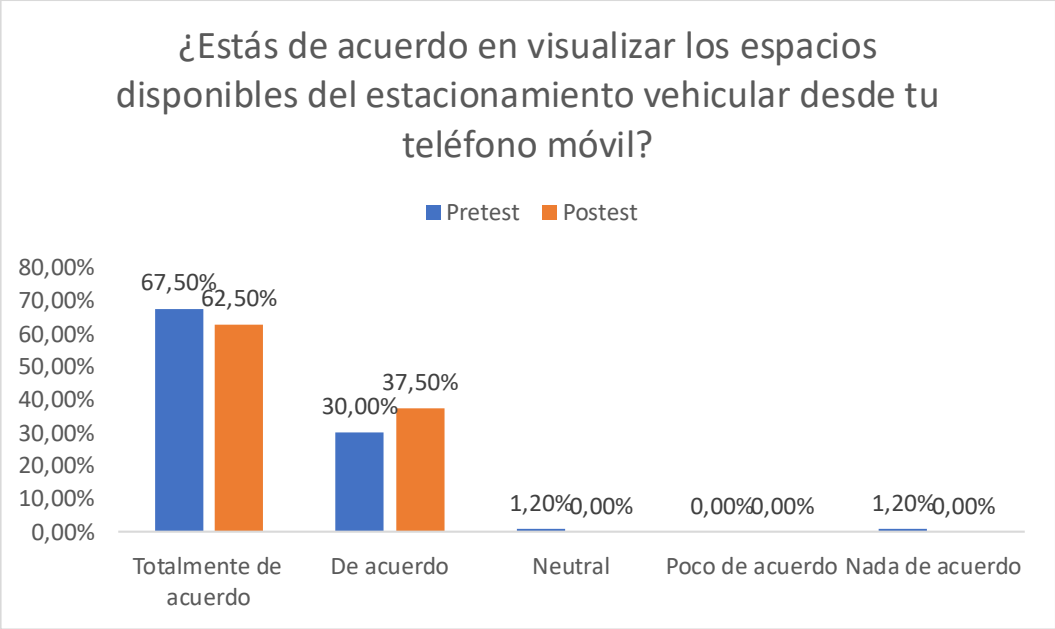
4.4.1. Resultados de la Encuesta

Mediante la ejecución de las encuestas a los clientes del Comercial Miñaca del Cantón Santo Domingo, se dio a conocer el nivel de satisfacción que tienen los clientes, sobre el control del estacionamiento vehicular, en base a las preguntas propuestas en el instrumento de recolección de información presente en el anexo III.

Se llevó a cabo un pre test y un post test, sobre la muestra de 80 clientes que utilizaron los primeros 4 estacionamientos de los 23 en total, donde se logró demostrar un cambio significativo en las respuestas de las preguntas correspondientes al control del estacionamiento vehicular. Como se evidencia en la comparación de resultados, se contemplaron 8 preguntas de la encuesta de 18 en total.

Pregunta 1: ¿Estás de acuerdo en visualizar los espacios disponibles del estacionamiento vehicular desde tu teléfono móvil?

Figura 100. Visualizar los espacios disponibles del estacionamiento vehicular desde el teléfono móvil

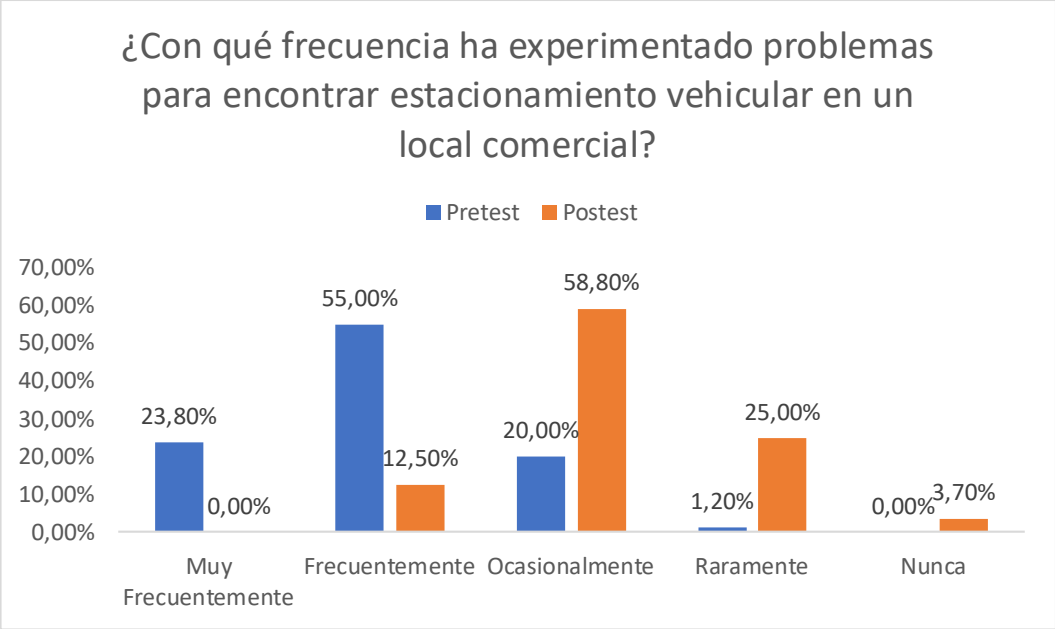


Análisis e interpretación: Dada la figura 100, se visualizan los resultados del *pre test*, donde se identifica que el 67.50% de los encuestados están totalmente de acuerdo en visualizar los estacionamientos disponibles desde el teléfono móvil, mientras que el 30% se mantienen de acuerdo, un 1.20% tienen una opinión neutral y un 1.20% no están de acuerdo. Demostrando de esta forma el interés que tienen los clientes sobre la aplicación.

Posteriormente, se realiza la misma encuesta a los clientes que utilizaron el sistema, obteniendo los resultados del *post test*, demostrando que el 62.50% están totalmente de acuerdo en visualizar los estacionamientos disponibles desde el teléfono móvil, mientras que el 37.50% están de acuerdo. Demostrando que el uso del sistema *IoT* con *Machine Learning*, incrementó el interés de los clientes por visualizar el estacionamiento desde el teléfono móvil.

Pregunta 2: ¿Con qué frecuencia ha experimentado problemas para encontrar estacionamiento vehicular en un local comercial?

Figura 101. Experiencia con problemas para encontrar estacionamiento vehicular en un local comercial

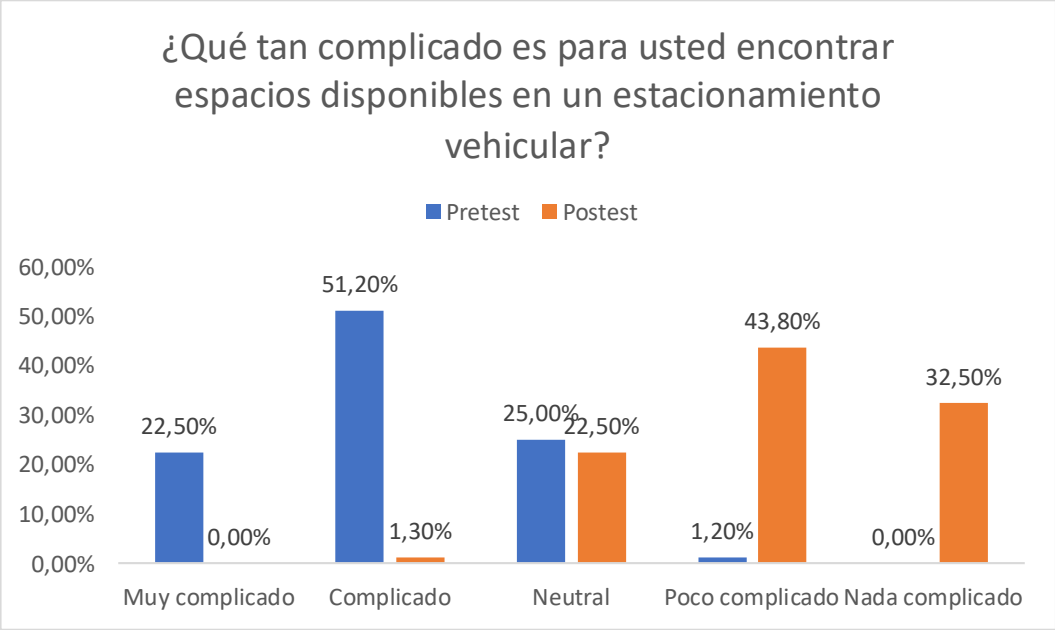


Análisis e Interpretación: En base a la figura 101, se aprecia los resultados pertinentes al *pre test*, que indican que un 55% de los encuestados frecuentemente experimentan problemas para encontrar estacionamiento vehicular en el local comercial y un 23.80% lo experimentan de manera muy frecuente, mientras que un 20% ocasionalmente tienen problemas y un 1.20% raramente experimentan problemas. Demostrando así el nivel de dificultad que los clientes tienen, al buscar un estacionamiento vehicular.

Al momento de realizar el *post test*, se encontra que únicamente el 12.50% de los clientes experimentan problemas para encontrar estacionamiento vehicular en el local comercial, mientras que un 58.80% ocasionalmente tiene problemas, un 25% raramente y un 3.70% nunca experimentan problemas, Se observa que después de implementar el sistema *IoT* con *Machine Learning*, disminuyeron los problemas para encontrar estacionamiento vehicular.

Pregunta 3: ¿Qué tan complicado es para usted encontrar espacios disponibles en un estacionamiento vehicular?

Figura 102. Complejidad en encontrar espacios disponibles en un estacionamiento

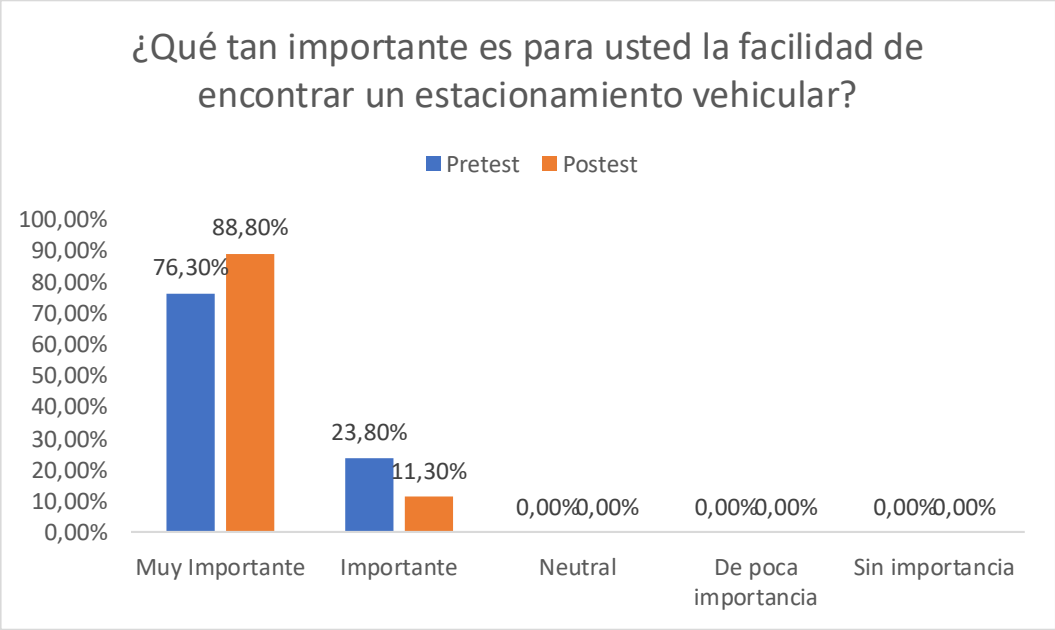


Análisis e Interpretación: Como se aprecia en la figura 102, los resultados del *pre test*, reflejan que un 51.20% de los encuestados tienen complicaciones para encontrar espacios disponibles en un estacionamiento vehicular y un 22.50% presentan mayor complejidad, mientras que un 25% tiene una experiencia neutral y solo un 1.20% pocas complicaciones. Por lo cual, los clientes han demostrado el nivel de complejidad que tienen a la hora de encontrar espacios disponibles.

Con los resultados del *post test*, se observa que solo el 1.30% ha tenido problemas para encontrar espacios disponibles en un estacionamiento vehicular, mientras que un 22.50% reflejó una opinión neutral, un 43.80% evidenció poca complicación y un 32.50% nada de complicación. Se demuestra que después de implementar el sistema *IoT* con *Machine Learning*, hubo una disminución en la complejidad para encontrar espacios disponibles en un estacionamiento.

Pregunta 4: ¿Qué tan importante es para usted la facilidad de encontrar un estacionamiento vehicular?

Figura 103. Facilidad de encontrar un estacionamiento vehicular

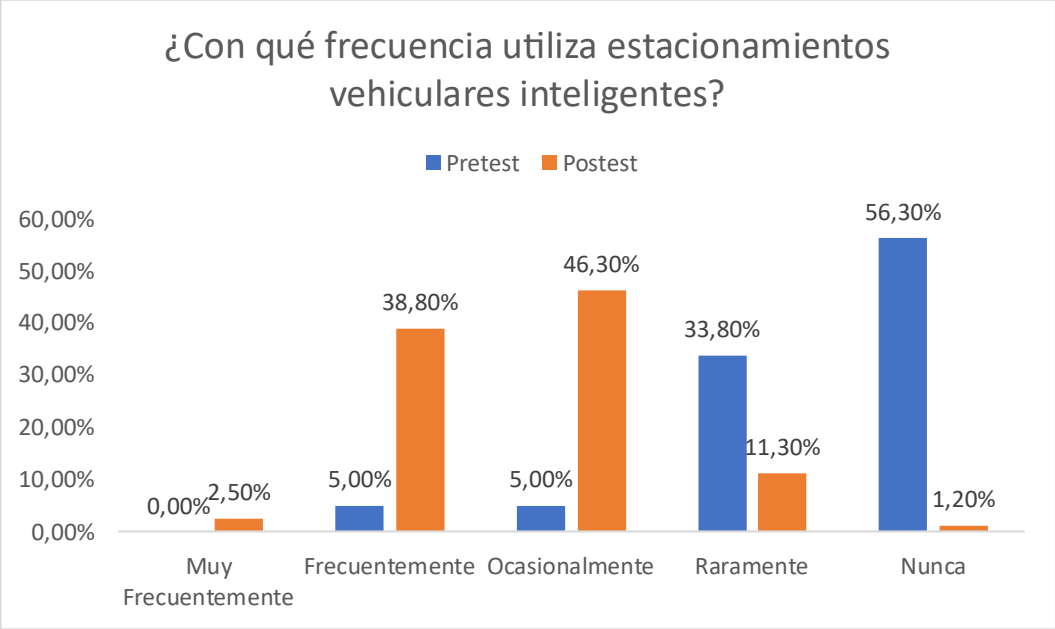


Análisis e Interpretación: Dada la figura 103, se puede apreciar los resultados del *pre test*, donde se demuestra que el 76.30% de los clientes consideran que es muy importante encontrar estacionamiento vehicular fácilmente, mientras que el otro 23.80% consideran que es importante. De esta manera los clientes demuestran la importancia de encontrar un estacionamiento de manera sencilla.

En base a los resultados del *post test*, se evidencia que el 88.80% de los encuestados considera que es muy importante encontrar estacionamiento fácilmente, mientras que el 11.30% restante refleja que es importante en menor medida. Por lo tanto, se refleja que la implementación del sistema *IoT* con *Machine Learning*, mejoró el nivel de aceptación que tienen los clientes sobre los procesos del estacionamiento.

Pregunta 5: ¿Con qué frecuencia utiliza estacionamientos vehiculares inteligentes?

Figura 104. Frecuencia de utilizar estacionamientos vehicular inteligentes

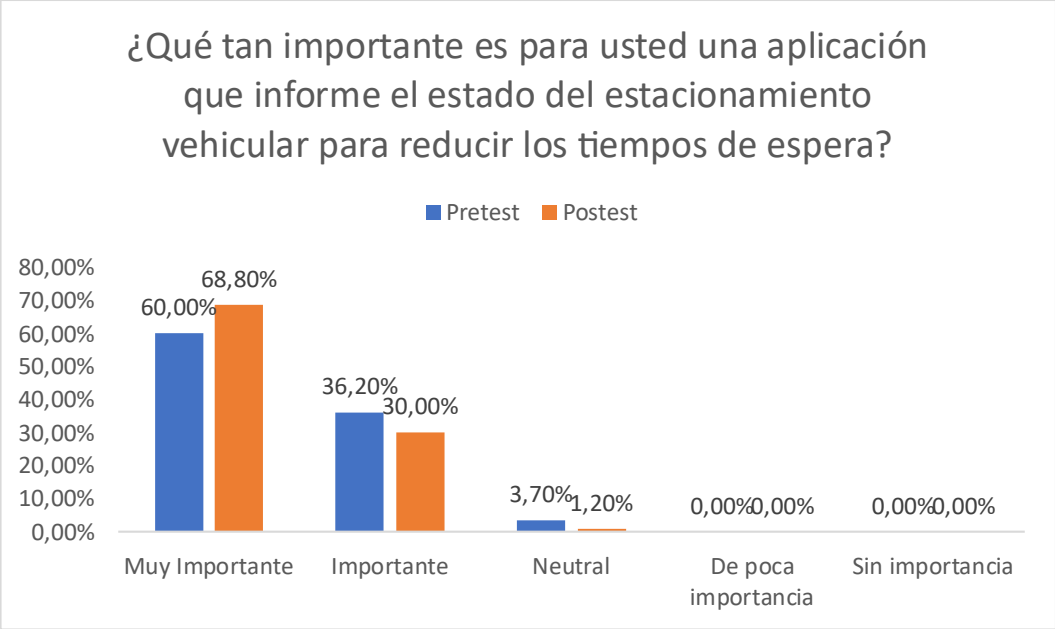


Análisis e Interpretación: Se puede apreciar en la figura 104, que los resultados del *pre test*, demuestran que un 56.30% de los encuestados nunca utiliza un estacionamiento vehicular inteligente, mientras que un 33.80% lo utiliza raramente, un 5% ocasionalmente y un 5% frecuentemente. Estos resultados demuestran el nivel de uso que tienen este tipo de estacionamiento por parte de los clientes.

Al momento de realizar el *post test*, se refleja que el 46.30% utiliza ocasionalmente estacionamientos vehiculares inteligentes, el 38.80% los utiliza frecuentemente y el 2.50% los utiliza muy frecuentemente, mientras que el 11.30% raramente los utiliza y el 1.20% nunca los utiliza. Con los datos, se demuestra que la implementación del sistema *IoT* con *Machine Learning*, mejoró el nivel de adopción en el uso de estacionamientos inteligentes.

Pregunta 6: ¿Qué tan importante es para usted una aplicación que informe el estado del estacionamiento vehicular para reducir los tiempos de espera?

Figura 105. Aplicación que informe el estado de un estacionamiento para reducir tiempos de espera

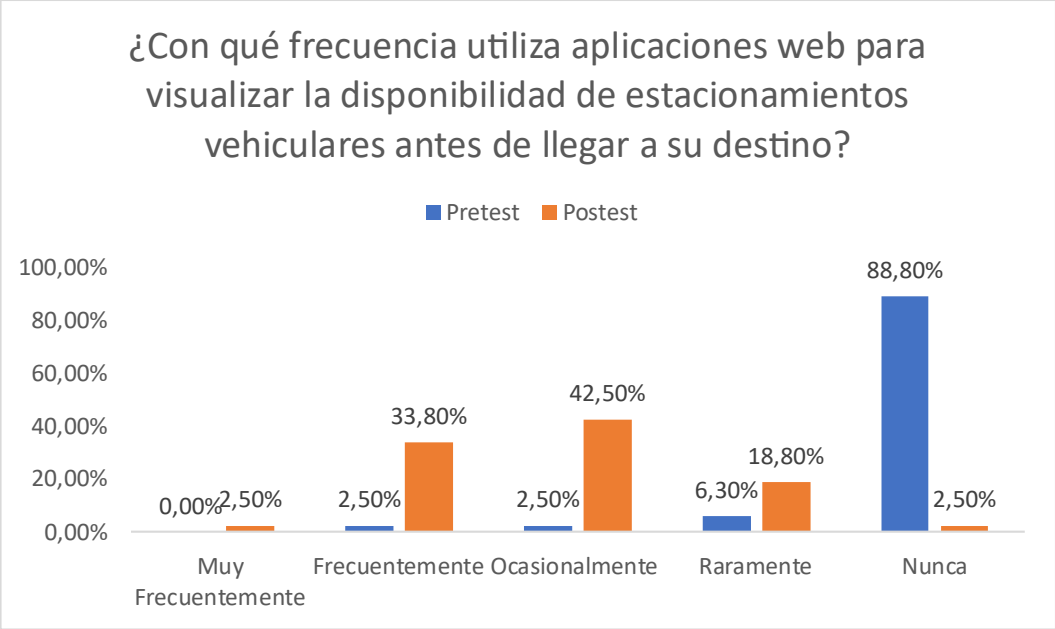


Análisis e Interpretación: Como se visualiza en la figura 105, los resultados del *pre test* reflejan que el 60% de los clientes consideran muy importante una aplicación que informe del estado del estacionamiento vehicular para reducir los tiempos de espera y un 36.20% consideran que es importante en menor medida, mientras que el 3.70% mantiene una opinión neutral. Estos resultados demuestran la importancia que tienen los clientes sobre la implementación de una aplicación.

En base a los resultados obtenido con el *post test*, se demuestra que el 68.80% de los encuestados consideran que es muy importante una aplicación que informe el estado de un estacionamiento para la reducción del tiempo de espera y un 30% considera que es importante en menor medida, mientras que el 1.20% tiene una opinión neutral. Por lo tanto, se refleja que la implementación del sistema *IoT* con *Machine Learning*, mejoró la aceptación de los clientes ante la capacidad de mejorar los tiempos de espera.

Pregunta 7: ¿Con qué frecuencia utiliza aplicaciones web para visualizar la disponibilidad de estacionamientos vehiculares antes de llegar a su destino?

Figura 106. Uso de aplicaciones web para visualizar la disponibilidad de estacionamientos vehiculares

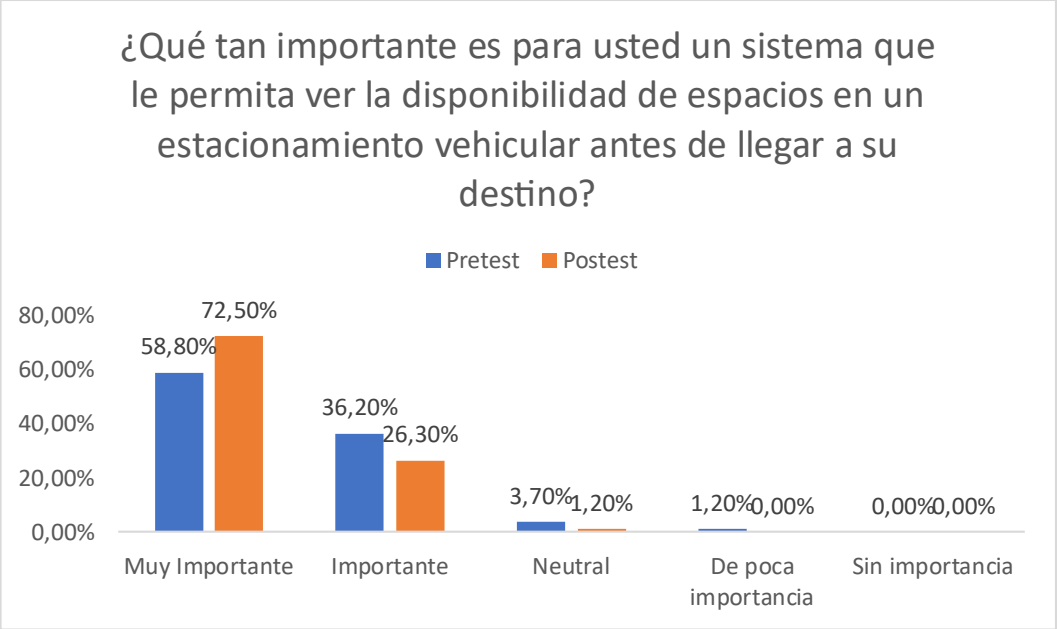


Análisis e Interpretación: Dada la tabla 106, los resultados del *pre test*, demuestran que el 88.80% de los clientes nunca utilizan una aplicación web para visualizar la disponibilidad de estacionamientos antes de llegar a su destino y un 6.30% raramente lo utiliza, mientras que un 2.50% menciona que frecuentemente lo usan y un 2.50% ocasionalmente. Por lo tanto, estos resultados reflejan el nivel de uso que tienen los clientes sobre aplicaciones *web* de estacionamientos vehiculares.

Con los resultados del *post test*, se refleja que el 42.50% de los encuestado utiliza ocasionalmente aplicaciones web para ver la disponibilidad de estacionamientos vehiculares, un 33.80% lo utilizan frecuentemente y un 2.50% lo utilizan con mucha frecuencia, mientras que 18.80% raramente utiliza aplicaciones web para este propósito y un 2.50% nunca lo utiliza. Con estos datos, se demuestra que la implementación del sistema *IoT* con *Machine Learning*, mejoró el nivel de uso de aplicaciones *web*, enfocándose en ver la disponibilidad de estacionamientos vehiculares antes de llegar a su destino.

Pregunta 8: ¿Qué tan importante es para usted un sistema que le permita ver la disponibilidad de espacios en un estacionamiento vehicular antes de llegar a su destino?

Figura 107. Sistema que permite ver la disponibilidad de un estacionamiento antes de llegar a su destino



Análisis e Interpretación: De acuerdo a lo visualizado en la figura 107, los resultados del *pre test*, reflejan que el 58.80% de los clientes consideran muy importante un sistema que permita ver la disponibilidad de espacios de estacionamiento y un 36.20% lo consideran importante en menor medida, mientras que un 3.70% tienen una opinión neutral sobre la importancia. Demostrando así, el nivel de importancia que tienen los clientes, sobre los sistemas para ver la disponibilidad de espacios.

En base a los resultados del *post test*, se determina que el 72.50% de los encuestados considera que es muy importante un sistema que permita ver la disponibilidad antes de llegar al estacionamiento y un 26.30% considera que es importante en menor medida, mientras que el 1.20% restante mantiene una respuesta neutral sobre la importancia. Por lo tanto, la implementación del sistema *IoT* con *Machine Learning* en el estacionamiento vehicular, mejoró el nivel de aceptación por parte de los clientes, sobre un sistema que refleja el estado del parqueadero antes de llegar al destino.

4.5. Validación de la Hipótesis

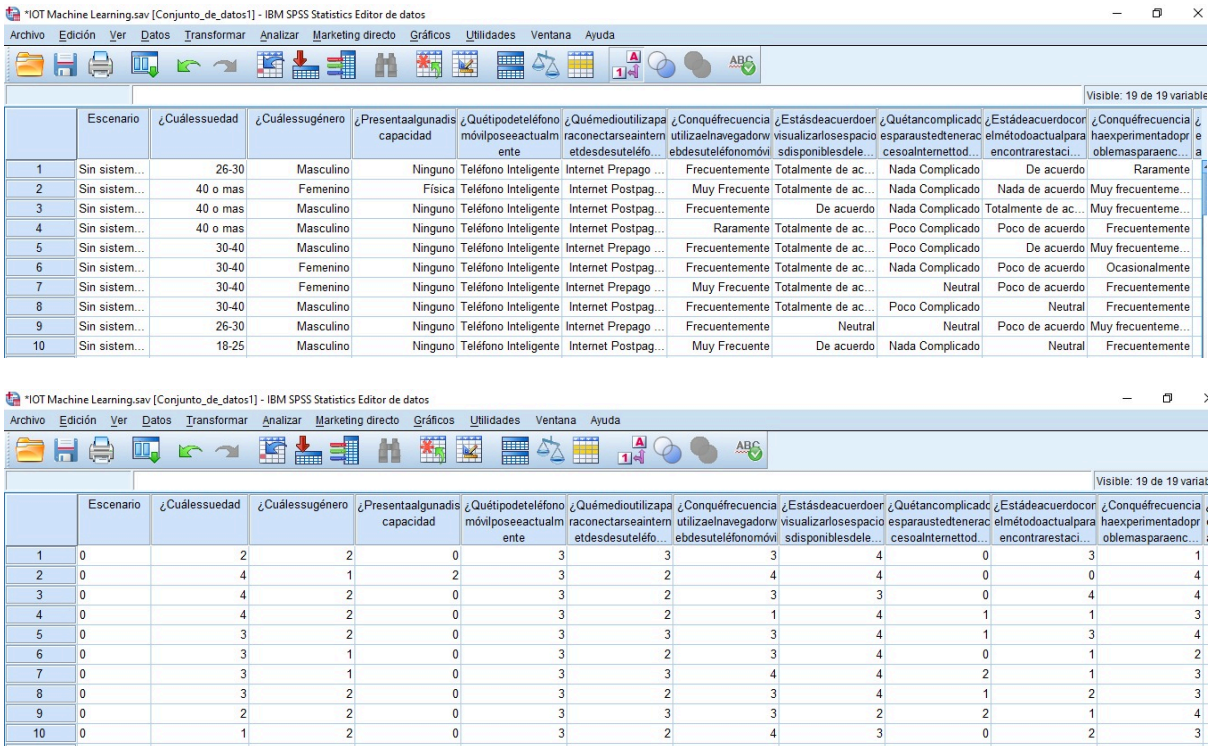
El proceso de validación de la hipótesis, tiene como requisito recodificar los escenarios del pre test y el post test. Además, es necesario recodificar las respuestas a las preguntas definidas en el instrumento de recolección de datos. La codificación de los escenarios se visualiza en la tabla 28.

Tabla 28. Recodificación de Escenarios

Recodificación	Escenarios
0	Sin sistema IoT y sin Machine Learning
1	Con sistema IoT y Machine Learning

Se recodificaron los resultados de las encuestas, tanto para la escala de Likert y las respuestas no estructuradas, estos valores recodificados se pueden visualizar en el Anexo XIV. Se emplearon los resultados recodificados de las encuestas, para desarrollar un análisis utilizando el *software IBM SPSS*, este proceso se evidencia en la figura 108.

Figura 108. Proceso de Análisis en el SPSS (IBM Corporation, 2011)



Se llevó a cabo un análisis estadístico, mediante regresión logística binaria, con grado de libertad (gl) en 1, además de la significancia de probabilidad (p) relacionado a las

preguntas que constan en el instrumento de recolección de datos, los resultados se pueden visualizar en la tabla 29. Donde se evidencia: la frecuencia de uso del navegador *web*, la aceptación de accesibilidad del estacionamiento, la complejidad para encontrar estacionamiento, la importancia en la facilidad de búsqueda, la frecuencia de uso de estacionamientos inteligentes y aplicaciones *web* para ver la disponibilidad, la importancia en visualizar los espacios de estacionamiento y el nivel de satisfacción sobre el estacionamiento. Todas estas interrogantes cumplen con la significancia de probabilidad menor a 0.05 ($p < 0.05$).

Tabla 29. Estudio cruzado en base a los parámetros del sistema IoT

Preguntas	**Puntuación**	**gl**	**p**
¿Con qué frecuencia utiliza el navegador web de su teléfono móvil?	8,682	1	,003
¿Está de acuerdo con el método actual para encontrar estacionamiento vehicular accesible para personas con discapacidad?	4,336	1	,037
¿Con qué frecuencia ha experimentado problemas para encontrar estacionamiento vehicular en un local comercial?	68,883	1	,000
¿Qué tan complicado es para usted encontrar espacios disponibles en un estacionamiento vehicular?	103,578	1	,000
¿Qué tan importante es para usted la facilidad de encontrar un estacionamiento vehicular?	4,329	1	,037
¿Con qué frecuencia utiliza estacionamientos vehiculares inteligentes?	87,914	1	,000
¿Con qué frecuencia utiliza aplicaciones web para visualizar la disponibilidad de estacionamientos vehiculares antes de llegar a su destino?	103,374	1	,000
¿Qué tan importante es para usted un sistema que le permita ver la disponibilidad de espacios en un estacionamiento vehicular antes de llegar a su destino?	4,348	1	,037
¿Qué te parece el servicio de estacionamiento vehicular que ofrece el local comercial?	47,904	1	,000

Por lo tanto, los datos obtenidos en la tabla 29, permiten validar la hipótesis alternativa (H1), determinando que, el sistema *IoT* con *Machine Learning* incide significativamente en el control de estacionamiento vehicular en el comercial Miñaca del cantón Santo Domingo.

5. DISCUSIÓN

A través de la aplicación de instrumentos de recolección de datos, se realizaron encuestas a los clientes, que contribuyeron a proporcionar información de las necesidades que tenían, sobre el estacionamiento vehicular, identificando así los requisitos mencionados en el primer objetivo de éste trabajo de titulación. Donde se determinó que los clientes están totalmente de acuerdo en que se pueda visualizar la disponibilidad de plazas de estacionamiento desde sus teléfonos inteligentes. Con este propósito, la implementación de los sistemas de control de estacionamiento surge como una solución clave, para elevar la experiencia y satisfacción del usuario en términos de reducir los tiempos de espera y verificar la disponibilidad de plazas. Este criterio se alinea con lo mencionado por Elsonbaty y Shams (2020), sobre la reducción del tiempo de búsqueda y la disponibilidad en tiempo real, junto con la perspectiva de Richard (2015), sobre la asignación eficiente de espacios.

Por otro lado, el segundo objetivo define las herramientas y tecnologías de desarrollo adecuadas para la implementación del control de estacionamiento vehicular, para conseguir este objetivo se aplicaron varios análisis comparativos entre tecnologías, que cumplen con las necesidades requeridas. Se utilizaron servicios en la nube que ayudaron en el desligue de la aplicación desarrollada en *React* y *Django*, además de la base de datos en *MySQL*, gracias a una tabla comparativa se optó por desplegar estos servicios en *Digital Ocean*, ya que facilita el proceso de despliegue, ayudando a los desarrolladores concentrarse únicamente en el código y no en la gestión de infraestructura, además de tener costos mensuales flexibles al presupuesto de un proyecto. Este criterio se ajusta con lo mencionado por D'Agostino (2019), donde la computación en la nube permite ahorrar costes y facilita la asignación de recursos de manera flexible, además que los servicios son suministrados mediante el modelo de pago por uso.

Otra área de desarrollo fueron los sensores, donde se utilizó el sensor ultrasónico *HC-SR04* por su capacidad de detección a gran distancia, junto con el microcontrolador

ESP8266 que cumple las necesidades técnicas requeridas y mediante la conexión a un servidor con *Ubuntu Server*. Estos sensores, una vez desplegados en el estacionamiento, se ocupan en detectar la presencia de vehículos y reportar al sistema su estado. Esto se alinea con lo documentado por Nagaraj (2021), donde los sensores son dispositivos de características básicas que pueden ser desplegados en cualquier lugar para monitorear un objeto de interés.

Además, para lograr el mismo objetivo, como funcionalidad para el soporte de toma de decisiones de negocios, se aplicó el algoritmo supervisado de *Machine Learning*, regresión logística, sobre los datos generados por el sistema, que soporta una cantidad finita de valores de entrada, para generar un pronóstico de dos clases como salida, los resultados generados por el algoritmo se pueden mejorar gracias a la capacidad de adoptar nuevos datos de entrada. Esto coincide con lo alegado por Balaji et al. (2021), donde la regresión logística solo tiene dos resultados posibles que representen la variable dependiente, utilizando las variables independientes como entrada para predecir la probabilidad como resultado.

Asimismo, para lograr el tercer objetivo, que corresponde al desarrollo del sistema *IoT* con *Machine Learning* para el control de estacionamiento vehicular. Se utilizó el marco de trabajo ágil *Scrum*, ya que permitió agilizar el desarrollo del sistema, optimizando el tiempo mediante la asignación y estimación de tareas de ingeniería e incidencias, además de permitir cambios durante la ejecución de tareas, facilitando mejoras entre las transiciones de los s*prints*. Esto se alinea con lo mencionado por Schwaber y Sutherland (2020), donde el marco de trabajo *Scrum* facilita la gestión y estructuración de trabajo a grandes equipos, con el objetivo de generar valor mediante la creación de soluciones adaptables a desafíos complejos.

Finalmente, para ejecutar el proceso de validación de la hipótesis, sobre los resultados de las encuestas en el *pre test* y *post test*. Se realizó un análisis estadístico, mediante regresión logística binaria, con el *software IBM SPSS*, donde se obtuvieron

indicadores sobre las variables que inciden en el sistema *IoT* con *Machine Learning*. Esto coincide con las 9 preguntas de 18, que demostraron tener un nivel de significancia menor al 0.05, permitiendo validar la hipótesis alternativa (H1).

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Se concluye que, tras llevar a cabo las encuestas a los clientes y la entrevista al gerente de manera satisfactoria, se evidenció un interés muy marcado, en la incorporación de sistemas inteligentes en el estacionamiento vehicular, dando paso a la oportunidad de fortalecer las funciones mediante elementos tecnológicos que son altamente aceptados por parte de los clientes. El gerente del comercial mostró un gran interés en la propuesta, por lo que facilitó la información necesaria para conocer el flujo de actividades existente en el proceso de estacionamiento de vehículos.

En relación a las herramientas y tecnologías implementadas, se desarrollaron cuadros comparativos que ayudaron a determinar las mejores soluciones. Como resultado, se emplearon servicios en *Digital Ocean*, por su flexibilidad en el despliegue de la aplicación web con *React, Django y MySQL*, ahorrando tiempo y recursos en el proceso de desarrollo. Además, se crearon sensores con el *ESP8266* y el *HC-SR04*, conectado a un servidor con *Ubuntu Server*, lo que simplificó el desarrollo para el proceso de detección de obstáculos, demostrando la eficiencia que tiene a la hora de detectar un vehículo. A su vez, la incorporación del algoritmo de *Machine Learning* regresión logística, redujo la complejidad para ejecutar pronósticos con datos limitados, evidenciando la capacidad de introducir nuevos datos que son previamente limpiados y manipulados.

Por otra parte, la flexibilidad que ofreció el marco de trabajo *Scrum*, permitió realizar mejoras continuas durante los *sprints*, optimizando así el tiempo de desarrollo mediante la asignación eficiente de tareas de ingeniería e incidencias. Como resultado, la aplicación de este marco de trabajo se destacó como una estrategia clave, para lograr los objetivos propuestos, ya que, al ser iterativa, demostró una gestión estructurada y eficaz en la planificación y ejecución de tareas.

Finalmente, con la aplicación del análisis estadístico en la validación de la hipótesis, se demostró mediante los datos obtenidos en el *pre test* y *post test*, que el sistema *IoT* con *Machine Learning* incide significativamente en el control de estacionamiento vehicular en el comercial Miñaca del cantón Santo Domingo.

6.2. Recomendaciones

Con el paso del tiempo, la experiencia de uso va cambiando, por lo que se recomienda realizar evaluaciones periódicas a los clientes, para identificar funcionalidades que agreguen nuevo valor al sistema de estacionamiento vehicular. Además, que se realicen visitas al gerente del comercial, con el fin de obtener retroalimentación a través de nuevas ideas, actualizando propiedades que permita adaptarse a los nuevos cambios e implementar estrategias en consecuencia de ello.

Por otra parte, la constante evolución tecnológica, impulsa la mejora y actualización de los sistemas, por tal motivo, se recomienda que, ante la adopción de servicios en la nube, se evalúen nuevas soluciones que mejor se adapten a la aplicación, esto se consigue mediante el estudio de la documentación y la ayuda de la comunidad en foros.

En relación a la implementación de sensores, es importante evaluar periódicamente las opciones tecnológicas disponibles, para asegurar la precisión y eficiencia de un sistema de detección de vehículos, ya que la innovación constante de este, ofrece mejores oportunidades para la gestión de plazas de estacionamiento.

En el ámbito de *Machine Learning*, se recomienda mantenerse al tanto de los avances constantes en algoritmos y técnicas supervisadas, para aumentar la precisión en el método de predicción. Así mismo, es necesario tener una mayor cantidad de datos, para tomar decisiones más exactas y adaptativas para los negocios.

Además, la adopción del marco de trabajo *Scrum*, ha demostrado ser fundamental para lograr el éxito en el desarrollo del sistema de control de estacionamiento vehicular. Por esta razón, se recomienda realizar retrospectivas al final de cada *sprint* para evaluar el

desempeño del equipo, identificar áreas de mejora y ajustar la planificación según sea necesario. Además, se recomienda capacitar constantemente a los integrantes del equipo, sobre el correcto uso del marco de trabajo.

Para terminar, el análisis estadístico permitió validar la hipótesis, por lo que se recomienda, que los instrumentos de recolección de datos, se revisen por expertos en el tema de estacionamientos vehiculares, con el objetivo de depurar las preguntas, para que el análisis sea lo más preciso.

7. REFERENCIAS

AG Electrónica. (2019). *Oky3127: sensor de proximidad infrarrojofc-51*.

Al-Turjman, F., & Malekloo, A. (2019). Smart parking in IoT-enabled cities: A survey. *Sustainable Cities and Society*, *49*, 101608. https://doi.org/10.1016/j.scs.2019.101608

Arduino. (2023). *MKR 1000 wifi*. https://docs.arduino.cc/hardware/mkr-1000-wifi

Arévalo Chávez, P., Cárdenas, J. C., Maldonado, G., Fierro, P., Bonilla Bedoya, S., Bastidas, A. E., Lanas, J. G., Zapata Rodríguez, M., Guerrero, J. J., Flores, H. A., Ramos, C., Patricio, G., Chávez, A., & Ramos Galarza, C. (2020). *Actualización en metodología de la investigación científica*. Editorial universidad tecnológica Indoamérica.

Arjona, J., Linares, M., Casanovas-Garcia, J., & Vázquez, J. J. (2020). Improving Parking Availability Information Using Deep Learning Techniques. *Transportation Research Procedia*, *47*, 385–392. https://doi.org/10.1016/j.trpro.2020.03.113

Asamblea Nacional de la República del Ecuador. (2021). *Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial. Agencia Nacional de Tránsito A.N.T*. https://www.ant.gob.ec/wp-admin/admin-ajax.php?juwpfisadmin=false&action=wpfd&task=file.download&wpfd_category_id=469&wpfd_file_id=7531

Atlassian. (2023). *Jira*. https://www.atlassian.com/es/software/jira

Babic, M., Vekic, A., Stanojevic, M., Ostojic, G., Borocki, J., & Stankovski, S. (2019). *Modern Parking Solutions for Smart Cities* (pp. 1075–1083). https://doi.org/10.2507/30th.daaam.proceedings.150

Baena Paz, Guillermina. (2017). *Metodología de la investigación* (3rd ed.). Grupo Editorial Patria.

Balaji, T. K., Annavarapu, C. S. R., & Bablani, A. (2021). Machine learning algorithms for social media analysis: A survey. *Computer Science Review*, *40*, 100395. https://doi.org/10.1016/j.cosrev.2021.100395

Bansal, M., & Priya. (2021). *Performance Comparison of MQTT and CoAP Protocols in Different Simulation Environments* (Vol. 145, pp. 549–560). https://doi.org/10.1007/978-981-15-7345-3_47

Bizagi. (2023). *Bizagi Modeler*. https://www.bizagi.com/es/plataforma/modeler

Campesato, O. (2020). *Artificial Intelligence, Machine Learning, and Deep Learning*. Mercury Learning and Information.

Canonical. (2023). *Ubuntu Server documentation*. https://ubuntu.com/server/docs

Canva. (2023). *Canva*. https://www.canva.com

Cohn, M. (2004). *User Stories Applied for Agile Software Development* (1st ed.). Addison-Wesley Professional.

Cohn, M. (2005). *Agile Estimating and Planning* (1st ed.). Perason Education.

Collin, D., & Perry, L. (2019). *Mastering IOT: Build Modern IoT Solutions That Secure and Monitor Your IoT Infrastructure*. Packt Publishing Ltd.

D'Agostino, G. (2019). *Data Security in Cloud Computing, Volume I* (1st ed.). Momentum Press.

DigitalOcean. (2023). *Product Docs Home*. https://docs.digitalocean.com/products/

Django. (2023). *Django Documentation*. https://docs.djangoproject.com/en/5.0/

Elsonbaty, A., & Shams, M. (2020). The Smart Parking Management System. *International Journal of Computer Science and Information Technology*, *12*(4), 55–66. https://doi.org/10.5121/ijcsit.2020.12405

Enciso, L., Sarango, J., Valladarez, A., & Condolo, J. (2019). Aplicación Móvil para un Aparcamiento Inteligente~ A Mobile Application for a Smart Car. *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–7. https://doi.org/10.23919/CISTI.2019.8760804

Espressif Systems. (2023). *Esp8266ex datasheet*.

European Parking Association. (2013). *The scope of parking in europe european parking association - eap*. https://www.europeanparking.eu/media/1180/epa_data_collection_rev.pdf

Gacovski, Z. (2019). *Internet of Things*. Arcler Press. https://ebsco.puce.elogim.com/login.aspx?direct=true&db=nlebk&AN=2013945&lang=es&site=ehost-live&ebv=EB&ppid=pp_iv

Gobierno Autónomo Descentralizado Municipal de Santo Domingo. (2020). *Ordenanza Municipal M-009-WEA Empresa Pública Municipal de Transporte Santo Domingo*. https://www.epmtsd.gob.ec/ordenanzas/views/modules/ordenanzas/Ordenanza%20Municipal%20No.%20M-009-WEA.pdf

Golrokh, M., & Hojjat, A. (2022). Machine learning techniques for diagnosis of alzheimer disease, mild cognitive disorder, and other types of dementia. *Biomedical Signal Processing and Control*, *72*, 103293. https://doi.org/10.1016/j.bspc.2021.103293

Google. (2023a). *Angular*. https://angular.io/docs

Google. (2023b). *Google Cloud Documentation*. https://cloud.google.com/docs

Hanes, D., Salgueiro, G., Grossetete, P., Barton, R., & Henry, J. (2017). *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*. Cisco Press.

Hernández Sampieri, R., & Mendoza Torres, C. P. (2018). *Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta* (1st ed.). McGRAW-HILL.

Ho, W. K. O., Tang, B.-S., & Wong, S. W. (2021). Predicting property prices with machine learning algorithms. *Journal of Property Research*, *38*(1), 48–70. https://doi.org/10.1080/09599916.2020.1832558

IBM Corporation. (2011). *SPSS Statistics Versión 20*. IBM.

Iglesias-Urkia, M., Mansilla, D. C.-, Mayer, S., & Urbieta, A. (2018). Validation of a CoAP to IEC 61850 Mapping and Benchmarking vs HTTP-REST and WS-SOAP. *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1015–1022. https://doi.org/10.1109/ETFA.2018.8502624

Instituto Ecuatoriano de Normalización. (2016). *Accesibilidad de las personas al medio físico. Estacionamientos norma técnica ecuatoriana INEN 2248*. https://www.habitatyvivienda.gob.ec/wp-content/uploads/downloads/2018/06/NTE-INEN-2248-ESTACIONAMIENTOS.pdf

Instituto Nacional de Estadística y Censos. (2022a). *Boletín Técnico N° 01-2022-Transporte Anuario de Estadísticas de Transportes, 2021*. https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Economicas/Estadistica%20de%20Transporte/ESTRA_2021/2021_BOLETIN_ESTRA.pdf

Instituto Nacional de Estadística y Censos. (2022b). *Boletín Técnico N°01-2022-ESED Estadísticas de Edificaciones (ESED), 2021*. https://www.ecuadorencifras.gob.ec/documentos/web-

inec/Estadisticas_Economicas/Encuesta_Edificaciones/2021/3.%202021_ESED_Boletin_tecnico.pdf

Instituto Nacional de Estadística y Censos. (2022c). *Tabulados y series históricas Anuario de Estadísticas de Transportes, 2021*. https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Economicas/Estadistica%20de%20Transporte/ESTRA_2021/2021_TABULADOS%20ESTRA.xlsx

Kline, K., & Kline, D. (2001). *SQL in a Nutshell* (Gigi Estabrook, Ed.; 1st ed.). O'Reilly & Associates.

Laravel. (2023). *Documentation*. https://laravel.com/docs/10.x

Manel, M.-R., Gupta, A., Rojo-Alvarez, J. L., & Christodoulou, C. (2021). *Machine Learning Applications in Electromagnetics and Antenna Array Processing*. ARTECH HOUSE.

Mejía, F. (2021). *Propuesta de solución Internet of Things (IoT) con tecnología LoRaWAN, para un Sistema de Parqueadero Inteligente (SPI). Caso de estudio Conjunto Habitacional Parque Real (CHPR)* [Pontificia Universidad Catolica del Ecuador]. http://repositorio.puce.edu.ec:80/handle/22000/18672

Meta Open Source. (2023). *React*. https://react.dev/

Microsoft. (2023a). *Azure Documentation*. https://learn.microsoft.com/en-us/azure/?product=popular

Microsoft. (2023b). *GitHub*. https://github.com/

Microsoft. (2023c). *Microsoft SQL documentation*. https://learn.microsoft.com/en-us/sql/?view=sql-server-ver16

Microsoft. (2023d). *Windows Server Documentation*. https://learn.microsoft.com/en-us/windows-server/

Morgan, E. J. (2014). *HCSR04 Ultrasonic Sensor*.

Naciones Unidas CEPAL. (2019). *La Agenda 2030 y los Objetivos de Desarrollo Sostenible: una oportunidad para América Latina y el Caribe. Objetivos, metas e indicadores mundiales*. https://repositorio.cepal.org/bitstream/handle/11362/40155/24/S1801141_es.pdf

Nagaraj, A. (2021). *Introduction to Sensors in IoT and Cloud Computing Applications*. Bentham Books imprint.

Naik, G. P., & Bapat, A. U. (2020). A Brief Comparative Analysis on Application Layer Protocols of Internet of Things: MQTT, CoAP, AMQP and HTTP. *International Journal of Computer Science and Mobile Computing*, *9*(9), 135–141. https://doi.org/10.47760/IJCSMC.2020.v09i09.014

Nikolov, N. (2020). Research of MQTT, CoAP, HTTP and XMPP IoT Communication protocols for Embedded Systems. *2020 XXIX International Scientific Conference Electronics (ET)*, 1–4. https://doi.org/10.1109/ET50336.2020.9238208

Oracle. (2023). *MySQL Documentation*. https://dev.mysql.com/doc/

Organización Internacional del Trabajo. (2021). *El futuro del trabajo en la industria automotriz y la necesidad de invertir en la capacidad de las personas y el trabajo decente y sostenible. Documento temático para la Reunión técnica sobre el futuro del trabajo en la industria automotriz*. https://www.ilo.org/wcmsp5/groups/public/—ed_dialogue/—sector/documents/meetingdocument/wcms_741663.pdf

Palacio, M. (2022). *Scrum Master Temario Troncal 1* (3rd ed.).

Perna, G., Trevisan, M., Giordano, D., & Drago, I. (2022). A first look at HTTP/3 adoption and performance. *Computer Communications*, *187*, 115–124. https://doi.org/10.1016/j.comcom.2022.02.005

Perry, L. (2020). *IoT and Edge Computing for Architects: Implementing Edge and IoT Systems From Sensors to Clouds with Communication Systems, Analytics, and Security, 2nd Edition* (2nd ed.). Packt Publishing Ltd.

Rahman, W. (2020). *AI and Machine Learning* (1st ed.). SAGE Publications.

Raspberry Pi. (2023). *Raspberry Pi Pico Datasheet*.

Richard, W. W. (2015). *Parking Management for Smart Growth*. Island Press.

Richards, M. (2022). *Software Architecture Patterns* (2nd ed.). O'Reilly Media.

Sadalage, P. J., & Fowler, M. (2013). *NoSQL Distilled*. Pearson Education.

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3), 160. https://doi.org/10.1007/s42979-021-00592-x

Sayarshad, H. (2023). Designing intelligent public parking locations for autonomous vehicles. *Expert Systems with Applications*, *222*, 119810. https://doi.org/10.1016/j.eswa.2023.119810

Schwaber, K., & Sutherland, J. (2020). *La Guía Definitiva de Scrum: Las Reglas del Juego*.

scikit-learn developers. (2023a). *Decision Trees*. https://scikit-learn.org/stable/modules/tree.html

scikit-learn developers. (2023b). *Linear Models*. https://scikit-learn.org/stable/modules/linear_model.html

Secretaría Nacional de Planificación. (2021). *Plan de Creación de Oportunidades 2021-2025 Observatorio Regional de Planificación para el Desarrollo de América Latina y el Caribe*. https://observatorioplanificacion.cepal.org/sites/default/files/plan/files/Plan-de-Creaci%C3%B3n-de-Oportunidades-2021-2025-Aprobado_compressed.pdf

Singh, R., Gehlot, A., Gupta, L. R., Singh, B., & Tyagi, P. (2019). *Getting Started for Internet of Things with Launch Pad and ESP8266*. River Publishers.

Susmita, R. (2019). A Quick Review of Machine Learning Algorithms. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 35–39. https://doi.org/10.1109/COMITCon.2019.8862451

Tang, R. (2023). Treatment Strategy for Parking Problems in Large Cities. *Highlights in Science, Engineering and Technology*, *37*, 157–165. https://doi.org/10.54097/hset.v37i.6070

The CentOS Project. (2023). *CentOS Documentation Home*. https://docs.centos.org/en-US/docs/

The PostgreSQL Global Development Group. (2023). *Documentation*. https://www.postgresql.org/docs/

Tonato, C. C., & Sinche, M. S. (2022). Análisis comparativo entre arquitecturas de sistemas IoT. *Revista de Investigación En Tecnologías de La Información*, *10*(21), 55–70. https://doi.org/10.36825/RITI.10.21.006

Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, *19*(1), 281. https://doi.org/10.1186/s12911-019-1004-8

Valeo, J. P., Gregoracci, S., Seijas, L. M., & Etcheverry, J. A. (2020). Sistema Automatizado de Estacionamiento para Patentes Argentinas. *Elektron*, *4*(2), 107–113. https://doi.org/10.37537/rev.elektron.4.2.112.2020

Vercel. (2023). *Next.js*. https://nextjs.org/docs

Vidal-Silva, C. L., Sánchez-Ortiz, A., Serrano, J., & Rubio, J. M. (2021). Experiencia académica en desarrollo rápido de sistemas de información web con Python y Django. *Formación Universitaria*, *14*(5), 85–94. https://doi.org/10.4067/S0718-50062021000500085

Westcott, S., & Westcott, J. (2020). *Basic Electronics: Theory and Practice* (3rd ed.). Mercury Learning and Information.

You, E. (2023). *Vue*. https://vuejs.org/

8. ANEXOS

Anexo I. Carta de asignación, registro de recursos y cronograma

Figura 109. Carta de Asignación

Pontificia Universidad Católica del Ecuador | SANTO DOMINGO
Seréis mis testigos

DIRECCIÓN DE INVESTIGACIÓN Y POSTGRADOS

Santo Domingo, 24 de octubre de 2023

Señor
Arturo Miñaca López
Gerente del Comercial Miñaca
Presente. -

Reciba un atento saludo de quienes formamos parte de la Pontificia Universidad Católica del Ecuador, Sede Santo Domingo.

Por medio del presente informo a usted que los estudiantes Carrascal Cedeño Byron David, con C.C: 1758605925 y Cruz Zambrano Patricio Ivan, con C.C: 1350176101, son estudiantes del octavo nivel, asignatura Integracion Curricular, Escuela de Sistemas. Actualmente se encuentra elaborando su Trabajo de Titulación denominado: **SISTEMA IOT CON MACHINE LEARNING PARA EL CONTROL DE ESTACIONAMIENTO VEHICULAR EN EL COMERCIAL MIÑACA DEL CANTÓN SANTO DOMINGO**; por tal motivo, solicito muy comedidamente, se le permita desarrollar en su Institución las siguientes actividades:

ACTIVIDADES:

- Solicitar información relacionada al estudio a los departamentos respectivos.
- Aplicación de la encuesta a los clientes del parqueadero.
- Aplicación de la entrevista al gerente del local comercial.
- Presentación de los avances de las funcionalidades del producto (productos mínimos viables).

El estudiante deberá mantener la disciplina durante su estancia en la Institución, así como acogerse a las normativas necesarias para su desempeño.

Es necesario aclarar que los resultados de la investigación serán difundidos, mismos que previamente deberán ser comunicados a su persona a partir de lo cual le emitirá una carta que avale el impacto generado.

De antemano le agradezco su amable atención y ayuda en formar profesionales idóneos para servir a la sociedad.

Cordialmente,

PUCE SANTO DOMINGO
PhD. Vallejo Alarcón Jaime
DIRECTOR DE INVESTIGACIÓN Y POSGRADOS PUCE SD
C.I.: 0959775016
dipostgrados@pucesd.edu.ec
(02)3702868 Ext. 204

Dirección: Vía a Chone Km. 2.
Código postal: 230203 / Teléfono: (593-0993283425)
Santo Domingo - Ecuador / www.pucesd.edu.ec

Figura 110. Registro de Recursos

Recursos	Cantidad	Valor unitario	Valor total USD
GASTOS			
Humano			
Estudiantes	2	$ -	$ -
Tecnologia			
Computadora	1	$ 1.400,00	$ 1.400,00
Impresora	1	$ 350,00	$ 350,00
Servidor	1	$ 500,00	$ 500,00
Punto de Acceso WiFi	1	$ 75,00	$ 75,00
Switch 8P	1	$ 10,00	$ 10,00
Reguladro de Voltaje	1	$ 38,00	$ 38,00
Regleta Electrica 6T	1	$ 10,00	$ 10,00
Sensor Ultrasonico	6	$ 2,63	$ 15,78
ESP8266 MODULO WIFI	4	$ 8,83	$ 35,32
BUZZER 5 VDC	4	$ 0,44	$ 1,76
2N2222 TRANSISTOR	8	$ 0,08	$ 0,64
ARDUINO FUENTE	4	$ 2,99	$ 11,96
ADAPTADOR DC12V 2A	4	$ 4,46	$ 17,84
BALUM DC HEMBRA	4	$ 0,53	$ 2,12
BALUM DC MACHO	4	$ 0,53	$ 2,12
Materiales			
SELLADO	1	$ 3,92	$ 3,92
CAUTIN	1	$5,27	$ 5,27
TAPON M	4	$ 1,51	$ 6,04
TAPON H	4	$ 1,33	$ 5,32
Cable UTP	20	$ 1,30	$ 26,00
Servicios			
Transporte	2	$ 5,35	$ 10,70
DigitalOcean (Frontend)	2	$ 5,00	$ 10,00
DigitalOcean (Backend)	2	$ 5,00	$ 10,00
DigitalOcean (Base de Datos)	2	$ 15,00	$ 30,00
TOTAL:			$ 2.577,79
INGRESOS			
Fuente de Ingresos			
Recursos Propios			$ 300,00
TOTAL:			$ 300,00

Figura 111. Cronograma de Implementación

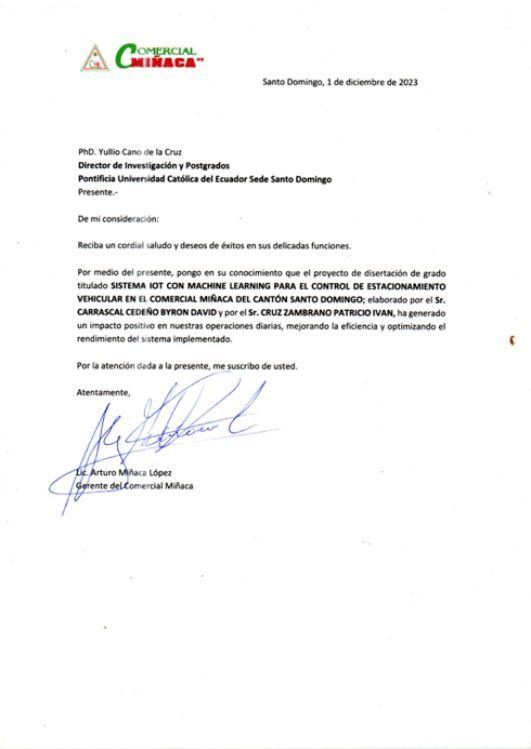
Radiador de información del Trabajo de Titulación de Grado (Trazabilidad entre PMV y las épicas)

N°	Actividades	Estado	2023										2024		
			Producto Mínimo Viable I = 7° nivel						Producto Mínimo Viable II = 8° nivel						
			MESES												
			Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	Enero	Febrero	Marzo
ÉPICA 1	Desarrollo de la propuesta	HECHO													
ÉPICA 2	Ejecución del Proyecto	HECHO													
ÉPICA 3	Revisión literaria	HECHO													
ÉPICA 4	Metodología de la investigación	HECHO													
ÉPICA 5	Resultados	HECHO													
ÉPICA 5.1	Del primer objetivo específico	HECHO													
ÉPICA 5.2	Del segundo objetivo específico	HECHO													
ÉPICA 5.3	Del tercer objetivo específico	HECHO													
ÉPICA 6	Desarrollo de la propuesta de intervención (Gestión adaptativa con incremento iterativo "Scrum")	HECHO													
ÉPICA 6.1	Prototipo de diseño operativo, llamado solución en punta	HECHO													
ÉPICA 6.2	Sprint 1	HECHO													
ÉPICA 6.3	Sprint 2	HECHO													
ÉPICA 6.4	Sprint 3	HECHO													
ÉPICA 7	Conclusiones y Recomendaciones	HECHO													
ÉPICA 8	Informe Final del Trabajo de Titulación de Grado	HECHO													
ÉPICA 8.1	Correcciones de Lectores	HECHO													
ÉPICA 8.2	Anexos	HECHO													
ÉPICA 9	Disertación de Grado	HECHO													

Nota: PMV=Producto mínimo viable; = Relaciona la semana con "Done"

Anexo II. Carta de impacto, consentimiento informado

Figura 112. Carta de Impacto



COMERCIAL MIÑACA

Santo Domingo, 1 de diciembre de 2023

PhD. Yullio Cano de la Cruz
Director de Investigación y Postgrados
Pontificia Universidad Católica del Ecuador Sede Santo Domingo
Presente.-

De mi consideración:

Reciba un cordial saludo y deseos de éxitos en sus delicadas funciones.

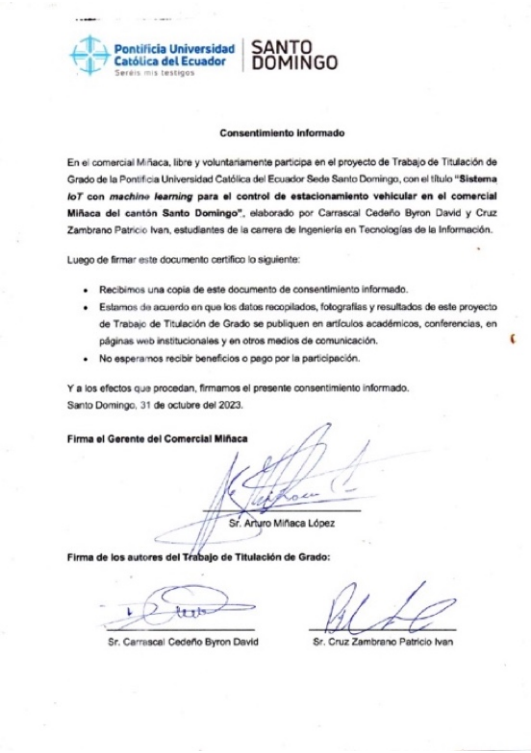
Por medio del presente, pongo en su conocimiento que el proyecto de disertación de grado titulado **SISTEMA IOT CON MACHINE LEARNING PARA EL CONTROL DE ESTACIONAMIENTO VEHICULAR EN EL COMERCIAL MIÑACA DEL CANTÓN SANTO DOMINGO;** elaborado por el **Sr. CARRASCAL CEDEÑO BYRON DAVID** y por el **Sr. CRUZ ZAMBRANO PATRICIO IVAN,** ha generado un impacto positivo en nuestras operaciones diarias, mejorando la eficiencia y optimizando el rendimiento del sistema implementado.

Por la atención dada a la presente, me suscribo de usted.

Atentamente,

Lic. Arturo Miñaca López
Gerente del Comercial Miñaca

Figura 113. Consentimiento Informado



Pontificia Universidad Católica del Ecuador
Seréis mis testigos
SANTO DOMINGO

Consentimiento Informado

En el comercial Miñaca, libre y voluntariamente participa en el proyecto de Trabajo de Titulación de Grado de la Pontificia Universidad Católica del Ecuador Sede Santo Domingo, con el título **"Sistema *IoT* con *machine learning* para el control de estacionamiento vehicular en el comercial Miñaca del cantón Santo Domingo"**, elaborado por Carrascal Cedeño Byron David y Cruz Zambrano Patricio Ivan, estudiantes de la carrera de Ingeniería en Tecnologías de la Información.

Luego de firmar este documento certifico lo siguiente:

- Recibimos una copia de este documento de consentimiento informado.
- Estamos de acuerdo en que los datos recopilados, fotografías y resultados de este proyecto de Trabajo de Titulación de Grado se publiquen en artículos académicos, conferencias, en páginas web institucionales y en otros medios de comunicación.
- No esperamos recibir beneficios o pago por la participación.

Y a los efectos que procedan, firmamos el presente consentimiento informado.
Santo Domingo, 31 de octubre del 2023.

Firma el Gerente del Comercial Miñaca

Sr. Arturo Miñaca López

Firma de los autores del Trabajo de Titulación de Grado:

Sr. Carrascal Cedeño Byron David

Sr. Cruz Zambrano Patricio Ivan

Anexo III. Validación de instrumentos de recolección de datos

PUCE

Santo Domingo, 16 de noviembre del 2023

Estimado Ph.D. Carrión Bósquez Nelson Geovany
De mi consideración:

El motivo del presente documento es que lo hemos elegido para redactar la solicitud de revisión y validación de los instrumentos de recolección de datos.

A continuación, encontrará la entrevista y encuesta que incluyen las interrogantes necesarias para recopilar los datos pertinentes para la elaboración del trabajo de titulación **"SISTEMA IOT CON MACHINE LEARNING PARA EL CONTROL DE ESTACIONAMIENTO VEHICULAR EN EL COMERCIAL MIÑACA DEL CANTÓN SANTO DOMINGO."**; dirigida a los clientes y a los empleados del Comercial Miñaca.

Para la validación de los instrumentos se adjunta la operacionalización de variables, con la finalidad de que se visualice la relación de las preguntas con las categorías e indicadores. Además, se encuentran divididos los instrumentos en dos partes, la primera corresponde a la entrevista (preguntas de fondo verde para las dos variables) y la segunda a la encuesta (preguntas de la variable independiente de fondo turquesa y las preguntas de la variable dependiente fondo gris).

Gracias por su valiosa colaboración en este trabajo de titulación de grado.
Atentamente,

Patricio Ivan Cruz Zambrano
picruzz@pucesd.edu.ec

Byron David Carrascal Cedeño
bdcarrascalc@pucesd.edu.ec

1

PUCE

Operacionalización de las variables
Operacionalización de la variable independiente

Tabla 1
Operacionalización variable independiente Internet de las Cosas.

Conceptualizaciones	Categorías	Indicadores	Preguntas	Herramienta
El internet de las cosas se considera un paradigma que equipa a los objetos con sensores, actuadores y procesadores, que permiten comunicarse entre sí para ejecutar funciones y cumplir un propósito significativo. Las tecnologías de IoT han logrado con el tiempo reducir significativamente el esfuerzo humano y mejorar la calidad de vida en muchas actividades. Los dispositivos IoT están conectados a una red que transmite datos y agrega la capacidad de realizar tareas complejas, con varias tecnologías que trabajan juntas, que a su vez requieren un alto grado de inteligencia. El internet de las cosas está implementado en diferentes áreas como lo es la educación, entretenimiento, vida social, conservación de la energía, domótica y sistemas de transporte (Gacovski, 2019, pp. 91–93).	Procesamiento	Cloud Computing	¿Qué tipo de teléfono móvil posee actualmente?	Encuesta a los Clientes
		Fog Computing	¿Con qué frecuencia utiliza el navegador web de su teléfono móvil?	Encuesta a los Clientes
		Edge Computing	¿Cuáles son las dificultades en la congestión o insuficiencia de plazas de estacionamiento vehicular, cuando la demanda es alta?	Entrevista al Gerente
	Protocolos	MQTT	¿Con qué frecuencia utiliza estacionamientos vehiculares inteligentes?	Encuesta a los Clientes
		CoAP, AMQP	¿Qué tan importante es para usted la facilidad de encontrar un estacionamiento vehicular?	Encuesta a los Clientes
		HTTP/REST	¿Cuál es su opinión acerca de las plataformas en línea que aportan valor a las decisiones de los locales comerciales?	Entrevista al Gerente
	Dispositivos Físicos	Sensores	¿Con qué frecuencia utiliza aplicaciones web para visualizar la disponibilidad de estacionamientos vehiculares antes de llegar a su destino?	Encuesta a los Clientes
		Actuadores	¿Cómo ayudaría el sistema de registro de información vehicular a mejorar la gestión del estacionamiento, para que sea más eficiente y beneficiosa?	Entrevista al Gerente
		Microcontroladores	¿Cuál es su perspectiva acerca del beneficioso de recopilar información de los clientes mediante el uso de estacionamientos inteligentes?	Entrevista al Gerente
	Tecnologías de Acceso	WPAN, WLAN	¿Qué medio utiliza para conectarse a internet desde su teléfono móvil?	Encuesta a los Clientes
		WAN	¿Qué tan complicado es para usted tener acceso a Internet todos los días en su teléfono móvil?	Encuesta a los Clientes

1

PUCE

Tabla 2
Operacionalización variable independiente Machine Learning.

Conceptualizaciones	Categorías	Indicadores	Preguntas	Herramienta
El machine learning (ML) se considera un subconjunto de la IA, que consiste en entrenar algoritmos para que estos hagan predicciones o tomen decisiones basadas en los datos, para comprender el ML, se requiere conocimientos técnicos previos en áreas como las matemáticas, estadística e informática. En el machine learning los datos son cruciales, porque los algoritmos aprenden de ejemplos y patrones en los datos. Se considera que la calidad y la cantidad de datos, pueden influir significativamente en el rendimiento de un modelo de ML (Was, 2020, pp. 38–57).	Tipos de Aprendizaje	Aprendizaje Supervisado, Aprendizaje No Supervisado	¿Cómo analiza el desempeño y la capacidad actual de su parqueadero, en términos de cumplir con las necesidades de los clientes?	Entrevista al Gerente
		Aprendizaje Semi-supervisado, Aprendizaje por Refuerzo	¿Cuál es la tendencia en términos de ocupación durante ciertas horas del día o días de la semana que aumenten la cantidad de vehículos?	Entrevista al Gerente
	Algoritmos	Regresión Lineal, Regresión Logística	¿Cómo se evalúa en la actualidad la eficiencia del estacionamiento vehicular en su local comercial, en términos de la gestión de plazas disponibles?	Encuesta a los Clientes
		Arboles de Decisión, KNN, Support Vector Machine, Gradient Boosting, Redes Neuronales	¿Qué relevancia tiene para usted incorporar sensores para recopilar datos, que impulsen mejoras en la gestión del estacionamiento vehicular?	Entrevista al Gerente

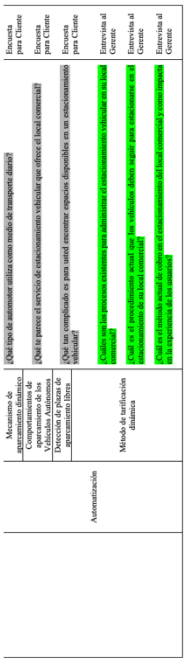
2

PUCE

Operacionalización de la variable dependiente
Tabla 3
Operacionalización variable dependiente Control de Estacionamiento Vehicular.

Conceptualizaciones	Categorías	Indicadores	Preguntas	Herramienta
El incremento del número de automóviles y la demanda de espacios de estacionamiento, han generado problemas con la congestión del tráfico y la seguridad vial, existen distintas soluciones para abordar este tema, por esta razón el implementar sistemas de gestión inteligente, consigue traer numeroso beneficios, entre ellos, la reducción del tiempo de búsqueda y la disminución de consumo de combustible, estas soluciones traen consigo un nivel de satisfacción para el cliente, ya que, brinda comodidad e información en tiempo real sobre la disponibilidad de los estacionamientos (A. Elsonbaty & Shams, 2020, pp. 55–56).	Sistemas Inteligentes Centralizado	Sistema de información y guiado de aparcamientos (PGIS)	¿Con qué frecuencia ha experimentado problemas para encontrar estacionamiento vehicular en un local comercial?	Encuesta a los Clientes
			¿Qué tan importante es para usted una aplicación que informe el estado del estacionamiento vehicular para reducir los tiempos de espera?	Encuesta a los Clientes
		Sistema de información sobre la ocupación de aparcamientos (COINS)	¿Está de acuerdo con el método actual para encontrar estacionamiento vehicular accesible para personas con discapacidad?	Encuesta a los Clientes
		Sistema de guiado basado en agentes (ABGS)	¿Considera que un sistema de estacionamiento vehicular inteligente podría contribuir a la identificación más eficiente de plazas accesibles para personas con discapacidad?	Entrevista al Gerente
			¿Cuáles son las medidas adoptadas para disponer de plazas de estacionamiento accesibles para personas con discapacidad?	Entrevista a los clientes
	Estrategias de Gestión	Cantidad de Estacionamientos Utilizados	¿Estás de acuerdo en visualizar los espacios disponibles del estacionamiento vehicular desde tu teléfono móvil?	Encuesta a los Clientes
		Eficiencia en el Uso del Espacio	¿Cuál sería el beneficio de utilizar la información del estacionamiento inteligente, para analizarla con inteligencia artificial y así obtener pronósticos?	Entrevista al Gerente
	Estrategias de Oferta	Diferenciación entre aparcamiento y construcción	¿Qué tan importante es para usted un sistema que le permita ver la disponibilidad de espacios en un estacionamiento vehicular antes de llegar a su destino?	Encuesta a los Clientes
		Aprovechamiento de los recursos de aparcamiento	¿Qué impacto tendría el despliegue de un sistema de estacionamiento vehicular, para reducir la congestión y maximizar el uso de plazas disponibles?	Entrevista al Gerente
			¿Cuál es la capacidad diaria del estacionamiento del local comercial, en términos de cantidad de vehículos que puede albergar?	Entrevista al Gerente

3



Preguntas
Objetivo:
Identificar las necesidades que tiene el control de estacionamiento vehicular para la obtención de los requerimientos.

Baremo:
Claridad: Se refiere si la pregunta está comprendida por los destinatarios.

Pertinencia: Se refiere si la pregunta corresponde con lo que se quiere indagar

Las preguntas en cuanto a su claridad y pertinencia se encuentran bajo la escala valorativa Likert del 1 al 5 (donde 1 es el menor valor y 5 el mayor). Podrá añadir una formulación alternativa y observación, en caso que considere necesario

Marque con una cruz (X) el tramo del baremo que exprese mejor su juicio "Claridad" y "Pertinencia" sobre los ítems propuestos:

Entrevista dirigida gerente de la empresa

Variable independiente: Machine Learning (**Categoría**: Tipos de Aprendizaje)

1. ¿Cómo analiza el desempeño y la capacidad actual de su parqueadero, en términos de cumplir con las necesidades de los clientes?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Machine Learning (**Categoría**: Algoritmos)

2. ¿Cómo se evalúa en la actualidad la eficiencia del estacionamiento vehicular en su local comercial, en términos de la gestión de plazas disponibles?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría:** Estrategias de Oferta)

3. ¿Cuál es la capacidad diaria del estacionamiento del local comercial, en términos de cantidad de vehículos que puede albergar?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría:** Automatización)

4. ¿Cuáles son los procesos existentes para administrar el estacionamiento vehicular en su local comercial?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría:** Automatización)

5. ¿Cuál es el procedimiento actual que los vehículos deben seguir para estacionarse en el estacionamiento de su local comercial?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría:** Automatización)

6. ¿Cuál es el método actual de cobro en el estacionamiento del local comercial y como impacta en la experiencia de los usuarios?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Machine Learning (**Categoría**: Tipos de Aprendizaje)

7. ¿Cuál es la tendencia en términos de ocupación durante ciertas horas del día o días de la semana que aumenten la cantidad de vehículos?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría:** Estrategias de Oferta)

8. ¿Cuáles son las dificultades en la congestión o insuficiencia de plazas de estacionamiento vehicular, cuando la demanda es alta?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría:** Sistemas Inteligentes Centralizado)

9. ¿Cuáles son las medidas adoptadas para disponer de plazas de estacionamiento accesibles para personas con discapacidad?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Internet de las Cosas (**Categoría:** Protocolos)

10. ¿Cómo ayudaría el sistema de registro de información vehicular a mejorar la gestión del estacionamiento, para que sea más eficiente y beneficiosa?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

PUCE

Variable independiente: Internet de las Cosas (**Categoría:** Dispositivos Físicos)

11. ¿Qué relevancia tiene para usted incorporar sensores para recopilar datos, que impulsen mejoras en la gestión del estacionamiento vehicular?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría:** Estrategias de Gestión)

12. ¿Qué impacto tendría el despliegue de un sistema de estacionamiento vehicular, para reducir la congestión y maximizar el uso de plazas disponibles?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría:** Sistemas Inteligentes Centralizado)

13. ¿Considera que un sistema de estacionamiento vehicular inteligente podría contribuir a la identificación más eficiente de plazas accesibles para personas con discapacidad?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Internet de las Cosas (**Categoría**: Dispositivos Físicos)

14. ¿Cuál es su perspectiva acerca del beneficioso de recopilar información de los clientes mediante el uso de estacionamientos inteligentes?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

PUCE

Variable independiente: Internet de las Cosas (**Categoría:** Procesamiento)

15. ¿Cuál es su opinión acerca de las plataformas en línea que aportan valor a las decisiones de los locales comerciales?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Machine Learning (**Categoría**: Algoritmos)

16. ¿Cuál sería el beneficio de utilizar la información del estacionamiento inteligente, para analizarla con inteligencia artificial y así obtener pronósticos?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

PUCE

Encuesta dirigida a los clientes

Tema del Trabajo de Titulación de grado: Aplicación móvil con realidad aumentada para el fortalecimiento del proceso de promoción de productos en la Empresa de Calzado Almacén Mónica del Cantón Santo Domingo

Objetivo: Recolectar información para validar la propuesta de intervención enfocada al fortalecimiento del proceso de promoción de productos de la línea de calzado.

Instrucciones al público objetivo: La encuesta está dirigida a los clientes de la Empresa de Calzado Almacén Mónica, en base a la información obtenida, permitirá conocer el manejo de nuevas tendencias tecnológicas en el manejo de procesos de promoción.

PREGUNTAS

Variable independiente: Internet de las Cosas (**Categoría**: Procesamiento)

1. ¿Qué tipo de teléfono móvil posee actualmente?
   A. Teléfono Inteligente
   B. Teléfono Básico
   C. Teléfono Satelital
   D. Ninguno

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Internet de las Cosas (**Categoría**: Tecnologías de Acceso)

2. ¿Qué medio utiliza para conectarse a internet desde su teléfono móvil?
   A. Internet Prepago (Recargas)
   B. Internet Pospago (Plan)
   C. Internet mediante WiFi
   D. Ninguno

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Internet de las Cosas (**Categoría**: Procesamiento)

3. ¿Con qué frecuencia utiliza el navegador web de su teléfono móvil?
   A. Muy Frecuentemente
   B. Frecuentemente
   C. Ocasionalmente
   D. Raramente
   E. Nunca

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

PUCE

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría**: Estrategias de Gestión)

4. ¿Estás de acuerdo en visualizar los espacios disponibles del estacionamiento vehicular desde tu teléfono móvil?
   A. Totalmente de acuerdo
   B. De acuerdo
   C. Neutral
   D. Poco de acuerdo
   E. Nada de acuerdo

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Internet de las Cosas (**Categoría**: Tecnologías de Acceso)

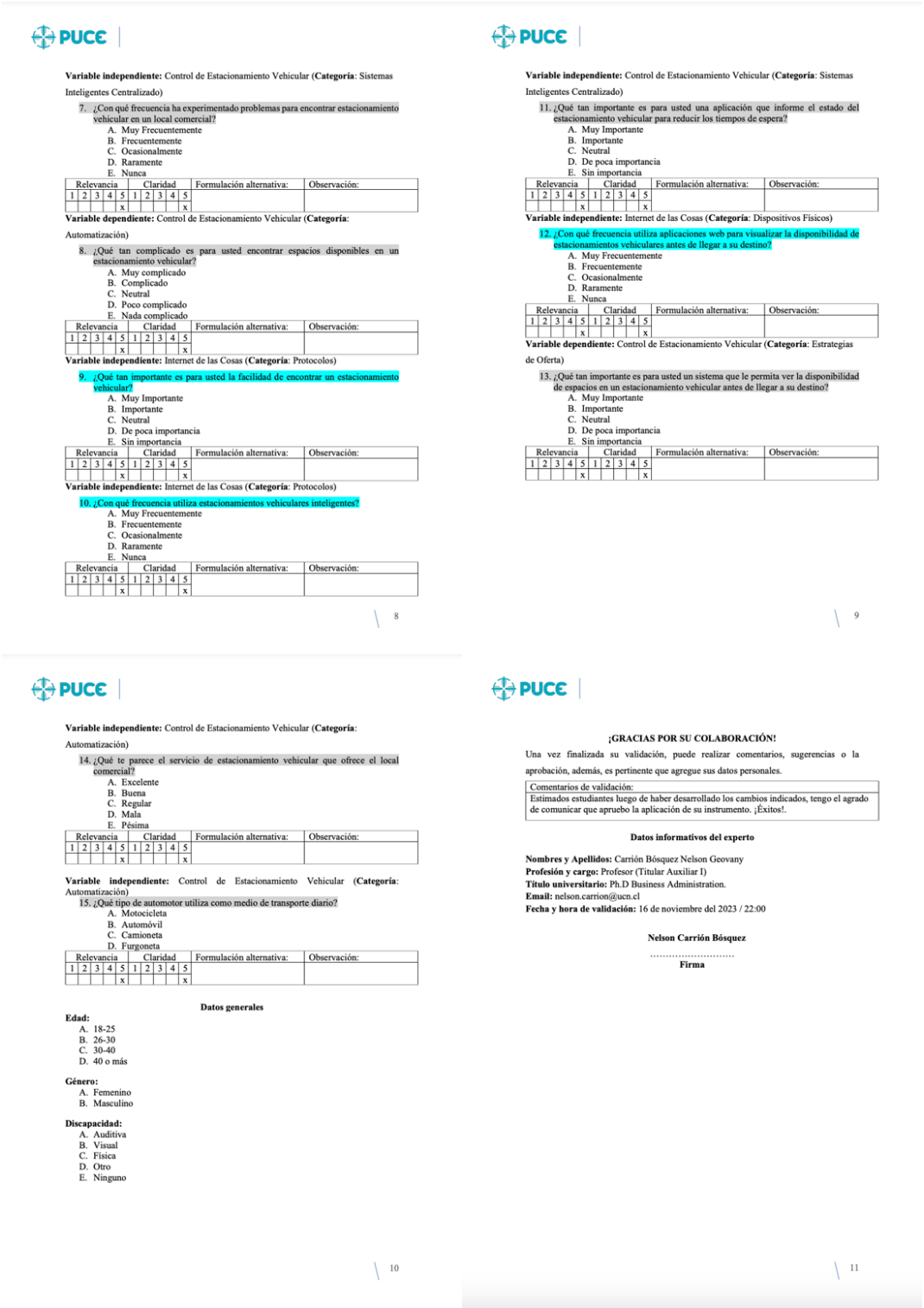
5. ¿Qué tan complicado es para usted tener acceso a Internet todos los días en su teléfono móvil?
   A. Muy complicado
   B. Complicado
   C. Neutral
   D. Poco complicado
   E. Nada complicado

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría**: Sistemas Inteligentes Centralizado)

6. ¿Está de acuerdo con el método actual para encontrar estacionamiento vehicular accesible para personas con discapacidad?
   A. Totalmente de acuerdo
   B. De acuerdo
   C. Neutral
   D. Poco de acuerdo
   E. Nada de acuerdo

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		



PUCE

Variable independiente: Control de Estacionamiento Vehicular (**Categoría**: Sistemas Inteligentes Centralizado)

7. ¿Con qué frecuencia ha experimentado problemas para encontrar estacionamiento vehicular en un local comercial?
   A. Muy Frecuentemente
   B. Frecuentemente
   C. Ocasionalmente
   D. Raramente
   E. Nunca

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría**: Automatización)

8. ¿Qué tan complicado es para usted encontrar espacios disponibles en un estacionamiento vehicular?
   A. Muy complicado
   B. Complicado
   C. Neutral
   D. Poco complicado
   E. Nada complicado

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Internet de las Cosas (**Categoría**: Protocolos)

9. ¿Qué tan importante es para usted la facilidad de encontrar un estacionamiento vehicular?
   A. Muy Importante
   B. Importante
   C. Neutral
   D. De poca importancia
   E. Sin importancia

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Internet de las Cosas (**Categoría**: Protocolos)

10. ¿Con qué frecuencia utiliza estacionamientos vehiculares inteligentes?
    A. Muy Frecuentemente
    B. Frecuentemente
    C. Ocasionalmente
    D. Raramente
    E. Nunca

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

8

PUCE

Variable independiente: Control de Estacionamiento Vehicular (**Categoría**: Sistemas Inteligentes Centralizado)

11. ¿Qué tan importante es para usted una aplicación que informe el estado del estacionamiento vehicular para reducir los tiempos de espera?
    A. Muy Importante
    B. Importante
    C. Neutral
    D. De poca importancia
    E. Sin importancia

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Internet de las Cosas (**Categoría**: Dispositivos Físicos)

12. ¿Con qué frecuencia utiliza aplicaciones web para visualizar la disponibilidad de estacionamientos vehiculares antes de llegar a su destino?
    A. Muy Frecuentemente
    B. Frecuentemente
    C. Ocasionalmente
    D. Raramente
    E. Nunca

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Control de Estacionamiento Vehicular (**Categoría**: Estrategias de Oferta)

13. ¿Qué tan importante es para usted un sistema que le permita ver la disponibilidad de espacios en un estacionamiento vehicular antes de llegar a su destino?
    A. Muy Importante
    B. Importante
    C. Neutral
    D. De poca importancia
    E. Sin importancia

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

9

PUCE

Variable independiente: Control de Estacionamiento Vehicular (**Categoría**: Automatización)

14. ¿Qué te parece el servicio de estacionamiento vehicular que ofrece el local comercial?
    A. Excelente
    B. Buena
    C. Regular
    D. Mala
    E. Pésima

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Control de Estacionamiento Vehicular (**Categoría**: Automatización)

15. ¿Qué tipo de automotor utiliza como medio de transporte diario?
    A. Motocicleta
    B. Automóvil
    C. Camioneta
    D. Furgoneta

Relevancia					Claridad					Formulación alternativa:	Observación:
1	2	3	4	5	1	2	3	4	5		
				x					x		

Datos generales

Edad:
A. 18-25
B. 26-30
C. 30-40
D. 40 o más

Género:
A. Femenino
B. Masculino

Discapacidad:
A. Auditiva
B. Visual
C. Física
D. Otro
E. Ninguno

10

PUCE

¡GRACIAS POR SU COLABORACIÓN!

Una vez finalizada su validación, puede realizar comentarios, sugerencias o la aprobación, además, es pertinente que agregue sus datos personales.

Comentarios de validación:
Estimados estudiantes luego de haber desarrollado los cambios indicados, tengo el agrado de comunicar que apruebo la aplicación de su instrumento. ¡Éxitos!.

Datos informativos del experto

Nombres y Apellidos: Carrión Bósquez Nelson Geovany
Profesión y cargo: Profesor (Titular Auxiliar I)
Título universitario: Ph.D Business Administration.
Email: nelson.carrion@ucn.cl
Fecha y hora de validación: 16 de noviembre del 2023 / 22:00

Nelson Carrión Bósquez
.....
Firma

11

Anexo IV: Expertos en Evaluación de Instrumentos

Expertos en Evaluación de Instrumentos para Recopilación de Datos

Nombres	***Titulo Academico***	***Area***
Carrión Bósquez Nelson Geovany	Ph.D Business Administration.	Administración de Negocios
Cordova Galvez Rodolfo Sirilo	Mg. Redes de Comunicaciones	Sistemas

Anexo V. Historias de Usuario

Historia de Usuario	
Número: 1	**Usuario:** Cliente
Nombre historia: Inicio de Sesión (*Login*)	
Prioridad en negocio: 100	**Riesgo en desarrollo:** ALTO
Puntos estimados: 3	**Sprint:** 1
Programador responsable: Patricio Cruz y Byron Carrascal	
Descripción:	
Como Cliente **Quiero** ingresar las credenciales del usuario **Para** tener acceso a la plataforma.	
Escenario de prueba:	
Dado el ingreso de las credenciales del usuario correctas **Cuando** pulse el botón "Iniciar Sesión" **Entonces** se permitirá el acceso a la plataforma de reservas.	
Dado el ingreso de las credenciales del usuario incorrectas **Cuando** pulse el botón de "Iniciar Sesión" **Entonces** se mostrará una alerta de usuario o contraseña incorrecta.	

Historia de Usuario	
Número: 2	**Usuario:** Cliente
Nombre historia: Registro de Cuentas	
Prioridad en negocio: 95	**Riesgo en desarrollo:** ALTO
Puntos estimados: 5	**Sprint:** 1
Programador responsable: Patricio Cruz y Byron Carrascal	
Descripción:	
Como Cliente **Quiero** registrar mis datos **Para** ser parte de los usuarios que utilizan la plataforma de consulta de estacionamientos.	
Escenario de prueba:	
Dado el ingresar de los datos completos **Cuando** pulse el botón "Registrarse" **Entonces** se confirma el registro de un nuevo usuario.	
Dado el ingreso de los datos incompletos **Cuando** pulse el botón "Registrarse" **Entonces** se mostrará una alerta de campos obligatorios incompletos	

Historia de Usuario	
Número: 3	**Usuario:** Cliente
Nombre historia: Generar Código QR	
Prioridad en negocio: 90	**Riesgo en desarrollo:** MEDIO
Puntos estimados: 3	**Sprint:** 1
Programador responsable: Patricio Cruz y Byron Carrascal	
Descripción:	
Como Cliente **Quiero** visualizar mi código QR **Para** identificarme con el sistema en la E/S del estacionamiento.	
Escenario de prueba:	
Dado la visualización del código QR **Cuando** pulse el botón "Generar código QR" **Entonces** se muestra en pantalla el código que identifica la cuenta.	

Historia de Usuario	
Número: 4	**Usuario:** Cliente
Nombre historia: Registro de Vehículos	
Prioridad en negocio: 85	**Riesgo en desarrollo:** ALTO
Puntos estimados: 5	**Sprint:** 1
Programador responsable: Patricio Cruz y Byron Carrascal	
Descripción:	
Como Cliente **Quiero** registrar los datos de mi vehículo **Para** poder ingresar al estacionamiento.	
Escenario de prueba:	
Dado el registro de los datos de mi vehículo **Cuando** pulse el botón "Agregar Vehículo" **Entonces** se confirma el ingreso de un vehículo en el perfil del estudiante.	
Dado el registro de los datos de mi vehículo incompletos y/o incorrectos **Cuando** pulse el botón "Agregar Vehículo" **Entonces** se notifica que existe un error.	

Historia de Usuario	
Número: 5	**Usuario:** Cliente
Nombre historia: Visualizar Estacionamientos	
Prioridad en negocio: 80	**Riesgo en desarrollo:** ALTO
Puntos estimados: 5	**Sprint:** 1
Programador responsable: Patricio Cruz y Byron Carrascal	
Descripción:	
Como Cliente **Quiero** visualizar los espacios actuales **Para** corroborar la disponibilidad del estacionamiento.	
Escenario de prueba:	
Dado la visualización de espacios disponibles **Cuando** el estado de la celda sea "Libre" Entonces se visualizará el espacio en color verde.	
Dado la visualización de espacios disponibles **Cuando** el estado de la celda sea "Ocupado" **Entonces** se visualizará el espacio en color rojo.	

Historia de Usuario	
Número: 6	**Usuario:** Administrador
Nombre historia: Pronóstico de Clientes	
Prioridad en negocio: 75	**Riesgo en desarrollo:** ALTO
Puntos estimados: 13	**Sprint:** 2
Programador responsable: Patricio Cruz y Byron Carrascal	
Descripción:	
Como Administrador **Quiero** visualizar la cantidad posible de clientes que puedan llegar al comercial **Para** tomar de decisiones de negocio.	
Escenario de prueba:	
Dada la visualización de posibles clientes que puedan llegar al comercial **Cuando se** presione el botón "Generar Pronóstico" **Entonces** se presenta información que indica comportamiento de los clientes.	

Historia de Usuario	
Número: 7	**Usuario:** Administrador
Nombre historia: Reporte de E/S Vehicular	
Prioridad en negocio: 70	**Riesgo en desarrollo:** MEDIO
Puntos estimados: 2	**Sprint:** 2
Programador responsable: Patricio Cruz y Byron Carrascal	
Descripción:	
Como Administrador **Quiero** seleccionar la información de los vehículos **Para** mejorar el control y la administración del estacionamiento.	
Escenario de prueba:	
Dado la selección de la información de los vehículos **Cuando** se pulse el botón “Generar Reportes” **Entonces** se muestra información detallada de la entrada y salida de los vehículos.	
Dado la no selección de la información de los vehículos **Cuando** se pulse el botón “Generar Reportes” **Entonces** se muestra un mensaje de error.	

Historia de Usuario	
Número: 8	**Usuario:** Guardia
Nombre historia: Registro Automático de E/S Vehicular	
Prioridad en negocio: 65	**Riesgo en desarrollo:** ALTO
Puntos estimados: 8	**Sprint:** 3
Programador responsable: Patricio Cruz y Byron Carrascal	
Descripción:	
Como Guardia **Quiero** escanear el Codigo QR del cliente **Para** registrar la entrada o la salida del vehiculo.	
Escenario de prueba:	
Dado el escaneo del Codigo QR del cliente **Cuando** seleccione un vehiculo y pulse el botón “Registra Entrada” **Entonces** se muestra un mensaje de confirmacion que el registro de entrada fue exitoso.	
Dado el escaneo del Codigo QR del cliente **Cuando** se pulse el botón “Registra Salida” **Entonces** se muestra un mensaje de confirmacion que el registro de salida fue exitoso.	

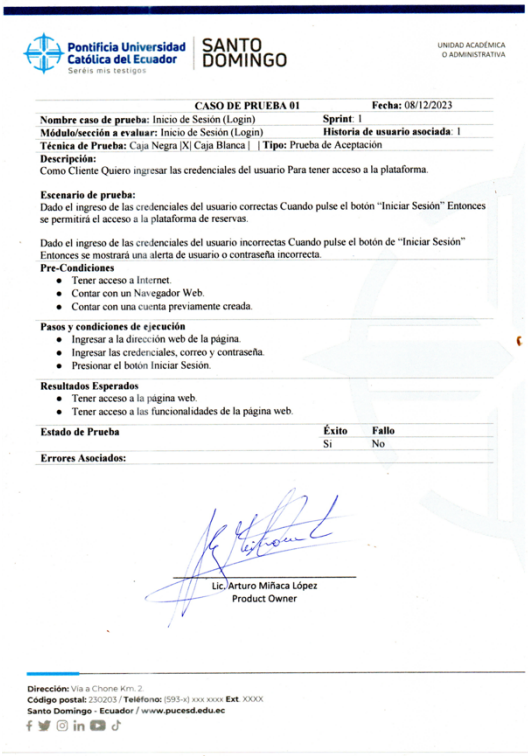
Historia de Usuario	
Número: 9	**Usuario:** Guardia
Nombre historia: Registro Manual de E/S Vehicular	
Prioridad en negocio: 60	**Riesgo en desarrollo:** ALTO
Puntos estimados: 8	**Sprint:** 3
Programador responsable: Patricio Cruz y Byron Carrascal	
Descripción:	
Como Guardia **Quiero** ingresar la placa **Para** registrar la entrada o la salida del vehiculo.	
Escenario de prueba:	
Dado el ingreso de la placa **Cuando** se pulse el botón “Verificar Placa” y el boton “Registrar Entrada” **Entonces** se muestra un mensaje de confirmacion que el registro de entrada fue exitoso.	
Dado el ingreso de la placa **Cuando** se pulse el botón “Verificar Placa” y el boton “Registrar Salida” **Entonces** se muestra un mensaje de confirmacion que el registro de salida fue exitoso.	

Anexo VI: Historias Técnicas

Historia Tecnica			
Título:	Interoperabilidad. Rendimiento - Detección Vehicular	**Estimación:**	8
Historias Relacionadas:	HU5 - Visualizar Estacionamientos		
Descripción:			
Implementar un sensor de proximidad ultrasónico HC-SR04 con un microcontrolador ESP8266, esta implementación permite determinar con precisión si está o no ocupado el estacionamiento vehicular, además esta información es utilizada para presentar el estado del estacionamiento en la aplicación web.			
Criterios de Aceptación:			
- La detección vehicular debe realizarse en tiempo real, con una latencia mínima entre la captura de datos y su procesamiento. - Los datos recopilados por los sensores se integran de manera efectiva con la aplicación web existente para que la información de detección esté disponible para los usuarios a través de la interfaz web.			
Beneficios de Justificación:			
Utilizar sensores para la detección vehicular permite de manera flexible obtener información en tiempo real del estado del estacionamiento, que es visualizado en la aplicación web.			
Impacto en el Producto o Proyecto:			
La capacidad de detectar el estado del estacionamiento en tiempo real, permite que la aplicación sea más dinámica y resulta en una mayor experiencia de usuario.			
Dependencias:			
- Sensores de proximidad ultrasónico HC-SR4 - Microcontroladores ESP8266 - Fuentes de alimentación de 12v. - Servidor con Ubuntu Server. - Programación en Python y JavaScript			
Tareas o Subtareas:			
- Diseñar las conexiones entre el sensor de proximidad y el microcontrolador. (ESP8266, HC-SR04) - Diseñar la fuente de energía del sensor y el microcontrolador. (ESP8266, PowerBank) - Falshear el microcontrolador con Micropython. (ESP8266, Micropython) - Programar el microcontrolador con el sensor crear archivo boot.py. (ESP8266, Micropython, HC-SR04) - Programar el microcontrolador como publicador crear archivo main.py. (ESP8266, Micropython, MQTT) - Configurar un servidor con un broker MQTT. (Ubuntu Server, Mosquitto, NodeRED) - Programar la aplicación NodeRED como suscriptor. (NodeRED, MQTT) - Realizar pruebas al funcionamiento del sensor. (Django REST Framework, ESP8266, MQTT)			

Historia Tecnica			
Título:	Operabilidad. Escalabilidad - Despliegue en Producción	**Estimación:**	5
Historias Relacionadas:	HU1 - Inicio de Sesión (*Login*), HU2 - Registro de Cuentas, HU3 - Generar Código *QR*, HU4 - Registro de Vehículos, HU5 - Visualizar Estacionamientos, HU6 - Pronóstico de Clientes, HU7 - Reportes de E/S Vehicular, HU8 - Registro Automático de E/S Vehicular, HU9 - Registro Manual de E/S Vehicular		
Descripción:			
Desplegar distintos servicios en la nube utilizando Digital Ocean como PaaS. La aplicación consta de un frontend en React, un backend en Django (Python) y una base de datos relacional en MySQL.			
Criterios de Aceptación:			
- Despliege sencillo en poco tiempo de los servicios necesarios para el funcionamiento correcto de la aplicacion. - Infraestructura escalable y segura con cerificados SSL incorporados en el subdominio entregado.			
Beneficios de Justificación:			
Este despliegue a producción con Digital Ocean proporciona una infraestructura escalable y segura para nuestra aplicación, garantizando un rendimiento óptimo, tolerancia a fallos y un proceso de implementación eficiente. La utilización de un PaaS como Digital Ocean simplifica la gestión de la infraestructura y permite un enfoque más centrado en el desarrollo y la mejora continua.			
Impacto en el Producto o Proyecto:			
"La capacidad de contar con un servicio escalable y seguro brinda flexibilidad para optimizar los recursos en la nube cuando la demanda del servicio es elevada, tanto por parte de la aplicación como de los sensores que realizan peticiones de manera continua.			
Dependencias:			
- Tarjeta de Credito para el pago mensual por los servicios. - Aplicacion de Node.js - Aplicacion de Python - Instancia de Base de Datos MySQL			
Tareas o Subtareas:			
- Registrar el medio de pago - Enlazar la cuenta de DigitalOcean con la cuenta de Github - Crear una aplicacion Node.js con el respositorio de frontend (GitHub) - Cambiar el script de build y start para que se ejecuten en produccion - Agregar variables de entorno a la aplicacion de Node.js - Crear una aplicacion Python con el repositorio de backend (GitHub) - Cambiar el script de start para que se ejecute con el servidor Gunicorn - Agregar variables de entorno a la aplicacion de Python - Crear una instancia de Base de Datos MySQL - Crear una base de datos en la configuracion de la instancia - Crear un usuario para conectarse de forma segura - Agregar la aplicacion python del backend y la base de datos al mismo grupo seguro de conexion.			

Anexo VII: Pruebas de Aceptación



Pontificia Universidad Católica del Ecuador
Seréis mis testigos
SANTO DOMINGO
UNIDAD ACADÉMICA O ADMINISTRATIVA

CASO DE PRUEBA 01	Fecha: 08/12/2023
Nombre caso de prueba: Inicio de Sesión (Login)	**Sprint**: 1
Módulo/sección a evaluar: Inicio de Sesión (Login)	**Historia de usuario asociada**: 1
Técnica de Prueba: Caja Negra [X] Caja Blanca [ ] **Tipo:** Prueba de Aceptación	

Descripción:
Como Cliente Quiero ingresar las credenciales del usuario Para tener acceso a la plataforma.

Escenario de prueba:
Dado el ingreso de las credenciales del usuario correctas Cuando pulse el botón "Iniciar Sesión" Entonces se permitirá el acceso a la plataforma de reservas.

Dado el ingreso de las credenciales del usuario incorrectas Cuando pulse el botón de "Iniciar Sesión" Entonces se mostrará una alerta de usuario o contraseña incorrecta.

Pre-Condiciones
- Tener acceso a Internet.
- Contar con un Navegador Web.
- Contar con una cuenta previamente creada.

Pasos y condiciones de ejecución
- Ingresar a la dirección web de la página.
- Ingresar las credenciales, correo y contraseña.
- Presionar el botón Iniciar Sesión.

Resultados Esperados
- Tener acceso a la página web.
- Tener acceso a las funcionalidades de la página web.

Estado de Prueba	Éxito	Fallo
	Si	No

Errores Asociados:

Lic. Arturo Miñaca López
Product Owner

Pontificia Universidad Católica del Ecuador
Seréis mis testigos
SANTO DOMINGO
UNIDAD ACADÉMICA O ADMINISTRATIVA

CASO DE PRUEBA 02	Fecha: 08/12/2023
Nombre caso de prueba: Registro de Cuentas	**Sprint**: 1
Módulo/sección a evaluar: Registro de Cuentas	**Historia de usuario asociada**: 2
Técnica de Prueba: Caja Negra [X] Caja Blanca [ ] **Tipo:** Prueba de Aceptación	

Descripción:
Como Cliente Quiero registrar mis datos Para ser parte de los usuarios que utilizan la plataforma de consulta de estacionamientos.

Escenario de prueba:
Dado el ingresar de los datos completos Cuando pulse el botón "Registrarse" Entonces se confirma el registro de un nuevo usuario.

Dado el ingreso de los datos incompletos Cuando pulse el botón "Registrarse" Entonces se mostrará una alerta de campos obligatorios incompletos

Pre-Condiciones
- Tener acceso a Internet.
- Contar con un Navegador Web.
- Contar con un correo electrónico.

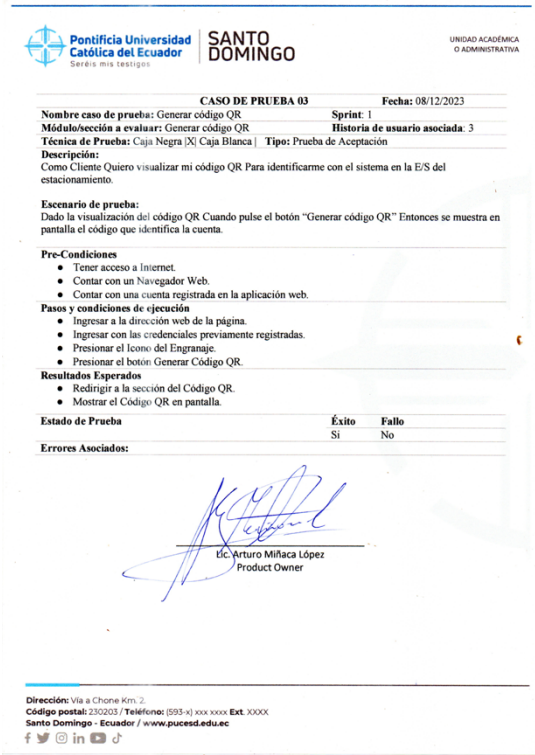
Pasos y condiciones de ejecución
- Ingresar a la dirección web de la página.
- Presionar el botón Registrarse.
- Ingresar los datos de los campos requeridos.
- Aceptar los Términos y Condiciones.
- Presionar el botón Registrarse.

Resultados Esperados
- Obtener una cuenta para acceder a la aplicación web.

Estado de Prueba	Éxito	Fallo
	Si	No

Errores Asociados:

Lic. Arturo Miñaca López
Product Owner



Pontificia Universidad Católica del Ecuador
Seréis mis testigos
SANTO DOMINGO
UNIDAD ACADÉMICA O ADMINISTRATIVA

CASO DE PRUEBA 03	Fecha: 08/12/2023
Nombre caso de prueba: Generar código QR	**Sprint**: 1
Módulo/sección a evaluar: Generar código QR	**Historia de usuario asociada**: 3
Técnica de Prueba: Caja Negra [X] Caja Blanca [ ] **Tipo:** Prueba de Aceptación	

Descripción:
Como Cliente Quiero visualizar mi código QR Para identificarme con el sistema en la E/S del estacionamiento.

Escenario de prueba:
Dado la visualización del código QR Cuando pulse el botón "Generar código QR" Entonces se muestra en pantalla el código que identifica la cuenta.

Pre-Condiciones
- Tener acceso a Internet.
- Contar con un Navegador Web.
- Contar con una cuenta registrada en la aplicación web.

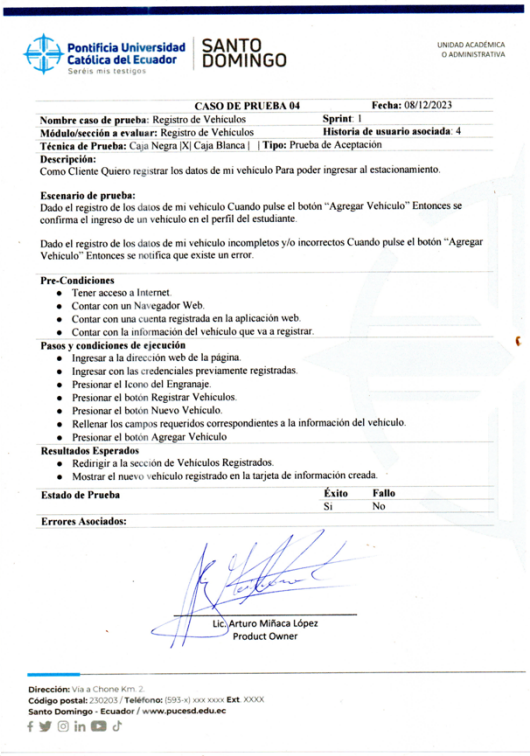
Pasos y condiciones de ejecución
- Ingresar a la dirección web de la página.
- Ingresar con las credenciales previamente registradas.
- Presionar el Icono del Engranaje.
- Presionar el botón Generar Código QR.

Resultados Esperados
- Redirigir a la sección del Código QR.
- Mostrar el Código QR en pantalla.

Estado de Prueba	Éxito	Fallo
	Si	No

Errores Asociados:

Lic. Arturo Miñaca López
Product Owner



Pontificia Universidad Católica del Ecuador
Seréis mis testigos
SANTO DOMINGO
UNIDAD ACADÉMICA O ADMINISTRATIVA

CASO DE PRUEBA 04	Fecha: 08/12/2023
Nombre caso de prueba: Registro de Vehículos	**Sprint**: 1
Módulo/sección a evaluar: Registro de Vehículos	**Historia de usuario asociada**: 4
Técnica de Prueba: Caja Negra [X] Caja Blanca [ ] **Tipo:** Prueba de Aceptación	

Descripción:
Como Cliente Quiero registrar los datos de mi vehículo Para poder ingresar al estacionamiento.

Escenario de prueba:
Dado el registro de los datos de mi vehículo Cuando pulse el botón "Agregar Vehículo" Entonces se confirma el ingreso de un vehículo en el perfil del estudiante.

Dado el registro de los datos de mi vehículo incompletos y/o incorrectos Cuando pulse el botón "Agregar Vehículo" Entonces se notifica que existe un error.

Pre-Condiciones
- Tener acceso a Internet.
- Contar con un Navegador Web.
- Contar con una cuenta registrada en la aplicación web.
- Contar con la información del vehículo que va a registrar.

Pasos y condiciones de ejecución
- Ingresar a la dirección web de la página.
- Ingresar con las credenciales previamente registradas.
- Presionar el Icono del Engranaje.
- Presionar el botón Registrar Vehículos.
- Presionar el botón Nuevo Vehículo.
- Rellenar los campos requeridos correspondientes a la información del vehículo.
- Presionar el botón Agregar Vehículo

Resultados Esperados
- Redirigir a la sección de Vehículos Registrados.
- Mostrar el nuevo vehículo registrado en la tarjeta de información creada.

Estado de Prueba	Éxito	Fallo
	Si	No

Errores Asociados:

Lic. Arturo Miñaca López
Product Owner

Pontificia Universidad Católica del Ecuador | SANTO DOMINGO
Seréis mis testigos
UNIDAD ACADÉMICA O ADMINISTRATIVA

CASO DE PRUEBA 05 **Fecha:** 08/12/2023

Nombre caso de prueba: Visualizar Estacionamientos **Sprint**: 1
Módulo/sección a evaluar: Visualizar Estacionamientos **Historia de usuario asociada**: 5
Técnica de Prueba: Caja Negra [X] Caja Blanca | **Tipo:** Prueba de Aceptación
Descripción:
Como Cliente Quiero visualizar los espacios disponibles Para corroborar la disponibilidad del estacionamiento.

Escenario de prueba:
Dado la visualización de espacios disponibles Cuando el estado de la celda sea "Libre" Entonces se visualizará el espacio en color verde.

Dado la visualización de espacios disponibles Cuando el estado de la celda sea "Ocupado" Entonces se visualizará el espacio en color rojo.

Pre-Condiciones
- Tener acceso a Internet.
- Contar con un Navegador Web.
- Contar con una cuenta registrada en la aplicación web.

Pasos y condiciones de ejecución
- Ingresar a la dirección web de la página.
- Ingresar con las credenciales previamente registradas.
- Presionar el Icono de 3 rayas.
- Presionar el botón Parqueadero.

Resultados Esperados
- Redirigir a la sección de Estacionamientos Disponibles.
- Mostrar las tarjetas de Información de cada Estacionamiento.

Estado de Prueba	Éxito	Fallo
	Si	No

Errores Asociados:

Lic. Arturo Miñaca López
Product Owner

Pontificia Universidad Católica del Ecuador | SANTO DOMINGO
Seréis mis testigos
UNIDAD ACADÉMICA O ADMINISTRATIVA

CASO DE PRUEBA 06 **Fecha:** 08/12/2023

Nombre caso de prueba: Pronóstico de Clientes **Sprint**: 2
Módulo/sección a evaluar: Pronóstico de Clientes **Historia de usuario asociada**: 6
Técnica de Prueba: Caja Negra [X] Caja Blanca | **Tipo:** Prueba de Aceptación
Descripción:
Como Administrador Quiero visualizar la cantidad posible de clientes que puedan llegar al local comercial Para tomar de decisiones de negocio.

Escenario de prueba:
Dada la visualización de posibles clientes que puedan llegar al comercial Cuando se presione el botón "Generar Pronóstico" Entonces se presentara información que indique la cantidad de vehículos que pueden llegar al estacionamiento.

Pre-Condiciones
- Tener acceso a Internet.
- Contar con un Navegador Web.
- Contar con una cuenta registrada en la aplicación web.

Pasos y condiciones de ejecución
- Ingresar a la dirección web de la página.
- Ingresar con las credenciales previamente registradas.
- Presionar el Icono de 3 rayas.
- Presionar el botón "Pronostico IA".
- Presionar el Botón "Entrenar Modelo" si el estado es "No Entrenado".
- Presionar el Botón "Volver a Entrenar Modelo" si el estado es "Entrenado".
- Seleccionar una Fecha, máximos hasta 10 días de la fecha actual.
- Elija la Parte del Día en la lista.
- Presione el Botón "Generar Pronostico".

Resultados Esperados
- Mostrar si la cantidad de vehículos que llegaran será "ALTA" o "BAJA".

Estado de Prueba	Éxito	Fallo
	Si	No

Errores Asociados:

Lic. Arturo Miñaca López
Product Owner

Dirección: Vía a Chone Km. 2.
Código postal: 230203 / **Teléfono:** (593-x) xxx xxxx **Ext.** XXXX
Santo Domingo - Ecuador / www.pucesd.edu.ec

Pontificia Universidad Católica del Ecuador | SANTO DOMINGO
Seréis mis testigos
UNIDAD ACADÉMICA O ADMINISTRATIVA

CASO DE PRUEBA 07 **Fecha:** 08/12/2023

Nombre caso de prueba: Reporte de E/S Vehicular **Sprint**: 2
Módulo/sección a evaluar: Reporte de E/S Vehicular **Historia de usuario asociada**: 7
Técnica de Prueba: Caja Negra [X] Caja Blanca | **Tipo:** Prueba de Aceptación
Descripción:
Como Administrador Quiero seleccionar la información de los vehículos Para mejorar el control y la administración del estacionamiento.

Escenario de prueba:
Dado la selección de la información de los vehículos Cuando se pulse el botón "Generar Reporte" Entonces se muestra información detallada de la entrada y salida de los vehículos.

Dado la no selección de la información de los vehículos Cuando se pulse el botón "Generar Reporte" Entonces se muestra un mensaje de error.

Pre-Condiciones
- Tener acceso a Internet.
- Contar con un Navegador Web.
- Contar con una cuenta registrada en la aplicación web.

Pasos y condiciones de ejecución
- Ingresar a la dirección web de la página.
- Ingresar con las credenciales previamente registradas.
- Presionar el Icono de 3 rayas.
- Presionar el botón "Reportes".
- Presionar el botón "E/S de Vehículos".
- Seleccionar una Fecha.
- Ingresar la Placa del Vehículo (Opcional).
- Presione el Botón "Generar Reporte".

Resultados Esperados
- Mostrar una tabla con información de los siguientes campos: ID, Fecha, Hora de Entrada, Hora de Salida, Usuario, Placa del Vehículo, Active y Guardia.

Estado de Prueba	Éxito	Fallo
	Si	No

Errores Asociados:

Lic. Arturo Miñaca López
Product Owner

Dirección: Vía a Chone Km. 2.
Código postal: 230203 / **Teléfono:** (593-x) xxx xxxx **Ext.** XXXX
Santo Domingo - Ecuador / www.pucesd.edu.ec

Pontificia Universidad Católica del Ecuador | SANTO DOMINGO
Seréis mis testigos
UNIDAD ACADÉMICA O ADMINISTRATIVA

CASO DE PRUEBA 08 **Fecha:** 08/12/2023

Nombre caso de prueba: Registro Automático de E/S Vehicular **Sprint**: 3
Módulo/sección a evaluar: Registro Automático de E/S Vehicular **Historia de usuario asociada**: 8
Técnica de Prueba: Caja Negra [X] Caja Blanca | **Tipo:** Prueba de Aceptación
Descripción:
Como Guardia Quiero escanear el Código QR del cliente Para registrar la entrada o la salida del vehículo.

Escenario de prueba:
Dado el escaneo del Código QR del cliente Cuando seleccione un vehículo y pulse el botón "Registra Entrada" Entonces se muestra un mensaje de confirmación que el registro de entrada fue exitoso.

Dado el escaneo del Código QR del cliente Cuando se pulse el botón "Registra Salida" Entonces se muestra un mensaje de confirmación que el registro de salida fue exitoso.

Pre-Condiciones
- Tener acceso a Internet.
- Contar con un Navegador Web.
- Contar con una cuenta registrada en la aplicación web.

Pasos y condiciones de ejecución
- Ingresar a la dirección web de la página.
- Ingresar con las credenciales previamente registradas.
- Escanear el Código QR de un usuario registrado.
- Seleccionar el Vehículo que va a ingresar al estacionamiento.
- Presionar el botón "Registrar Entrada".
- Escanear el Código QR cuando el vehículo salga.
- Presionar el botón "Registrar Salida".

Resultados Esperados
- Mostrar un mensaje que diga "Registro de Entrada Exitoso" para la entrada.
- Mostrar un mense que diga "Registro de Salida Exitoso" para la salida.

Estado de Prueba	Éxito	Fallo
	Si	No

Errores Asociados:

Lic. Arturo Miñaca López
Product Owner

Dirección: Vía a Chone Km. 2.
Código postal: 230203 / **Teléfono:** (593-x) xxx xxxx **Ext.** XXXX
Santo Domingo - Ecuador / www.pucesd.edu.ec

Pontificia Universidad Católica del Ecuador | SANTO DOMINGO
Seréis mis testigos
UNIDAD ACADÉMICA O ADMINISTRATIVA

CASO DE PRUEBA 09 **Fecha:** 08/12/2023

Nombre caso de prueba: Registro Manual de E/S Vehicular **Sprint**: 3
Módulo/sección a evaluar: Registro Manual de E/S Vehicular **Historia de usuario asociada**: 9
Técnica de Prueba: Caja Negra [X] Caja Blanca | **Tipo:** Prueba de Aceptación
Descripción:
Como Guardia Quiero ingresar la placa Para registrar la entrada o la salida del vehículo.

Escenario de prueba:
Dado el ingreso de la placa Cuando se pulse el botón "Verificar Placa" y el botón "Registrar Entrada" Entonces se muestra un mensaje de confirmación que el registro de entrada fue exitoso.

Dado el ingreso de la placa Cuando se pulse el botón "Verificar Placa" y el botón "Registrar Salida" Entonces se muestra un mensaje de confirmación que el registro de salida fue exitoso.

Pre-Condiciones
- Tener acceso a Internet.
- Contar con un Navegador Web.
- Contar con una cuenta registrada en la aplicación web.

Pasos y condiciones de ejecución
- Ingresar a la dirección web de la página.
- Ingresar con las credenciales previamente registradas.
- Presionar el Icono de 3 rayas.
- Presionar el botón "Registro Manual".
- Ingresar las Placa del Vehículo.
- Presionar el botón "Verificar La Placa".
- Presionar el botón "Registrar Entrada".
- Ingresar las Placa del Vehículo cuando salga del estacionamiento.
- Presionar el botón "Verificar La Placa".
- Presionar el botón "Registrar Salida".

Resultados Esperados
- Mostrar un mensaje que diga "Registro de Entrada Exitoso" para la entrada.
- Mostrar un mense que diga "Registro de Salida Exitoso" para la salida.

Estado de Prueba	Éxito	Fallo
	Si	No

Errores Asociados:

Lic. Arturo Miñaca López
Product Owner

Dirección: Vía a Chone Km. 2.
Código postal: 230203 / **Teléfono:** (593-x) xxx xxxx **Ext.** XXXX
Santo Domingo - Ecuador / www.pucesd.edu.ec

Anexo VIII: Evidencia por foto de Implementación

Anexo IX: Evidencia por video de Implementación

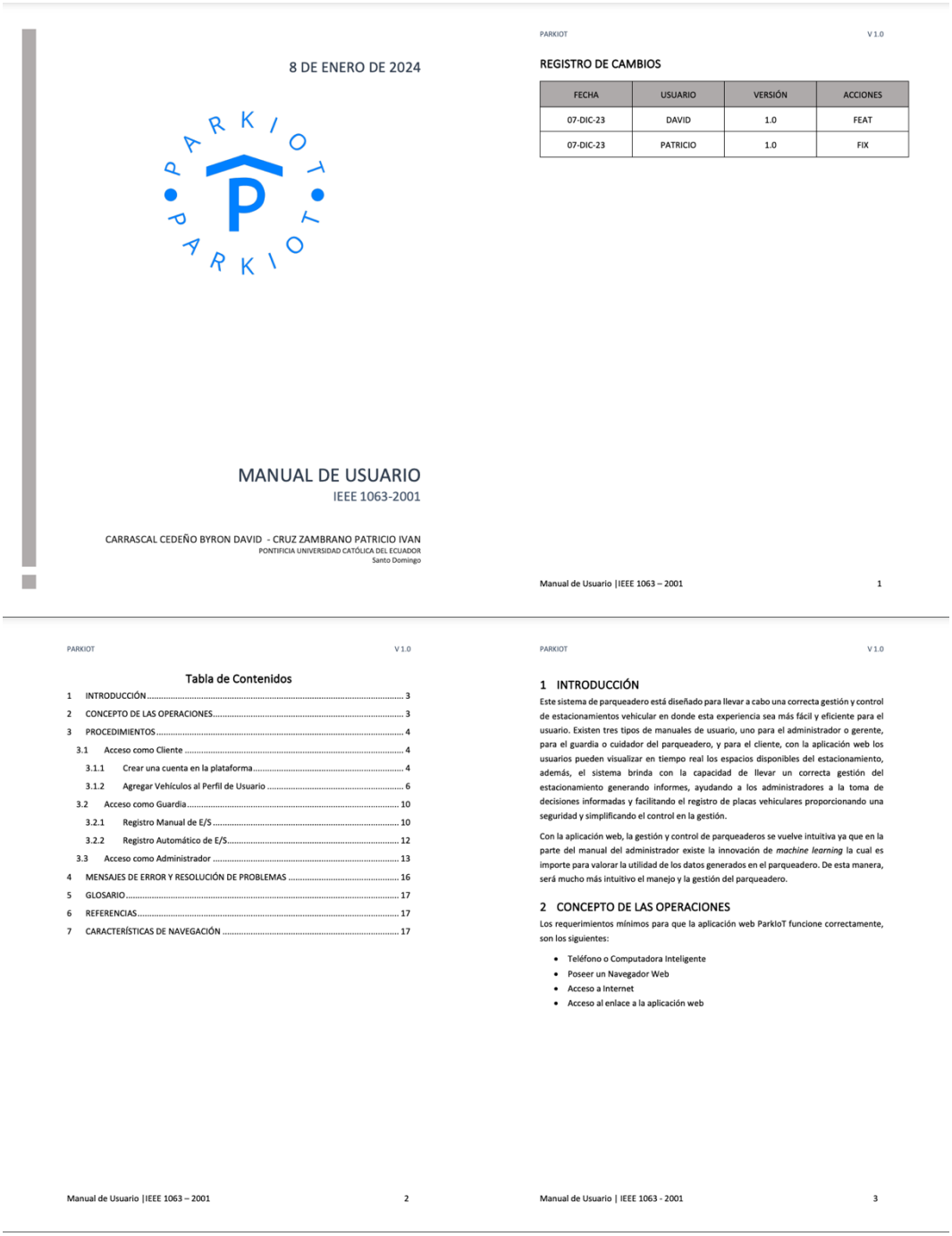
https://drive.google.com/file/d/1azMBtfCPxwu7LkUTOEufRysDmxFWuQtb/view?usp=sharing

https://drive.google.com/file/d/1S2LLcFarl2UwWkuEdyPCYUQN1Pkf3M97/view?usp=sharing

Anexo X: Datos de Registros Generados

https://www.kaggle.com/datasets/patriciocruz/registros-de-parkiot

Anexo XI: Manual de Usuario



8 DE ENERO DE 2024

MANUAL DE USUARIO

IEEE 1063-2001

CARRASCAL CEDEÑO BYRON DAVID - CRUZ ZAMBRANO PATRICIO IVAN
PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
Santo Domingo

REGISTRO DE CAMBIOS

FECHA	USUARIO	VERSIÓN	ACCIONES
07-DIC-23	DAVID	1.0	FEAT
07-DIC-23	PATRICIO	1.0	FIX

Tabla de Contenidos

1 INTRODUCCIÓN

Este sistema de parqueadero está diseñado para llevar a cabo una correcta gestión y control de estacionamientos vehicular en donde esta experiencia sea más fácil y eficiente para el usuario. Existen tres tipos de manuales de usuario, uno para el administrador o gerente, para el guardia o cuidador del parqueadero, y para el cliente, con la aplicación web los usuarios pueden visualizar en tiempo real los espacios disponibles del estacionamiento, además, el sistema brinda con la capacidad de llevar un correcta gestión del estacionamiento generando informes, ayudando a los administradores a la toma de decisiones informadas y facilitando el registro de placas vehiculares proporcionando una seguridad y simplificando el control en la gestión.

Con la aplicación web, la gestión y control de parqueaderos se vuelve intuitiva ya que en la parte del manual del administrador existe la innovación de *machine learning* la cual es importe para valorar la utilidad de los datos generados en el parqueadero. De esta manera, será mucho más intuitivo el manejo y la gestión del parqueadero.

2 CONCEPTO DE LAS OPERACIONES

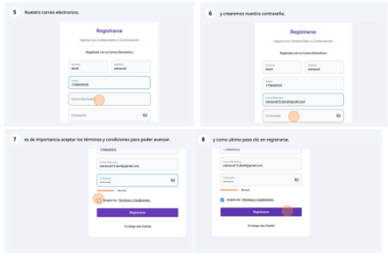
Los requerimientos mínimos para que la aplicación web ParkIoT funcione correctamente, son los siguientes:

- Teléfono o Computadora Inteligente
- Poseer un Navegador Web
- Acceso a Internet
- Acceso al enlace a la aplicación web

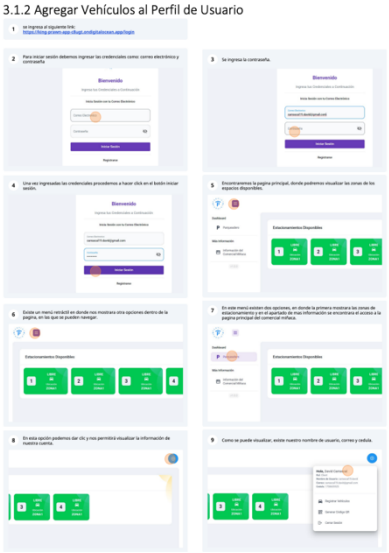
3 PROCEDIMIENTOS

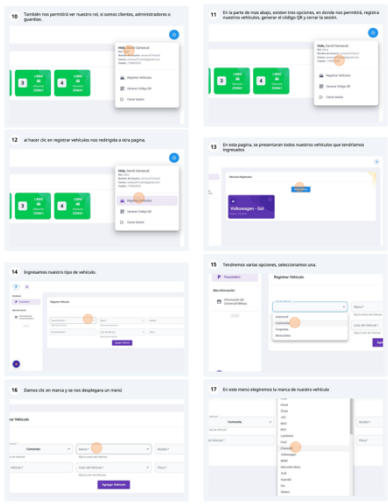
3.1 Acceso como Cliente

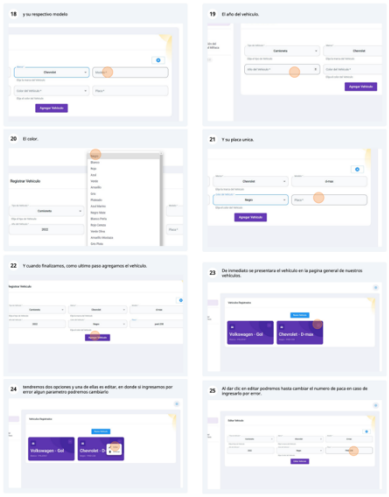
3.1.1 Crear una cuenta en la plataforma

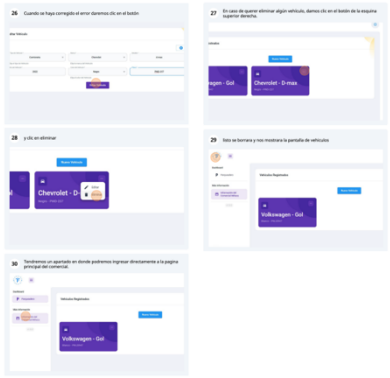


3.1.2 Agregar Vehículos al Perfil de Usuario



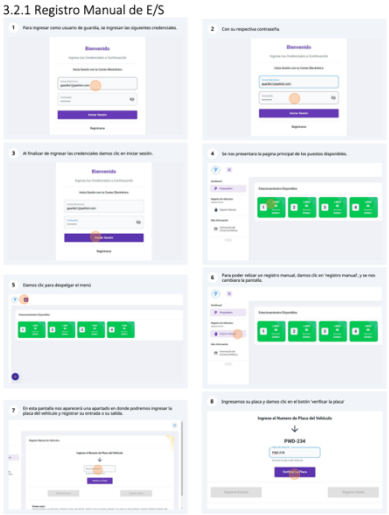


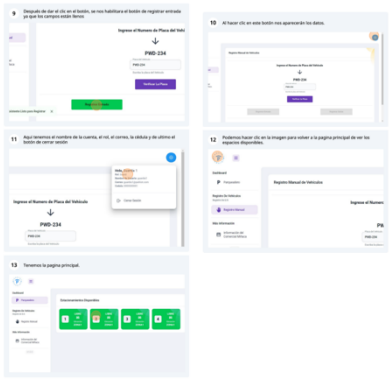




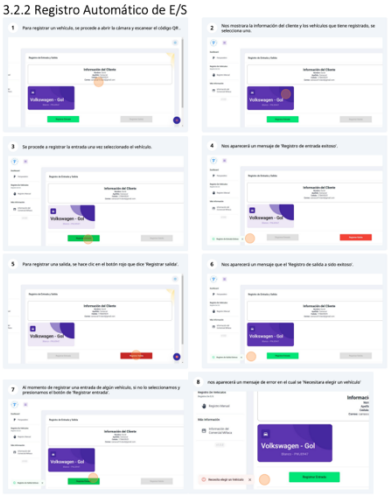
3.2 Acceso como Guardia

3.2.1 Registro Manual de E/S

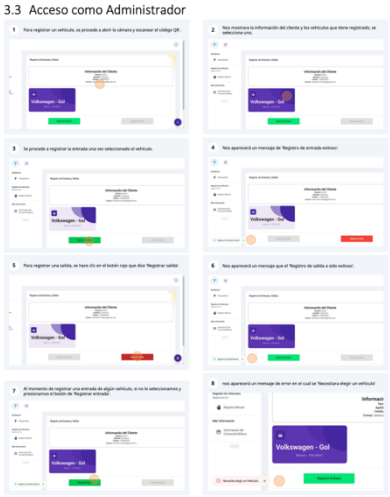


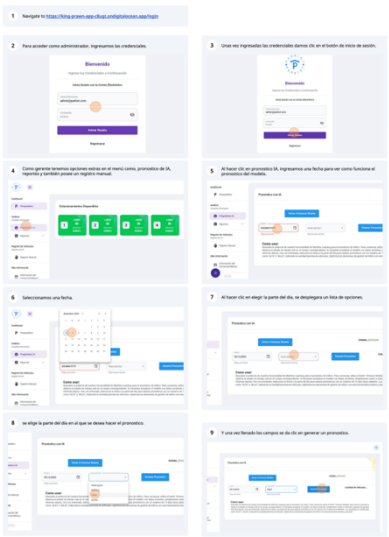


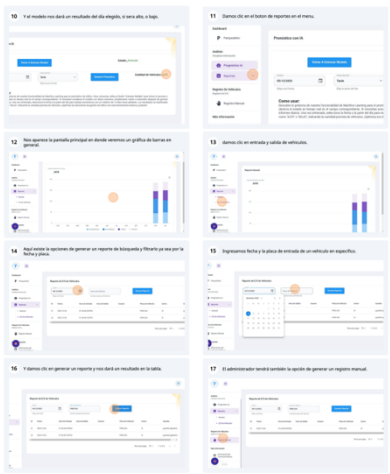
3.2.2 Registro Automático de E/S

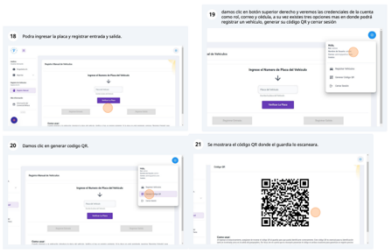


3.3 Acceso como Administrador









4 MENSAJES DE ERROR Y RESOLUCIÓN DE PROBLEMAS

Todos los errores que pueden producirse dentro de ParkIoT, son presentados mediante un cuadro de color con un mensaje que describe la causa del error, haciendo fácil su comprensión e implícitamente dándole al usuario la manera de solucionarlo.

5 GLOSARIO

- **Pronostico IA:** Es una estimación futura en donde la inteligencia artificial se encarga de proporcionar.
- **Modelo:** Es una representación estructura que describe un concepto.
- **Reportes:** Documento que presenta información detalla y organizada.
- **Credenciales:** Es la información de identificación o acceso como nombres de usuario y contraseña .

6 REFERENCIAS

- IEEE (2001). *1063-2001 - IEEE Standard for Software User Documentation* [Internet]. Recuperado de: https://ieeexplore.ieee.org/document/974401

7 CARACTERÍSTICAS DE NAVEGACIÓN

Botón	Funcionalidad
	Despliega el menú lateral
	Presenta la lista de opciones disponibles
	Ingresa al menú de Opciones

Anexo XII: Manual Técnico

8 DE ENERO DE 2024

MANUAL TÉCNICO

IEEE 1063-2001

CARRASCAL CEDEÑO BYRON DAVID - CRUZ ZAMBRANO PATRICIO IVAN
PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
Santo Domingo

REGISTRO DE CAMBIOS

FECHA	USUARIO	VERSIÓN	ACCIONES
07-DIC-23	DAVID	1.0	FEAT
07-DIC-23	PATRICIO	1.0	FIX

Tabla de Contenidos

1 INTRODUCCIÓN

En este manual técnico se visualizará la estructura de la aplicación web en base a los frameworks utilizados, para iniciar, es importante comprender el proceso de clonación del repositorio oficial de la aplicación, que se encuentra en GitHub. Este manual guía, empieza en la clonación y procede en la explicación de los archivos en el backend, para culminar con la explicación de la estructura de archivos del frontend.

Al adentrarse en la estructura de las carpetas y su funcionalidad, se describirá como cada directorio está ubicado de manera estratégica para lograr un funcionamiento global en la aplicación web. El conocimiento obtenido de este manual, permitirá obtener una visión clara de la organización y funcionamiento internos, facilitando así el mantenimiento y la colaboración, en el proceso de desarrollo, mejoras y arreglos.

2 CONCEPTO DE LAS OPERACIONES

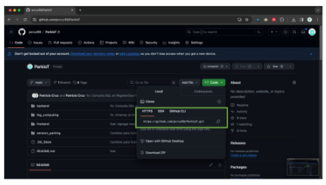
Los requerimientos mínimos para que la aplicación web ParkIoT funcione correctamente, son los siguientes:

- Teléfono o Computadora Inteligente
- Poseer un Navegador Web
- Acceso a Internet
- Acceso al enlace de la aplicación web

3 PROCEDIMIENTOS

3.1 Clonar repositorio de GitHub

Ingrese a la página de GitHub que contiene el repositorio del proyecto y copie la dirección https del repositorio.



Esta dirección http será utilizada para copiar el repositorio con en local con la herramienta Git.

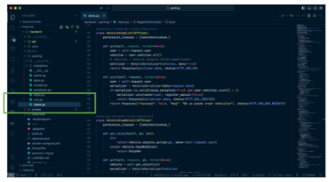
Utilice la terminal para dirigirse al directorio donde quiere clonar todo el proyecto, una vez posicionado en el directorio, ingrese el siguiente comando:

"git clone https://github.com/pcruz99/ParkIoT.git"

3.2 Archivos del Backend

3.2.1 Archivo de Vistas

En este archivo se estructura todas las vistas que forman para de la arquitectura MVT, en este archivo se encuentra toda la lógica de la aplicación web que sera utilizada para recoger y responder peticiones HTTP REST.



3.2.2 Archivo de Serializadores

Este archivo se complementa con las vistas, ya que permite responder a peticiones HTTP GET con el formato JSON de la información requerida, además de personalizar la manipulación de los datos, tanto para agregar como para actualizar la información.

3.2.3 Archivo de Rutas

En este archivo se almacenan todas las url que redirigen a las vistas, estas url son capaces de obtener datos mediante la ruta como path params o query params, cada ruta está directamente relacionada con una vista del archivo de vistas.

Estas rutas son utilizadas por el frontend, para realizar peticiones http y de esta forma obtener información del backend y realizar procesos lógicos que logren el objetivo en la funcionalidad de la aplicación web.

3.3 Archivos del Frontend

3.3.1 Carpeta de Componentes

En la carpeta de componentes, se estructura todos los archivos como tarjetas, grafico o barras de carga, que forman parte de una estructura mayor, estos componentes son reutilizables en cualquier página.

3.3.2 Carpeta de Paginas

En esta carpeta se encuentran todos las paginas que son visualizadas en la aplicación web, las páginas se estructura por secciones y cada página contiene diferentes componentes para crearla. Las páginas usan Axios para realizar peticiones la backend.

3.3.3 Carpeta de Rutas

En esta carpeta se encuentran archivos JavaScript que utilizan la librería React Router para crear rutas que redirijan a las distintas paginas creadas, además que las rutas permiten definir los privilegios de acceso que tienen los usuarios que inicias sesión.

3.3.4 Carpeta de Contextos

En esta carpeta se encuentran archivos de JavaScript que utilizan la librería de Redux para crear contextos que son accesibles de manera global, además de crear Reducers, que permiten manipular los contextos con funciones personalizadas.

3.4 Archivos del Microcontrolador

3.4.1 Archivo main

En este archivo se encuentra el condigo que se ejecutara continuamente, en el tiempo que este encendido el microcontrolador, el objetivo del código es enviar la distancia que mide el sensor al servidor bróker MQTT.

3.4.2 Archivo boot

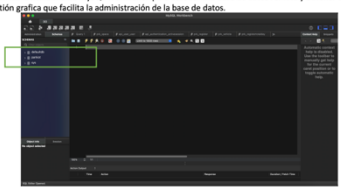
En este archivo se configura la conexión del microcontrolador con la red wifi y con el servidor bróker MQTT.

4 ACCESO A LOS DATOS

4.1 DigitalOcean Databases (MySQL)

La base de datos fue desplegada en un servidor en la nube, mas concretamente en los servicios PaaS de DigitalOcean, para poder acceder a la base de datos se utiliza las credenciales y direcciones, proveídas por la plataforma.

Se utiliza la herramienta MySQL Workbench para acceder a la base de datos remota y obtener una gestión grafica que facilita la administración de la base de datos.



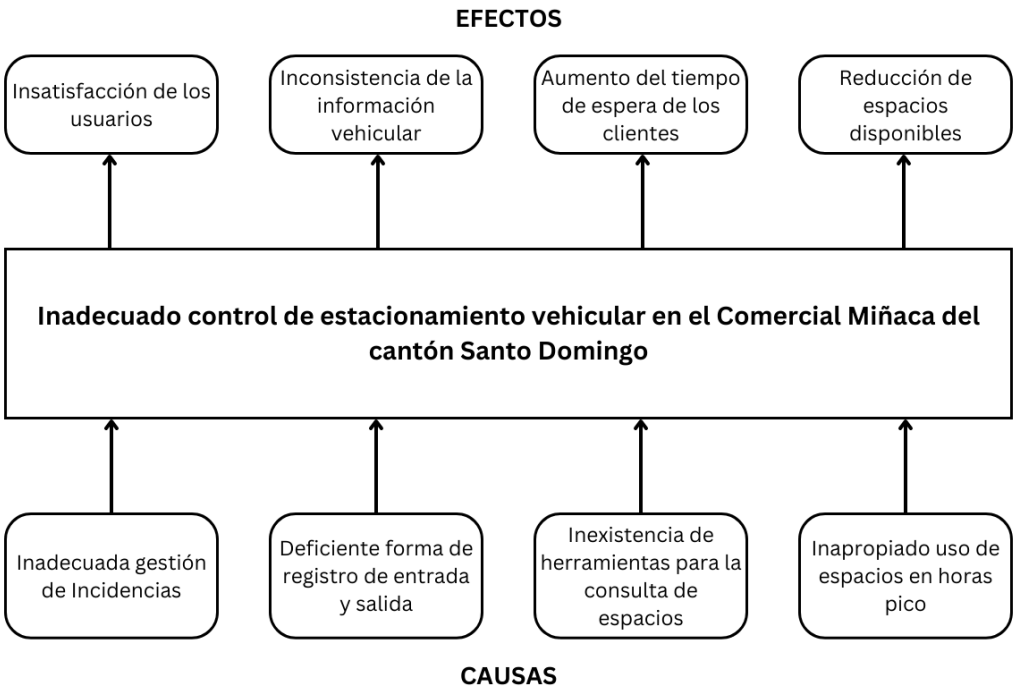
5 GLOSARIO

- **Aplicación Web:** La aplicación web ofrece una experiencia interactiva e innovadora, fusionando una interfaz amigable con la potencia de avanzados sensores.
- **Backend:** El backend constituye la columna vertebral de la aplicación, gestionando datos y operaciones fundamentales para su funcionamiento.
- **Broker MQTT:** El broker MQTT actúa como intermediario en la comunicación de dispositivos.
- **Framework:** El framework simplifica el desarrollo de la aplicación y acelera la implementación de nuevas funcionalidades.
- **Frontend:** El frontend es la interfaz de usuario de la aplicación.
- **Microcontrolador:** El microcontrolador es el cerebro embebido de nuestro sistema, ejecutando tareas y gestionando la interacción con los sensores.
- **MVT:** Modelo-Vista-Controlador, es una arquitectura que organiza la aplicación en tres componentes fundamentales.
- **Repositorio:** El repositorio es un almacén centralizado que guarda y organiza el código fuente de la aplicación.

6 REFERENCIAS

- IEEE (2001). *1063-2001 - IEEE Standard for Software User Documentation* [Internet]. Recuperado de: https://ieeexplore.ieee.org/document/974401

Anexo XIII: Árbol del Problema



Anexo XIV: Recodificación del Instrumento de Recolección de Datos

Recodificación de las Respuestas que no firma parte de la escala de Likert

Recodificación	Respuestas
Tipo de Teléfono	
3	Teléfono Inteligente
2	Teléfono Básico
1	Teléfono convencional
0	Ninguna
Tipo de Internet	
3	Internet Prepago (Recargas)
2	Internet Pospago (Plan)
1	Internet mediante WiFi
0	Ninguno
Tipo de Experiencia	
4	Excelente
3	Buena
2	Regular
1	Mala
0	Pésima
Tipo de Vehiculó	
1	Motocicleta
2	Automóvil
3	Camioneta
4	Furgoneta
Edad	
1	18-25
2	26-30
3	30-40
4	40 o más
Genero	
1	Femenino
2	Masculino
Tipo de Discapacidad	
4	Auditiva
3	Visual
2	Fisica
1	Otro
0	Ninguna

Recodificación de la Esca de Likert

Recodificación	Escala de Likert
Frecuencia	
4	Muy Frecuentemente
3	Frecuentemente
2	Ocasionalmente
1	raramente
0	Nunca
De Acuerdo	
4	Totalmente de acuerdo
3	De acuerdo
2	Neutral
1	Poco de acuerdo
0	Nada de acuerdo
Complejidad	
4	Muy complicado
3	Complicado
2	Neutral
1	Poco complicado
0	Nada complicado
Importancia	
4	Muy importante
3	Importante
2	Neutral
1	De poca importancia
0	Sin importancia

Anexo XV: Informe del Turniting

TTG_09022024_verFINAL

INFORME DE ORIGINALIDAD

4%	3%	1%	1%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

ENCONTRAR COINCIDENCIAS CON TODAS LAS FUENTES (SOLO SE IMPRIMIRÁ LA FUENTE SELECCIONADA)

1%

★ www.scribd.com

Fuente de Internet

Excluir citas	Activo	Excluir coincidencias	< 10 words
Excluir bibliografía	Activo		