UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA DE INGENIERÍA DE SISTEMAS

DISEÑO DE UNA APLICACIÓN MÓVIL QUE PERMITA EL REGISTRO DE VEHÍCULOS VISITANTES Y CONSULTA DE LA DISPONIBILIDAD DE ESPACIOS DE PARQUEO.

Trabajo de titulación previo a la obtención del

Título de Ingenieros de Sistemas

AUTORES: CRISTHIAN ORLANDO GUACHÁN CUÁSQUER

DIEGO JUVENAL SIGCHO ALVARADO

TUTORA: LINA PATRICIA ZAPATA MOLINA

Quito - Ecuador

2022

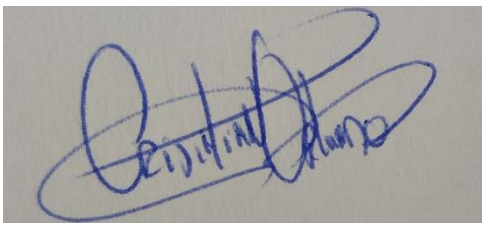
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

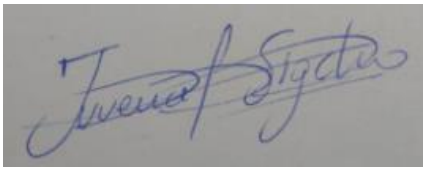
Nosotros, Cristhian Orlando Guachán Cuásquer, con documento de identificación N° 1724070287 y Diego Juvenal Sigcho Alvarado con documento de identificación N° 1727089961; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 16 de septiembre del año 2022

Atentamente,





Cristhian Orlando Guachán Cuásquer

1724070287

Diego Juvenal Sigcho Alvarado

1727089961

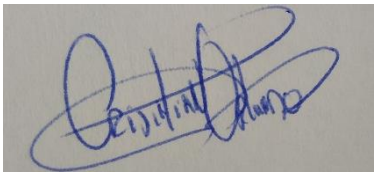
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Cristhian Orlando Guachán Cuásquer, con documento de identificación N° 1724070287 y Diego Juvenal Sigcho Alvarado con documento de identificación N° 1727089961, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: "Diseño de una aplicación móvil que permita el registro de vehículos visitantes y consulta de la disponibilidad de espacios de parqueo.", el cual ha sido desarrollado para optar por el título de: Ingenieros de Sistemas, en la Universidad Politécnica Salesiana, quedando la universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

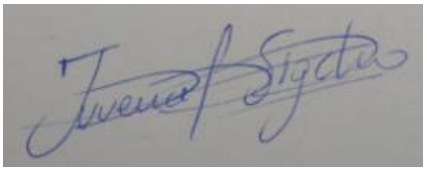
Quito, 16 de septiembre del año 2022

Atentamente,



Cristhian Orlando Guachán Cuásquer

1724070287



Diego Juvenal Sigcho Alvarado

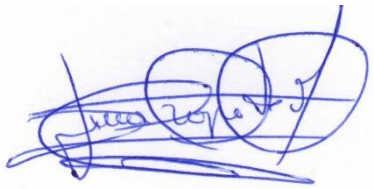
1727089961

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Lina Patricia Zapata Molina con documento de identificación Nº 0501877278 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO DE UNA APLICACIÓN MÓVIL QUE PERMITA EL REGISTRO DE VEHÍCULOS VISITANTES Y CONSULTA DE LA DISPONIBILIDAD DE ESPACIOS DE PARQUEO., realizado por Cristhian Orlando Guachán Cuásquer, con documento de identificación N° 1724070287 y Diego Juvenal Sigcho Alvarado con documento de identificación N° 1727089961, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 16 de septiembre del año 2022

Atentamente,



_____

Ing. Lina Patricia Zapata Molina, PhD

0501877278

DEDICATORIA

Esta tesis está dedicada a:

Mis padres que con su esfuerzo y tiempo me acompañaron en el camino a conseguir un sueño más, por inculcarme siempre el ejemplo de la responsabilidad y dedicación, por siempre ayudarme a levantar de cualquier tropiezo.

A mi hermana que con su ejemplo siempre me guio a seguir adelante y no desfallecer para conseguir lo que quería, por demostrarme que todo se puede y por darme siempre palabras de aliento.

Finalmente, a todas las personas que estuvieron conmigo durante este periodo dándome palabras de apoyo y ayudándome a lograr mi meta.

De: Cristhian Guachán

Esta dedicatoria va destinada principalmente a mi madre, quien me ha dado la fuerza, el apoyo y el valor para seguir adelante siendo mi inspiración y el motivo de haber llegado hasta donde estoy el día de hoy.

A mi familia a quien tengo especial amor y gratitud por haberme apoyado y hacer lo posible para darme una mano y continuar cuando más lo necesite.

A los amigos y personas que conocí en el camino los cuales fueron clave para mi crecimiento como persona y profesional.

Y finalmente a todos los profesores que en su momento me dieron la oportunidad de aprender de ellos.

De: Diego Sigcho

AGRADECIMIENTOS

Queremos agradecer principalmente a nuestros profesores por habernos brindado de su conocimiento para nuestro crecimiento profesional, gracias a cada uno de ustedes por haber sido pacientes durante este periodo de aprendizaje.

A nuestra tutora, Ing. Lina Zapata por habernos apoyado durante este periodo de realización de nuestro trabajo de titulación.

A nuestros amigos que han estado ahí para apoyarnos y darnos consejos durante todo el tiempo compartido en las aulas y fuera de ellas, por cada una de las palabras de aliento para seguir adelante.

Por último, a todas las personas que directa o indirectamente formaron parte de este proceso de conseguir nuestro título universitario, que con las vivencias y tiempo compartido nos ayudaron obtener una mejor formación.

Diego Sigcho, Cristhian Guachán

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

El presente documento describe el proceso para el desarrollo de una aplicación que se desplegará en dispositivos móviles permitiendo al usuario final identificar con anticipación, de forma fácil, rápida y en tiempo real los lugares que se encuentran disponibles en una zona de parqueo. El sistema detecta y obtiene los dos posibles estados de un lugar de parqueo, disponible u ocupado, basado en el reconocimiento de los vehículos mediante cámaras de seguridad.

Se adquirieron estos datos para presentarlos en una interfaz gráfica, que contiene un mapa del lugar de estacionamiento que permita al usuario visualizar gráficamente la información y ubicación de los espacios disponibles u ocupados, tomando en cuenta que, solo los usuarios registrados en la base de datos del parqueadero tendrán acceso a esta interfaz. Además, también se desarrolló un segundo aplicativo móvil que permite llevar un registro de las visitas para aquellos usuarios que no forman parte de las personas que tienen acceso a la aplicación descrita anteriormente, definiendo a estos usuarios como visitantes.

Como resultado del desarrollo de este proyecto, se obtuvo dos aplicaciones que cumplen con los requerimientos establecidos y los objetivos planteados, para entregar al usuario una nueva herramienta que optimice y facilite su proceso de parqueo, agilizando la movilidad dentro de la zona y el control de los accesos a la misma.

ABSTRACT

This document describes the process for the development of an application that will be deployed on mobile devices allowing the end user to identify in advance, easily, quickly and in real time the places that are available in a parking area. The system detects and obtains the two possible states of a parking space, available or occupied, based on the recognition of vehicles by security cameras.

These data were acquired to present them in a graphical interface, which contains a map of the parking place that allows the user to graphically visualize the information and location of the available or occupied spaces, considering that only users registered in the database of the parking lot will have access to this interface. In addition, a second mobile application was also developed that allows keeping a record of visits for those users who are not part of the people who have access to the application described above, defining these users as visitors.

As a result of the development of this project, two applications were obtained that meet the established requirements and the proposed objectives, to provide the user with a new tool that optimizes and facilitates their parking process, speeding up mobility within the area and controlling traffic.

INTRODUCCIÓN

Actualmente, el tener un automóvil más que un lujo se ha convertido en una necesidad para ciertas personas, poder movilizarse de forma libre, cómoda y segura. Este hecho ha generado un incremento en el número de automóviles de las ciudades. Por otro lado, la tecnología y el internet hoy en día cumplen un rol casi indispensable, ya que es una fuente casi infinita de información y ya con los avances existen funciones que han superado las capacidades humanas Rueda-López, J. J. (2007).

Según el INEC (2020), el número de vehículos matriculados en Ecuador desde el año 2009 ha crecido desde las 872.388 unidades hasta alcanzar las 2,361.175 en el año 2020 encontrando su punto más alto en el año 2018 con 2.403.651, es decir, en 11 años el índice se ha incrementado en un 67.3% y la cantidad de vehículos matriculados desde el 2018 tiene una tendencia a mantenerse sobre los 2 millones de unidades a nivel nacional.

Según datos recogidos por el INEC en los años comprendidos entre el 2015 y 2020 el parque automotor ha ido aumentando con un promedio de más de dos millones de unidades anuales, esto se traduce a una mayor cantidad de tránsito en las principales calles y accesos a la ciudad incrementa los tiempos de espera que una persona debe esperar para llegar desde sus hogares hacia sus trabajos y viceversa, y este es un efecto difícil de ignorar ya que movilizarse es parte del día a día de cada persona.

Otro punto para tomar en cuenta es que las plazas de parqueo en zonas urbanas, tales como zona azul, parqueaderos de ciertos lugares concurridos por temporadas como centros comerciales, unidades educativas, instituciones en las que las personas acuden de manera recurrente, no solo es una pérdida de tiempo, según Zadeh y Dela Cruz (2016) a parte de generar un gasto innecesario de energía, esta búsqueda causa más tráfico y contaminación al aire.

Algunos estacionamientos cuentan con sistemas que permiten contabilizar la cantidad de espacios de parqueo disponibles en el lugar, estos se muestran a los conductores en la entrada de los mismos, y se basan en sensores que delimitan cada zona de parqueo e identifican si un vehículo la está ocupando o no, así los conductores pueden identificar antes de ingresar a la zona de parqueo si hay espacios para su vehículo inclusive, pueden identificar en qué nivel puede encontrar una zona en la que aparcar (M. P. Thakre, P. S. Borse, N. P. Matale and P. Sharma,2021).

Pero hoy en día, es complicado encontrar soluciones eficaces para dar solución a este problema. En estos días, la comunicación es tan importante para las personas el acceder a la información desde cualquier lugar, por tanto, proponer una solución de este tipo busca darle la posibilidad a los usuarios que se transportan mediante sus vehículos de anticipar cual sería la mejor opción a la cual dirigirse con información en tiempo real.

CAPÍTULO I

PROBLEMA DE ESTUDIO

En los últimos años la tasa de delincuencia en el país ha ido incrementando, podría decirse que en la misma medida que la de los vehículos que adquieren las personas en Ecuador, esto ha llevado a encontrar sucesos delictivos ya no solo a personas que transitan por la calle, sino también a los conductores en sus vehículos, los cuales ahora suelen ser víctimas de la delincuencia mientras se encuentran en el tráfico o son asaltados en los parqueaderos donde acuden para dejar sus vehículos mientras van a realizar sus trámites personales (Carrera, F. M., Govea, F. K., Hurtado, G. E., & Freire, C. E. (2019)).

Teniendo en cuenta que al momento de que un vehículo acuda a una zona de parqueo y tenga problemas para ubicar un lugar disponible puede perjudicar al flujo de tráfico generando embotellamientos, la necesidad de crear esta aplicación nace también por las cuestiones de optimización de tiempo para estacionarse, evitar un poco la contaminación ya que, según el sitio fueleconomy.gov, mantener un vehículo encendido sin la necesidad de que este avance puede llegar a consumir entre un cuarto a medio tanque de combustible en el transcurso de una hora dependiendo del cilindraje del motor y que tanto combustible consuma el mismo.

Esto a gran escala puede representar un gran impacto tanto para el medio ambiente debido a que las emisiones de CO2 innecesarias, ya que según Faraji, S. J., & Naozar, M. J cerca del 80% de contaminación la producen vehículos, esto también afecta a la economía de los usuarios de automotores debido a que el consumo de combustible también se mantiene de manera constante solo por el hecho de esperar con su vehículo encendido y el consumo incluso puede aumentar si el auto cuenta con aire acondicionado y se usa constantemente.

Por cuestiones de seguridad, cuidados del medio ambiente y optimización de tiempos, es necesario disponer de aplicaciones informáticas que ayude optimizar el tiempo de acceso vehicular a las zonas de parqueo ya sea en conjuntos habitacionales, centros comerciales, instituciones educativas, entre otros.

El objetivo de este trabajo es desarrollar una aplicación móvil que permita al conductor verificar los espacios de parqueo disponibles en un sitio determinado para que pueda tener un enfoque global del lugar al que acude, así como también conocer anticipadamente hacia donde debe dirigirse.

También involucra el desarrollar una aplicación de registro para vehículos entrantes o salientes que no sean visitantes frecuentes del lugar ya que no se observa que exista una solución tecnológica que tome en cuenta estos aspectos, principalmente enfocando el punto de la seguridad ya que el no llevar el control automatizado de acceso de las personas a ciertos lugares como conjuntos, instituciones educativas etc. puede dar paso a que personas dedicadas a la delincuencia ingresen por ejemplo, con vehículos sin placas a realizar robos, que es algo que en la actualidad se está viendo con mayor frecuencia. Generar medios por el cual se requiera llevar un control de acceso y registro se considera el punto principal de esta implementación

JUSTIFICACIÓN

El presente trabajo está destinado a desarrollar dos aplicaciones móviles, mediante las cuales se busca generar una solución que ayude a disminuir las problemáticas generadas por aquellos factores que dieron lugar al planteamiento del problema de estudio.

La primera aplicación móvil se enfocará en realizar una interfaz visual, que permita al usuario conductor conocer de forma anticipada y en tiempo real, si el estacionamiento al que se dirige posee espacios disponibles para aparcar, así como también las zonas con las que cuenta su lugar de destino.

Otro de los beneficios que se busca generar, también incluye el poder anticipar cual es el lugar al que debe dirigirse para encontrar un espacio libre para dejar su automóvil sin la necesidad de que tenga que invertir tiempo y energía buscándolo o esperando por que se desocupe un lugar.

La segunda aplicación móvil tiene como objetivo implementar un registro digital mediante el cual se pueda llevar un control de los accesos al estacionamiento para los usuarios que son ajenos a la institución, a estos conductores se los denomina como visitantes dentro del sistema. Los datos que se consideran para registrar el acceso serán: el tipo de vehículo, la cantidad de ocupantes que lleva y la placa de este.

GRUPO OBJETIVO (Beneficiarios)

Las personas beneficiarias son: el Guardia de seguridad del lugar en el que se implementará el sistema y las personas que se van a hacer uso de los lugares de parqueo.

OBJETIVOS

Generales

- Desarrollar una aplicación móvil para consultar la disponibilidad o no disponibilidad en las zonas que demanden (el estado de zonas de parqueo.)
- Desarrollar una aplicación móvil para (el registro de visitantes para el ingreso a una determinada zona de parqueo) que el guardia registre los vehículos que ingresan.

Específicos

- Desarrollar la aplicación móvil para conocer la disponibilidad de los espacios de parqueo.
- Desarrollar la aplicación móvil que permita al guardia de una determinada zona de parqueo registrar datos del vehículo visitante.

- Reducir el tiempo de espera para un sitio de parqueo de los visitantes a los parqueaderos donde se use el sistema.
- Mantener un control sobre los visitantes externos a la organización donde se implante el sistema con el fin de que no ingresen personas dedicadas al robo.

METODOLOGÍA

Para el desarrollo del presente proyecto se utilizará la metodología ágil XP, la cual tiene enfoque en proyectos pequeños, es ágil, y eficiente. Esta metodología ayuda a reducir los tiempos de desarrollo de la aplicación (Vicenzo Lozano Angulo, J., 2017). El proyecto se regirá a las fases de la metodología XP, que son: Planificación, Diseño, Desarrollo y Pruebas.

Planificación

La planificación se basa en fechas límite, haciendo uso de tableros Kanban para una mejor organización y control de las iteraciones, organizando reuniones para dar seguimiento a los avances.

Dentro de esta etapa se establece los tiempos para recopilar los requerimientos de todo lo que involucrara el desarrollo, diseñar prototipos tanto funcionales como gráficos de la idea que se planteó para desarrollar la solución, el desarrollo como tal, las pruebas y finalmente la documentación (Borman, R. I., Priandika, A. T., & Edison, A. R. 2020). También se toma de la metodología la distribución del trabajo, que herramientas se utilizara tanto de desarrollo como colaborativas para cada una de las fases y se planifican reuniones para aclarar dudas acerca del desarrollo, y controlar que los tiempos se vayan a cumplir.

Diseño

En cuanto al diseño de la aplicación, se recopilarán los requisitos necesarios que engloben todas las funciones que esta deba cumplir, tanto para la aplicación destinada a los usuarios que se

encuentren registrados en el estacionamiento, así como la aplicación para el registro de visitantes dentro del mismo.

Se maquetarán las interfaces graficas utilizando la herramienta colaborativa Figma para tener una visión clara de lo que se va a desarrollar, y para la funcionalidad y estructuras se utiliza como guía los diagramas UML tales como: diagramas de clases, casos de uso y la navegación de las aplicaciones.

Desarrollo

Se utiliza un repositorio Git para el control de versiones, las iteraciones se realizan en conjunto al desarrollo dirigido por pruebas, que se trata de un sistema mediante el cual se generan pruebas unitarias para ver en que falla la aplicación y así guiar el desarrollo, de este modo, el código que se genere tendrá el mínimo de fallas al momento de las pruebas.

En esta etapa, en base a los requerimientos que se recopilaron y a la diagramación de las interfaces se desarrollará primeramente el aplicativo para los usuarios registrados dentro del estacionamiento. Se toma de la base de datos que proporciona el sistema de reconocimiento de espacios disponibles, los datos de los lugares para poder ilustrar sus estados, una vez diagramadas las zonas por separado se crea una vista que contendrá las zonas de parqueo indicando la cantidad que tiene cada una de espacios disponibles a manera de vista previa, y finalmente, se crea un formulario de acceso para los usuarios registrados, los cuales también se toman de la base de datos y como valor adicional se agregará una pantalla de bienvenida.

En el caso de la segunda aplicación se usa como base del desarrollo anterior, la misma estructura y funcionalidad para crear una pantalla de bienvenida para el usuario, un formulario de registro para recopilar los datos necesarios para registrar una visita, así como también una vista de confirmación para validar que la información ingresada sea correcta y se envié a guardar en la base de datos.

CAPÍTULO II

MARCO TEÓRICO

CONTAMINACIÓN

Factor influyente en el planteamiento del problema, se considera debido a que el gran aumento de vehículos en zonas urbanas ha generado una gran dificultad al momento de buscar un lugar donde estacionar, esto con el tiempo ha aumentado la contaminación en zonas urbanas ya que con la falta de información sobre lugares de estacionamientos libres lleva a que los autos pasen más tiempo emitiendo gases al medio ambiente (Caicedo, F.,2010).

Se toma este tema como relevante debido a que el antecedente acerca de las últimas dos décadas indica que se ha registrado un gran aumento en la contaminación mundial, así como también un gran aumento en la población que no se toma en serio la contaminación ambiental, lo que ha resultado en graves problemas para el planeta como para la sociedad (Chandrasekaran, S. S., Muthukumar, S., & Rajendran, S., 2013).

PARQUEADEROS

Esta será el área de aplicación sobre la cual se implementará la solución tecnología que será el resultado de este trabajo. Se considero como antecedente que, el aumento de vehículos ha generado un problema al momento de conseguir una zona de parqueo libre y la poca información proporcionada por los lugares de parqueo dificultan aún más la búsqueda, generando así más contaminación, el encontrar un espacio disponible requiere más tiempo y causa más estrés en las personas que requieren aparcar su auto, por lo cual una buena información de zonas de parqueo puede reducir en gran medida estos inconvenientes. (Caicedo, F., 2010)

APLICACIÓN MÓVIL

Se refiere al producto terminado de este desarrollo, es un tipo de aplicación diseñada para ejecutarse en dispositivos móviles, los cuales pueden ser teléfonos celulares o tabletas. Cada aplicación puede ser desarrollada con un fin especifico y con funciones limitadas, sin embargo, estas pueden expandirse con el fin de proporcionar a los usuarios servicios más complejos. (Luiz Herazo, 2020)

ANDROID

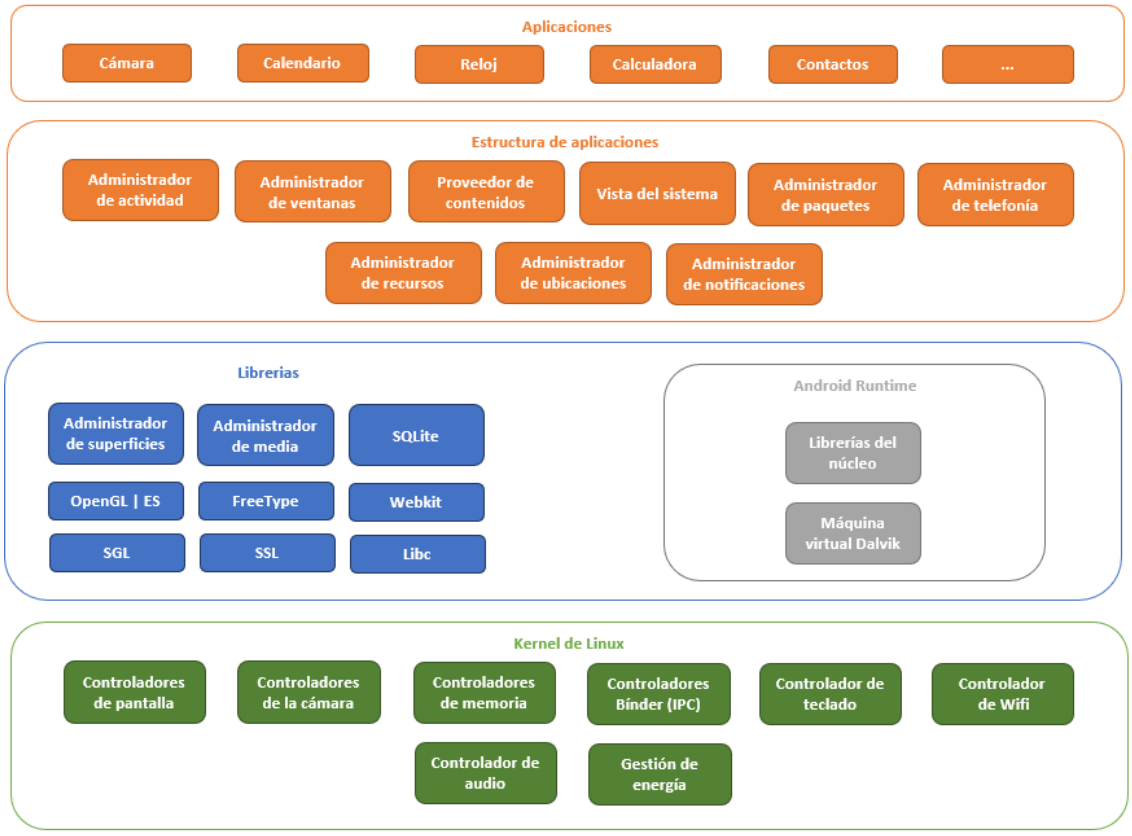
Sistema operativo inicialmente diseñado para teléfonos móviles, que actualmente también se puede encontrar en Tabletas, GPS, Televisores, el cual está basado en Linux, el cual es un núcleo de sistema operativo libre, gratuito y multiplataforma. Este sistema operativo permite desarrollar aplicaciones utilizando una variación del lenguaje de programación conocido como Java. (Robledo, 2016).

ARQUITECTURA DE ANDROID

La arquitectura que implementa Android está basada en 4 capas, las cuales se encuentran relacionadas entre sí. A continuación, se presenta un modelo grafico de cómo se encuentran distribuidas estas capas con cada uno de sus componentes.

Figura 1

Arquitectura Android



Nota. *Capas que presenta la arquitectura del sistema operativo Android conformada por: Kernel, Runtime, Librerías, Estructura de aplicaciones y Aplicaciones. Elaborado por: Guachán, Sigcho.*

Cada una de estas capas que se ilustran en el grafico antes presentado cumplen con una función determinada, las cuales se detallan a continuación, desde la más interna hacia la más externa:

Kernel de Linux: Esta capa contiene el acceso a las funciones que integran los controladores del dispositivo, mediante ellos, se puede entablar la comunicación entre Hardware y Software.

Librerías: Se pueden definir como segmentos de código ya desarrollado mediante el cual se puede implementar de forma rápida y sencilla funcionalidades en el código que se está escribiendo según el lenguaje de programación utilizado.

Estructura de aplicaciones: Administra y gestiona los recursos de los procesos que ejecutan las aplicaciones.

Aplicaciones: Esta es la parte perceptible para el usuario, se puede definir como la interfaz visual que le permite al usuario realizar acciones sobre el dispositivo en cuestión mediante el uso de aplicaciones.

BASES DEL DESARROLLO

Arquitectura Cliente-Servidor

Se utilizo esta arquitectura para poder conectar las aplicaciones con un servidor en el cual se encuentra alojada la base de datos y así poder consultar los estados de cada zona de parqueo. Esta arquitectura corresponde a un patrón de desarrollo, el cual sirve de apoyo para construir el software. Se basa en realizar y atender peticiones entre el usuario o cliente hacia el servidor.

Su funcionamiento se basa en la interacción del usuario que mediante la aplicación realiza peticiones o solicita recursos del servidor, utilizando mensajes los cuales son un conjunto de ordenes donde se detalla que tipo de requerimiento hace el cliente hacia el servidor, el cual puede componerse de una sola estación servidor o puede ser una gran cantidad de equipos conectados entre sí.

Diagramación UML

Se utilizo la diagramación UML para obtener una vista general de como quedarán estructuradas y definidas las aplicaciones, esto proporciona una mejor perspectiva de los requerimientos que estas necesitan y debe cumplir, optimizando principalmente tiempo de desarrollo y minimizando los reprocesos o cambios que puedan llegarse a dar por aspectos que no fueron tomados en cuenta.

UML es un lenguaje que brinda da las pautas para construir modelos que describen que aspecto tendrá y como se va a comportar el software a desarrollar, esto se utiliza durante la planificación

y se compone de vistas mediante las cuales se muestran los distintos aspectos del sistema que se está modelando, los diagramas que no son más que la gráfica que describe el contenido de una vista, símbolos que son elementos que representan conceptos comúnmente referentes a objetos como pueden ser: clases, objetos, relaciones o mensajes. Finalmente se tienen las reglas, las cuales proveen información y dan contexto acerca del elemento dentro del modelo y el rol que desempeña en él.

HERRAMIENTAS DE DESARROLLO

Flutter

El desarrollo de las aplicaciones se realizó mediante Flutter, que es un kit de desarrollo de software (SDK) creado por Google para crear aplicaciones de forma nativa tanto para Android y iOS, la ventaja que ofrece este SDK es que el código que genera lo hace de manera nativa para cada plataforma, dando como resultado aplicaciones rápidas (Flutter).

Dart

El kit de Flutter utiliza como lenguaje de programación Dart, el cual tiene bastante similitud en su sintaxis con lenguajes como: JavaScript, Java, y C++. Dart es un Lenguaje de código abierto desarrollado por Google orientado al desarrollo con Flutter, una de las ventajas más notables que aporta es la compilación *"just-in-time"*, lo cual permite que los cambios que se hayan realizado en el código se puedan ejecutar de manera inmediata sobre el intérprete que se esté utilizando, sea este un navegador, un emulador de algún dispositivo móvil o el propio dispositivo móvil mediante la conexión USB.

Git

Para trabajar de forma conjunta se utiliza el sistema de control de versiones conocido como Git, y se utilizó el repositorio de Azure en donde se almacenan los archivos del proyecto que se está realizando para poder acceder a ellos desde cualquier lugar y ordenador, el uso de esto

fundamental para llevar de manera correcta el ciclo de vida del software (Deepa, N., Prabadevi, B., Krithika, L. B., & Deepa, B. 2020).

Figma

La maquetación de las interfaces fue realizada en la plataforma de Figma, esta es una herramienta colaborativa en línea que cuenta con varios componentes gráficos la cual ayuda con el diseño de interfaces tanto UI como UX. Utilizando esta herramienta se creó el diseño de las pantallas que contendrá el software.

Base de datos.

El resultado del proyecto de reconocimiento de espacios disponibles mediante cámaras es una base de datos que contiene, además de los datos referentes a la administración de ese sistema, la información que se va a mostrar en las aplicaciones móviles y que también resulta necesaria para su funcionamiento, como por ejemplo: el listado de usuarios que tendrán acceso a la aplicación para consultar espacios disponibles, los espacios disponibles con su correspondiente zona y la tabla en la cual se van a insertar los registros de los usuarios visitantes.

Web Service

La conexión de las aplicaciones con la base de datos se realiza con PHP, la estructura definida en el servidor envía los datos que necesita la aplicación para mostrar al usuario la información que solicita según la interfaz a la que este ingresando.

Un web service es una tecnología u objeto, el cual es accesible mediante protocolos de red, su principal ventaja radica en que contribuye al desarrollo de aplicaciones al compartir información y código de forma remota, eliminando la brecha o la distancia que existe entre el cliente y el servidor. Independientemente de las tecnologías en las que se encuentren desarrollados tanto cliente como servidor, el web service permite acceder a las funcionalidades

de cada uno según lo requiera las peticiones y respuestas que se ejecuten de parte y parte entre los sistemas u aplicaciones involucradas. (Ribas, 2003).

METODOLOGÍA XP

Durante la elaboración de este trabajo se utilizó esta metodología de desarrollo ágil, la cual es implementada en proyectos de software pequeño, ya que se asegura de involucrar más al cliente final en el proceso de desarrollo para que este se encuentre al tanto de los avances, de cómo está quedando el producto y pueda dar retroalimentación acerca de su perspectiva del proyecto y el resultado. Este involucramiento del cliente en el proceso de desarrollo permite entregar un producto final que sea muy cercano o totalmente fiel a lo que el desea obtener.

Otras ventajas que ofrece esta metodología incluyen: permitir dar una respuesta rápida a los cambios constantes que se pueden llegar a presentar en el desarrollo del proyecto, así como también que impulsa el trabajo en equipo, lo vuelve transparente y flexible al utilizar un cronograma de actividades en el cual se establecen las tareas y subtareas que conlleva el desarrollo.

El ciclo de vida de esta metodología incluye: Fase de exploración, Fase de planificación, Fase de iteraciones, Fase de puesta en producción, Fase de mantenimiento, siguiendo el flujo que se muestra en la figura 2.

Figura 2

Ciclo de vida de la metodología XP

Nota. Ciclo de vida de la metodología XP una metodología ágil que presenta las siguientes fases para el desarrollo de software. Elaborado por: Guachán, Sigcho.

Fase de exploración: Se recopilaron y analizaron los requerimientos del cliente y el alcance del proyecto, haciendo uso de historias de usuario, el giro de negocio, según Gonzáles (2013) no se deben confundir las historias de usuario con los requerimientos ya que su principal diferencia en la profundidad de análisis.

Fase de planificación: Luego de analizar las historias de usuario se estima el coste de implementación y se realizan una reunión para realizar el plan de entrega. De esto saldrán las iteraciones a las cuales se les debe asignar un tiempo de cada una y se determina el alcance del proyecto (González, J. F.,2013)

Fase de iteraciones: Aquí es donde se crea la solución, en cada una de estas iteraciones se realiza un proceso completo de análisis, diseño, desarrollo y pruebas. Se deben priorizar las iteraciones con más valor para el usuario y de cada iteración se obtienen tareas dándoles un tiempo de desarrollo (González, J. F.,2013).

Fase de producción: Para llegar a la fase de producción no es necesario tener la versión final del proyecto, si no que se alcanza esta fase cuando se tiene una versión que el usuario decida que es un producto válido para ponerlo en producción y aquí se siguen realizando iteraciones como en la fase anterior (González, J. F.,2013).

Figura 3

Iteraciones dentro de la metodología XP

Nota. Fases que se involucran dentro de cada iteración del proceso de desarrollo, estas empiezan desde el análisis y terminan con las pruebas, se puede realizar entre 10 y 15 ciclos.

Elaborado por: Guachán, Sigcho.

CAPÍTULO III

ANÁLISIS Y DISEÑO

Dentro de este capítulo se analizarán los requerimientos del usuario, cual es el alcance que tendrá el proyecto, que funcionalidad deberá tener, así como también se propone el maquetado de las interfaces graficas que serán parte de la solución al problema planteado.

ESPECIFICACIÓN DE REQUERIMIENTOS

Requerimientos de usuario

La aplicación móvil dirigida al guardia de seguridad permitirá a este realizar el registro de los visitantes que no se encuentran registrados en la zona de parqueo, este registro únicamente tomará en cuenta los datos necesarios para identificar al vehículo que ingresa y se almacenarán en una base de datos alojada en la nube.

Por otra parte, la aplicación móvil dirigida a los usuarios autorizados al uso de los parqueaderos permitirá desplegar la información sobre el estado de las zonas de parqueo de manera gráfica tomando los datos de la base de datos mencionada anteriormente.

Por lo descrito anteriormente, hay que señalar que el desarrollo de este proyecto depende de la implementación de un sistema de reconocimiento de zonas de parqueo en tiempo real, este requisito alimentará a la base de datos sobre el estado (libre/ocupado) de cada espacio de parqueo y las personas podrán visualizar la disponibilidad a través de esta aplicación móvil.

Para este desarrollo se utilizarán datos de prueba, los cuales se encuentran alojados en una base de datos cuya estructura estaba basada en la que se encuentra definida dentro de los parámetros del sistema de reconocimiento mediante cámaras.

Requerimientos funcionales

Este tipo de requerimientos describen el comportamiento de la aplicación, son tomados durante la fase de análisis y serán el punto de partida para entender más allá de cumplir la necesidad del cliente que otros puntos debe abarcar la realización del proyecto.

A continuación, se detallan los requerimientos funcionales recopilados para el desarrollo:

Tabla 1

Requerimientos funcionales

Id. Requerimiento	Nombre del requerimiento	Descripción del requerimiento	Usuario
RF-01	Página de inicio de sesión.	El usuario y la clave son proporcionados por el administrador de la aplicación, no se define registro de usuarios nuevos, debido a que es una aplicación orientada a parqueaderos privados, cuyo acceso debe validarse previamente.	Guardia
RF-02	Validar datos de inicio de sesión	Se toman los datos ingresados por el usuario en los campos correspondientes, se valida que se encuentren en el formato correcto para cada caso y se compara con la información correspondiente con la base de datos, si los datos son coincidentes, se le permite el acceso	Guardia

		al resto de funcionalidades del aplicativo.	
RF-03	Formulario de registro de vehículos	Se solicita al usuario que ingrese los siguientes datos para registrar la visita de un vehículo no registrado, o que no pertenezca a la zona, llámese "visitante": Placa del vehículo, tipo (haciendo referencia a si es un auto, una moto y otro vehículo) y también se solicitará la cantidad de personas que ingresan a la zona dentro del vehículo.	Guardia
RF-04	Confirmación de ingreso de datos	Se presenta al usuario una notificación sobre la acción que está realizando para registrar los datos, esta notificación contiene una vista previa de los datos que ingresó para que los pueda validar, cuando corrobore la información se confirma el ingreso.	Guardia
RF-05	Guardar datos de registro en la base de datos	Se valida la coherencia de los datos ingresados y luego estos se envían a guardar a registrar dentro de la base de datos en su correspondiente tabla, adicionalmente, para tener un mayor	Guardia

		control, al momento de guardar en la base se guardan también los datos de hora y fecha. En el caso de que los datos no hayan pasado la debida validación, se presenta un mensaje al usuario indicando cual es el problema y que dato debe corregir.	
RF-07	Opción de cierre de sesión	Se implementará un ítem tipo menú que contenga los datos del usuario, así como también la opción disponible en cuanto a registro de usuarios o visitas, y finalmente, el botón de cerrar sesión.	Guardia
RF-06	Página de inicio de sesión.	El usuario y la clave son proporcionados por el administrador de la aplicación, no se define registro de usuarios nuevos, debido a que es una aplicación orientada a parqueaderos privados, cuyo acceso debe validarse previamente.	Conductor
RF-07	Validar datos de inicio de sesión.	Se toman los datos ingresados por el usuario en los campos correspondientes, se valida que se encuentren en el formato correcto para cada caso y se compara con la	Conductor

		información correspondiente con la base de datos, si los datos son coincidentes, se le permite el acceso al resto de funcionalidades de la aplicación.	
-	Datos del usuario y cerrar sesión.	De la validación anterior, se toman los datos del usuario para presentar en el aplicativo desde una vista de perfil, adicionalmente, se colocará dentro de este apartado una opción de cierre de sesión para que el usuario pueda controlar el acceso a la aplicación.	Conductor
RF-08	Consultar de la base la cantidad y disponibilidad de las zonas	Se realiza una consulta a la base de datos para obtener las zonas totales de parqueo, posteriormente se valida su capacidad y cuantos espacios se encuentran ocupados para calcular si la zona ya se encuentra completamente ocupada o si aún tiene disponibilidad en cuanto a espacios.	Conductor
RF-09	Página de visualización de	Se presenta al usuario las zonas disponibles para el parqueo que está visitando, presentando una pequeña	Conductor

	las zonas de parqueo	descripción acerca de su estado, ya sea completamente lleno o si aún tiene espacios disponibles.	
	Consultar de la base la disponibilidad de los espacios	Luego de seleccionar una zona de parqueo en, en la siguiente interfaz se realiza una consulta a la base con la zona seleccionada para obtener los estados de los espacios que esta contiene, y así presentar al usuario sus correspondientes indicadores (negro: ocupado, verde: disponible).	Conductor
RF-10	Diagramar la zona de parqueo	Los mapas para cada zona se realizarán mediante herramientas de edición gráfica, tratando de graficar y de escalar los tamaños de cada zona de tal forma que la representación sea un diagrama bastante similar en cuanto a forma de dicha zona del parqueadero.	Conductor
RF-11	Página de visualización de los espacios de parqueo.	Representación gráfica de la zona de parqueo con los indicadores correspondientes en cada espacio para que el usuario visualice e identifique a cuál de ellos puede acceder.	Conductor

RF-12	Consultar con la base de datos el estado de cada una de las zonas de parqueo.	Según el identificador de cada zona, se toma de la base el estado que cada uno de los espacios y se presenta al usuario para ser presentado en el mapa correspondiente.	Conductor

Nota. Requisitos funcionales del sistema obtenidos del análisis del giro del negocio. Elaborado por: Guachán, Sigcho

Requerimientos no funcionales

Tabla 2

Requerimientos no funcionales

Id. requerimiento	Descripción del requerimiento	Tipo de requisito
RNF-01	Se creará un Splash Screen o pantalla de bienvenida que el usuario visualizará previo a las pantallas principales para iniciar sesión.	Técnico
RNF-02	La aplicación se desarrollará en utilizando el Framework de Flutter y el lenguaje de programación conocido como Dart, en un editor de código.	Técnico
RNF-03	Se utilizará como dispositivo de prueba un teléfono Android, en su versión de sistema operativo 10.0.	Técnico

RNF-04	Según la cantidad de espacios de parqueo disponibles se parametrizará el número de peticiones que se debe realizar al servidor donde esta alojada la base de datos con la información de las zonas.	De rendimiento
RNF-05	Los datos de inicio de sesión serán cifrados previo a ser enviados hacia el web service.	De seguridad
	La aplicación siempre estará disponible y tendrá cobertura dentro de los límites de la zona de parqueo a la que acuda.	De fiabilidad
RNF-06	La aplicación mostrará alertas al usuario acerca de si está realizando una acción de forma correcta o incorrecta.	De usabilidad
RNF-07	La aplicación contará con un diseño sencillo y llamativo mostrando de forma clara la información más relevante para el usuario.	De usabilidad
RNF-08	La aplicación estará alojada dentro del servidor de la zona de parqueo para que, mediante un enlace directo, el usuario pueda acceder a ella y posteriormente descargarla e instalarla.	De portabilidad

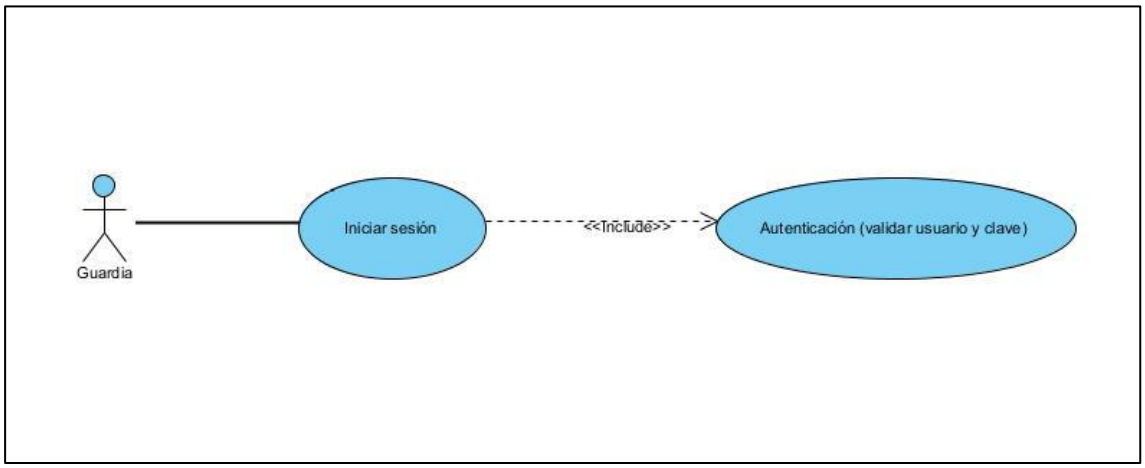
Nota. Tabla de requisitos no funcionales, son requisitos no fundamentales para el funcionamiento del sistema. Elaborado por: Guachán, Sigcho.

DISEÑO Y MODELADO DEL SISTEMA

Diagramas de casos de uso: Usuario que registra visitas

Figura 4

Caso de uso: Iniciar sesión



Nota. Proceso que se realiza para iniciar sesión en la aplicación en el que interviene un actor. Elaborado por: Guachán, Sigcho.

Tabla 3

Escenario del caso de uso Iniciar sesión

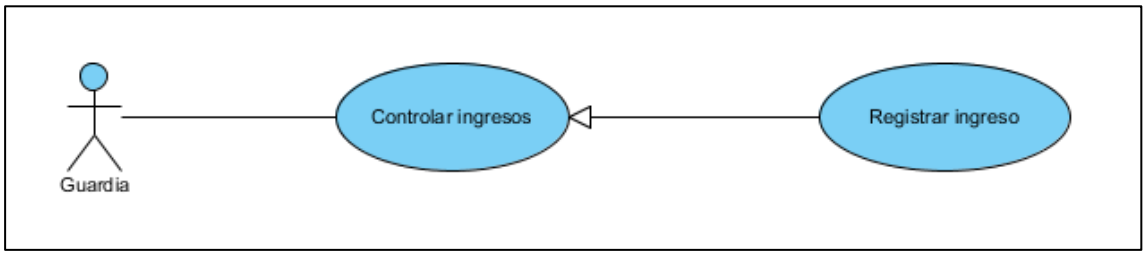
Caso de uso	**Iniciar sesión**
Actor/es	Guardia
Descripción	Proceso que sigue el inicio de sesión para cada usuario
Flujo normal	1. El usuario ingresa a la aplicación 2. Se muestra la pantalla de bienvenida durante x segundos 3. Se muestra la pantalla que contiene el formulario de inicio de sesión.

	4. El usuario ingresa su usuario y contraseña. 5. Se validan la coherencia de los datos ingresados, así como su formato. 6. La aplicación compara los datos validados con la base de datos para identificar si el usuario está registrado. 7. El usuario tiene acceso a la funcionalidad de la aplicación.
Flujo alternativo	7. La aplicación muestra al usuario mediante una alerta que no se encuentra registrado.

Nota. Descripción del caso de uso Iniciar sesión. Elaborado por: Guachán, Sigcho.

Figura 5

Caso de uso: Registrar invitado



Nota. Proceso que se realiza para registrar un usuario invitado en la aplicación. Elaborado por: Guachán, Sigcho.

Tabla 4

Escenario del caso de uso registrar invitado

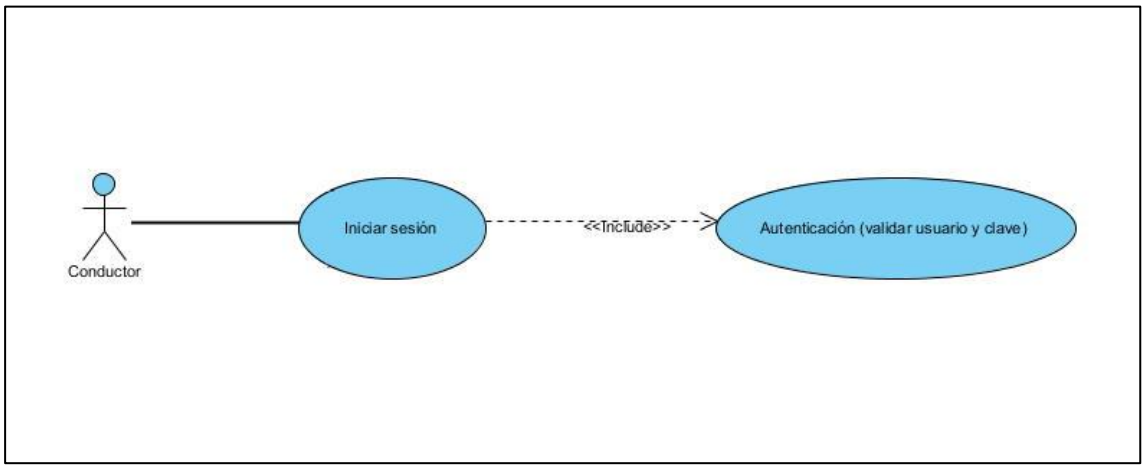
Caso de uso	**Registrar invitado**
Actor/es	Guardia
Descripción	Proceso que sigue el registrar el ingreso de un usuario invitado al parqueadero, es decir, de un conductor no recurrente en la zona.
Flujo normal	1. Una vez el usuario inicia sesión correctamente, selecciona la opción de registrar ingreso. 2. Se muestra el formulario de registro de visitas. 3. El usuario ingresa la información solicitada por el formulario. 4. La aplicación solicita confirmación por parte del usuario para registrar los datos. 5. La aplicación registra los datos del visitante. 6. La aplicación notifica el ingreso de los datos. 7. La aplicación regresa al usuario a la pantalla principal.
Flujo alternativo	5. El usuario corrige los datos correspondientes. 6. La aplicación solicita confirmación por parte del usuario para registrar los datos. 7. La aplicación registra los datos del visitante. 8. La aplicación notifica el ingreso de los datos. 9. La aplicación regresa al usuario a la pantalla principal.

Nota. Descripción del caso de uso registrar invitado. Elaborado por: Guachán, Sigcho.

Diagramas de casos de uso: Usuario conductor registrado

Figura 6

Caso de uso: Iniciar sesión.



Nota. Proceso que se realiza para iniciar sesión en la aplicación. Elaborado por: Guachán, Sigcho.

Tabla 5

Escenario del caso de uso: Iniciar sesión.

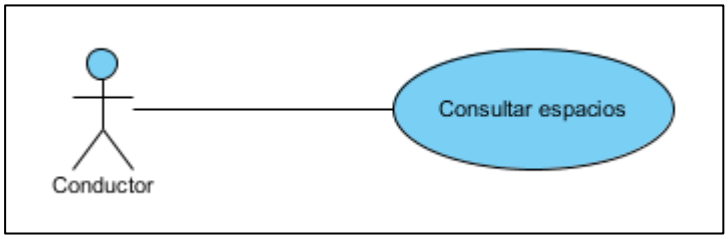
Caso de uso	**Iniciar sesión**
Actor/es	Guardia/Conductor
Descripción	Proceso que sigue el inicio de sesión para cada usuario.
Flujo normal	8. El usuario ingresa a la aplicación. 9. Se muestra la pantalla de bienvenida durante x segundos. 10. Se muestra la pantalla que contiene el formulario de inicio de sesión. 11. El usuario ingresa su usuario y contraseña. 12. Se validan la coherencia de los datos ingresados, así como su formato.

	13. La aplicación compara los datos validados con la base de datos para identificar si el usuario está registrado. 14. El usuario tiene acceso a la funcionalidad de la aplicación.
Flujo alternativo	7. La aplicación muestra al usuario mediante una alerta que no se encuentra registrado.

Nota. Descripción del caso de uso Iniciar sesión. Elaborado por: Guachán, Sigcho.

Figura 7

Caso de uso: Consultar espacios de parqueo



Nota. Proceso que se realiza para iniciar sesión en la aplicación. Elaborado por: Guachán, Sigcho.

Tabla 6

Escenario del caso de uso consultar espacios de parqueo.

Caso de uso	**Consultar espacios de parqueo**
Actor/es	Conductor
Descripción	Proceso que sigue el consultar la disponibilidad de zonas y espacios de parqueo.
Flujo normal	1. Una vez el usuario inicia sesión correctamente, la aplicación le muestra las zonas de parqueo disponibles para el lugar en cuestión, indicando parcialmente cuales de ellas están libres o llenas. 2. El usuario selecciona la zona de parqueo de su preferencia. 3. La aplicación muestra el mapa de la zona previamente seleccionada con los espacios que contiene, indicando al usuario cual está disponible y cual se encuentra ocupado. 4. El usuario se dirige al espacio de su preferencia.
Flujo alternativo	4. El usuario cambia la zona de parqueo que selecciono. 5. La aplicación muestra el mapa de la nueva zona seleccionada. 6. El usuario se dirige al espacio de su preferencia.

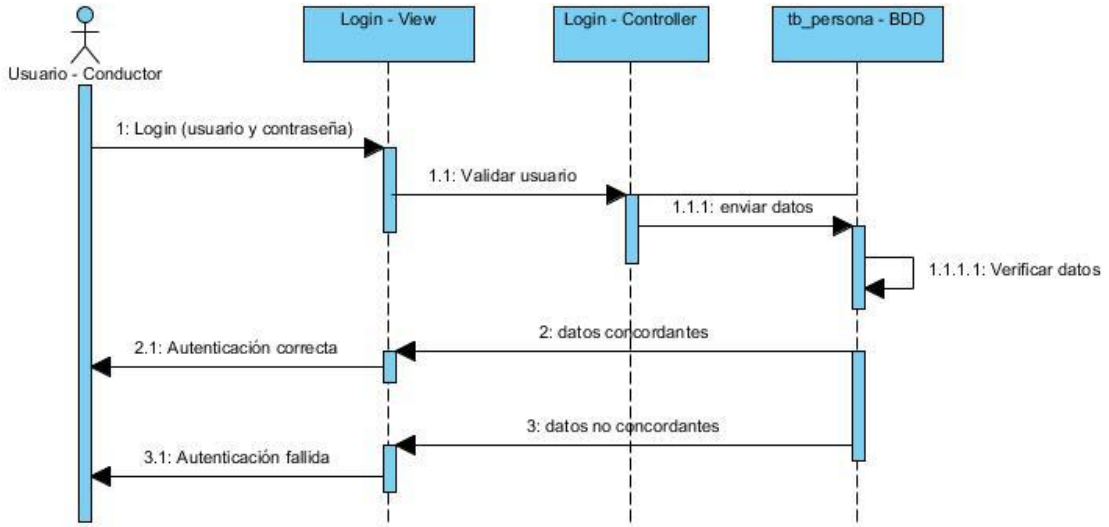
Nota. Descripción del caso de uso consultar espacios de parqueo. Elaborado por: Guachán, Sigcho.

Diagramas de secuencias

Diagrama de secuencia: iniciar sesión. En la figura 8, se visualiza la interacción entre los modelos que se realizan dentro de la aplicación para realizar el proceso de autenticación del usuario.

Figura 8

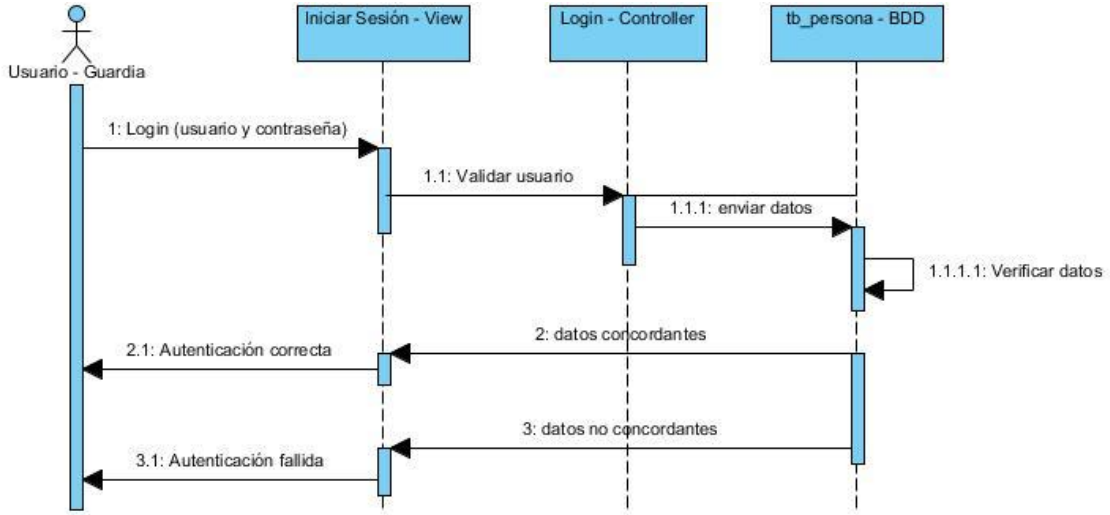
Diagrama de Secuencias: Iniciar sesión (Usuario que registra visitas)



Nota. Diagrama de secuencias que describe las interacciones entre los componentes de la aplicación para que el usuario que registra las visitas pueda iniciar sesión. Elaborado por: Guachán, Sigcho.

Figura 9

Diagrama de Secuencias: Iniciar sesión (Usuario que conductor)

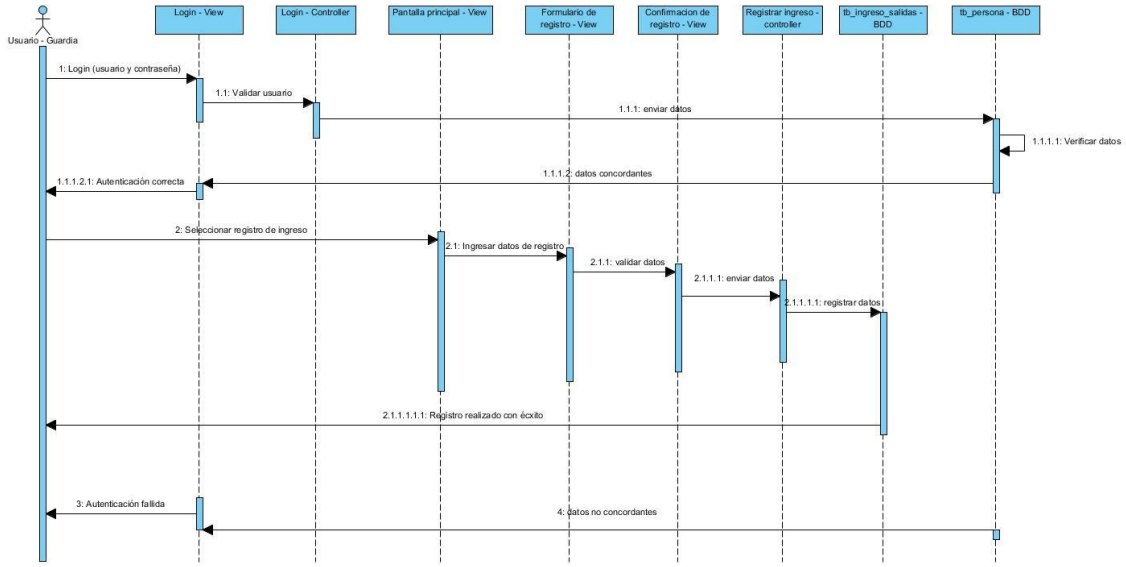


Nota. Diagrama de secuencias que describe las interacciones entre los componentes de la aplicación para que el usuario conductor pueda iniciar sesión. Elaborado por: Guachán, Sigcho.

Diagrama de secuencias: Registrar visitas. En la figura 10, se visualiza las interacciones entre los objetos dentro de la aplicación, para que el usuario con el rol de guardia pueda ingresar los datos de un vehículo visitante que ingresa a la zona de parqueo.

Figura 10

Diagrama de secuencias: Registrar visitas



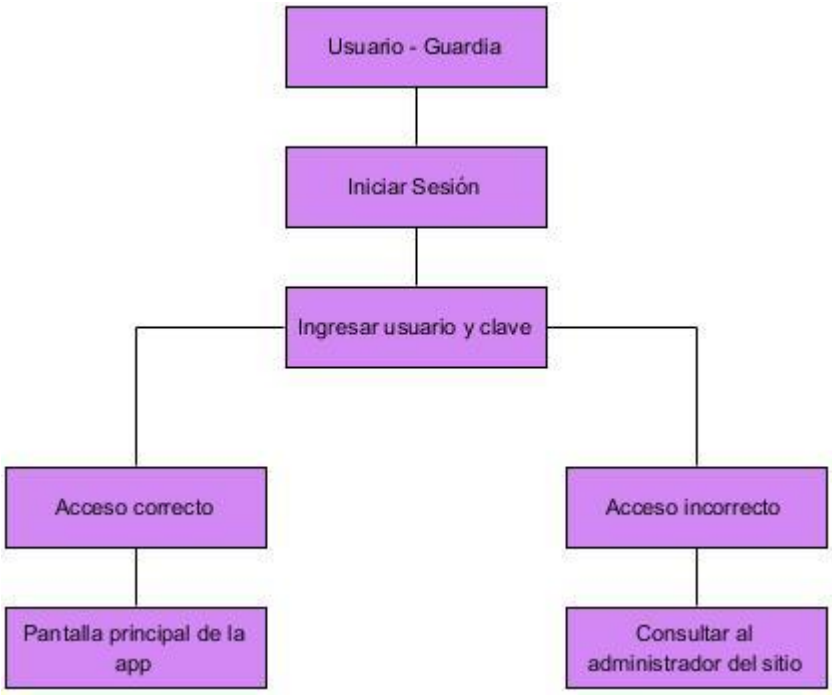
Nota. Diagrama de secuencias que describe el proceso para registrar una visita a la zona de parqueo. Elaborado por: Guachán, Sigcho.

Menú de navegación

Menú: iniciar sesión. En la figura 11, se visualiza el esquema de navegación mediante el cual se indica el proceso que sigue la aplicación para autenticar al usuario Guardia para que pueda utilizar la función de registrar las visitas.

Figura 11

Esquema de navegación: iniciar sesión (Guardia)

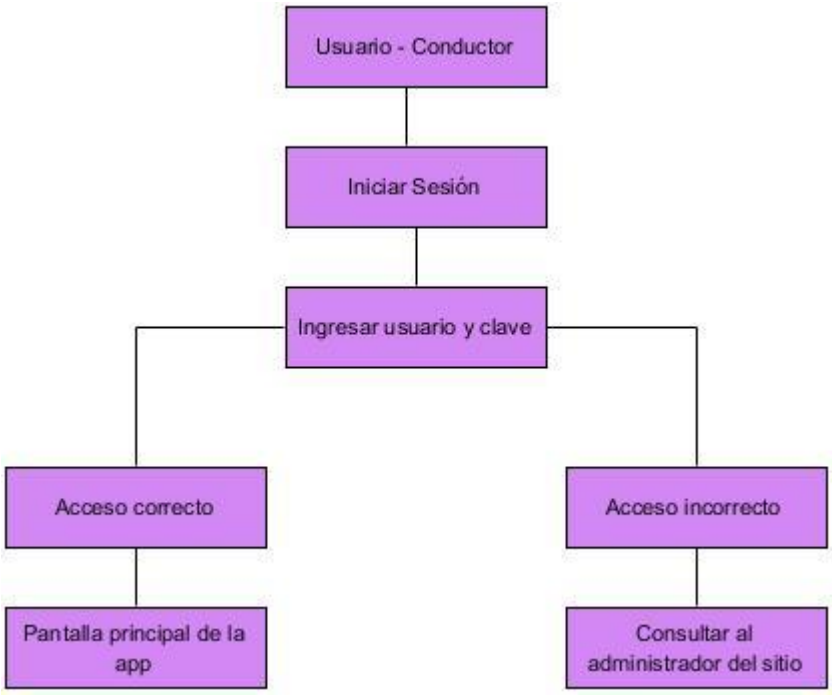


Nota. Representación visual del proceso de iniciar sesión para el rol de Guardia. Elaborado por: Guachán, Sigcho.

En la figura 12, se visualiza el esquema de navegación mediante el cual se indica el proceso que sigue la aplicación para autenticar a un usuario conductor, de tal forma que este pueda utilizar la función de consultar disponibilidad de la aplicación.

Figura 12

Esquema de navegación: iniciar sesión (Conductor)



Nota. Representación visual del proceso de iniciar sesión para el rol de conductor. Elaborado por: Guachán, Sigcho.

Menú: Registrar visita. En la figura 13, se visualiza el esquema de navegación que indica el flujo del proceso que realiza la aplicación para registrar una visita.

Figura 13

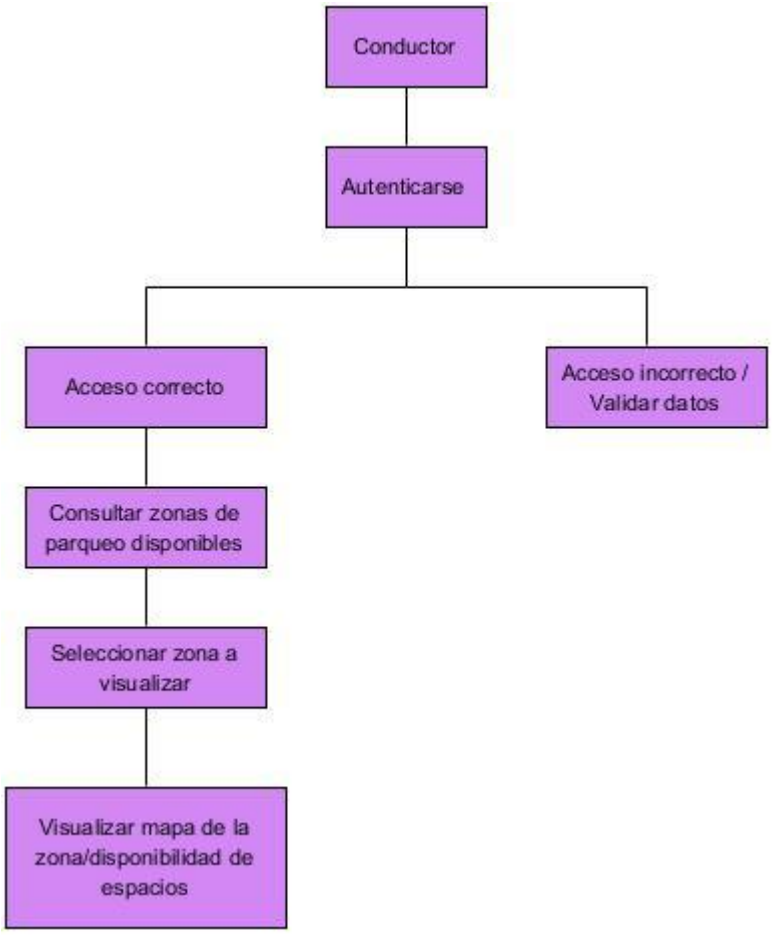
Esquema de navegación: Registrar visita

Nota. Vista del flujo que sigue el proceso de registrar de usuario. Elaborado por: Guachán, Sigcho

Menú: Consultar espacios disponibles. En la figura 14, se visualiza el flujo del proceso que sigue la aplicación para consultar las zonas disponibles dentro del parqueadero, junto con el mapa de cada zona y el estado de cada espacio.

Figura 14

Esquema de navegación: Consultar espacios disponibles



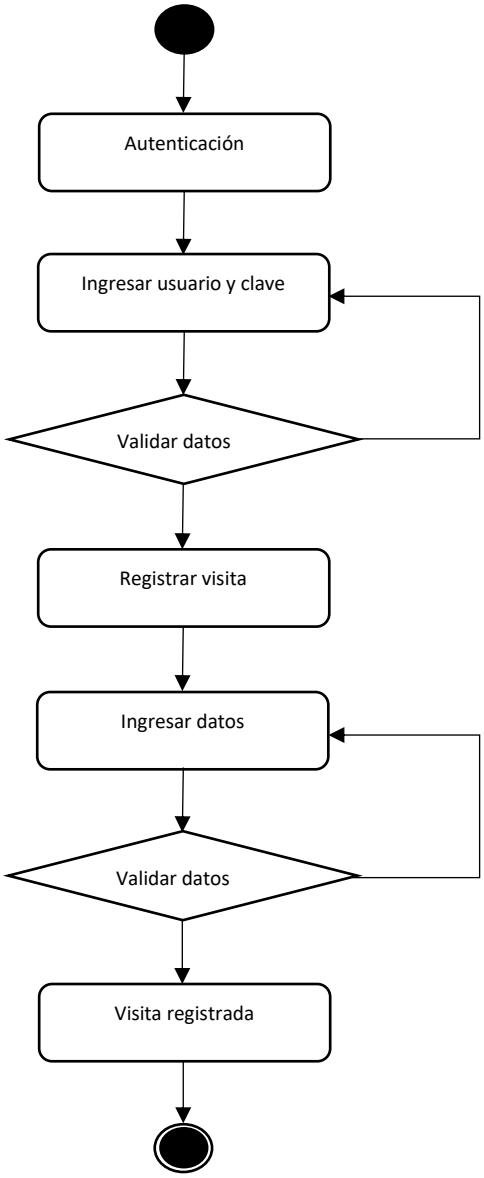
Nota. Vista del flujo que sigue el proceso de consultar zonas y espacios disponibles. Elaborado por: Guachán, Sigcho.

Diagramas de actividades

En la figura 15, se visualiza el flujograma de actividades para registrar una visita, el usuario que tiene el rol de guardia coloca sus datos para iniciar sesión, los cuales son proporcionados por el administrador del sitio, y serán validados comparando lo ingresado con los registros en la base de datos. Una vez dentro del sistema, se visualizará un formulario en el cual ingresará los datos descritos en el mismo para ser validados en una pantalla de confirmación, de ser correctos, estos datos se procesarán y se enviarán a registrar dentro de la base de datos.

Figura 15

Diagrama de actividades registro de visita: Usuario Guardia



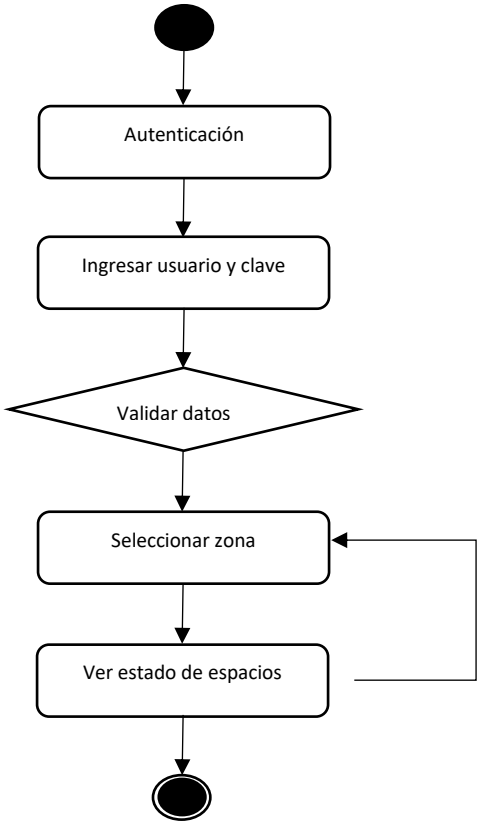
Nota. Flujograma de actividades para registrar una visita al parqueadero. Elaborado por: Guachán, Sigcho.

En la figura 16, se visualiza el flujograma de actividades que describe los pasos a seguir por el usuario conductor dentro de la aplicación para consultar la disponibilidad de espacios. Primero el usuario ingresa sus datos de acceso, los cuales son proporcionados por el administrador del

sistema, luego, se validan los datos y dependiendo del resultado se presentará un mensaje de error de inicio de sesión si no se encontraron coincidencias dentro de la base o se le da la bienvenida a la aplicación en el caso contrario. En la primera vista que presentará la aplicación se muestran las zonas de parqueo disponibles con indicadores globales que brindan la opción de visualizar la cantidad de espacios que dispone una zona y cuantos de ellos están disponibles. Dependiendo de la zona que elija, se mostrará su mapa correspondiente y los respectivos indicadores dentro de cada espacio.

Figura 16

Diagrama de actividades registro de visita: Usuario conductor



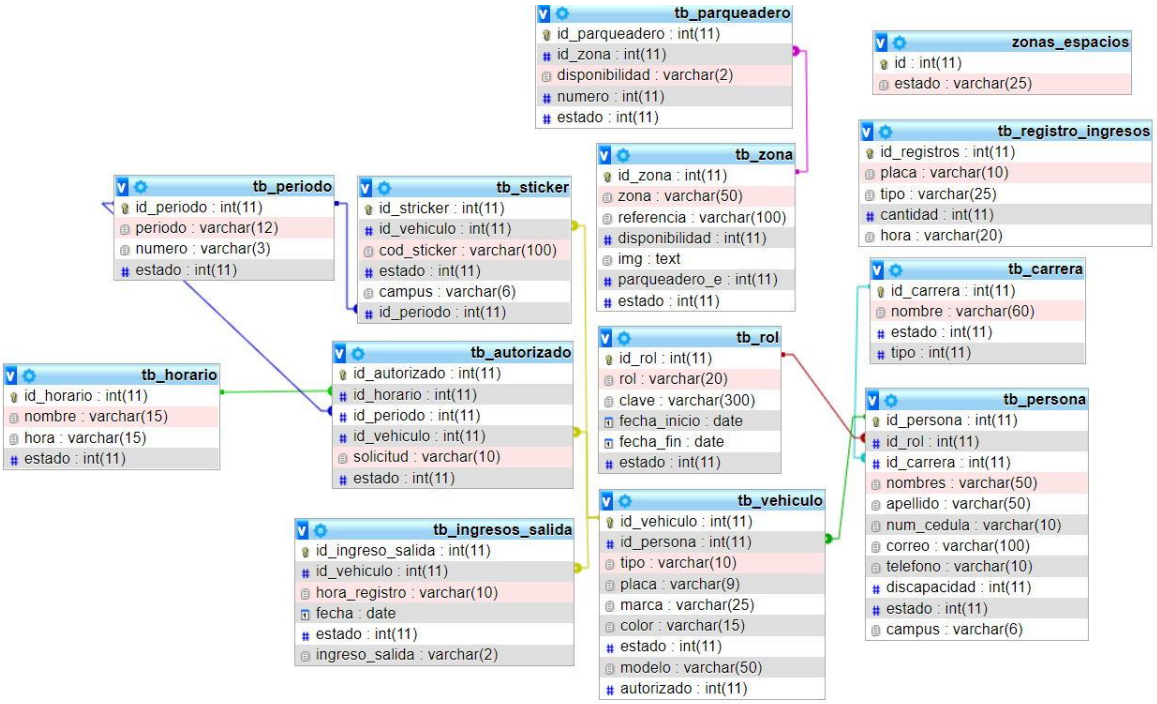
Nota. Flujograma de actividades que describen los pasos a seguir para que el usuario consulte los espacios disponibles por zonas. Elaborado por: Guachán, Sigcho.

Diagramas de clases

En la figura 17, se detallan todas las entidades que forman parte de la lógica de las dos aplicaciones, así como también, la forma en la que se relacionan cada una de estas, los datos relevantes y los tipos de dato que posee cada entidad.

Figura 17

Diagrama de clases



Nota. Diagrama de clases donde se detallan todas las entidades que intervienen dentro de la lógica del sistema. Elaborado por: Guachán, Sigcho.

DISEÑO DE INTERFACES

El diseño de la interfaz prototipo se desarrolló en Figma, una herramienta web que incluye ciertos componentes de diseño y maquetación, estos ayudan a construir las estructuras para las interfaces de usuario de manera colaborativa.

Prototipo de la interfaz del sistema: Usuario Guardia

En la figura 18, se visualiza el prototipo Splash Screen o pantalla de bienvenida para la aplicación móvil, esta será la primera vista que tendrá el usuario al abrir la aplicación.

Figura 18

Prototipo de pantalla de bienvenida a la aplicación.

Nota. Vista del prototipo de la pantalla de bienvenida de la aplicación móvil. Elaborado por: Guachán, Sigcho.

En la figura 19, se visualiza el prototipo de formulario de inicio de sesión para acceder a la aplicación móvil, el formulario recoge el correo electrónico del usuario, así como también su número de cedula que será su contraseña de acceso. Mediante esta vista se envían los datos de acceso cifrados para validar y comparar con los registros de la base de datos.

Figura 19

Prototipo de formulario de inicio de sesión.

Nota. Vista del prototipo de formulario de inicio de sesión para la aplicación móvil. Elaborado por: Guachán, Sigcho.

En la figura 20, se visualiza el prototipo del formulario para el registro de ingreso para visitantes con los campos requeridos, en este caso se solicita al usuario ingresar la placa del vehículo que ingresa como visitante, así como también la cantidad de ocupantes que lo acompañan y el tipo de vehículo al que corresponde el registro.

Figura 20

Prototipo de formulario de registro para vehículo visitante.

Nota: Vista del prototipo de la pantalla de registro de ingresos a las zonas de parqueo. Elaborado por: Guachán, Sigcho.

En la figura 21, se visualiza el prototipo de la pantalla de confirmación para registrar los datos de la visita, esta vista permite al usuario verificar si los datos que se van a registrar dentro de la base de datos son correctos haciendo una validación de lo que se está ingresando, todo esto previo a registrar permanentemente los datos dado que los registros no van a poder editarse una vez se guarden.

Figura 21

Prototipo de pantalla de validación para confirmar los datos a registrar.

Nota: Vista prototipo de la pantalla de validación de datos ingresados. Elaborado por: Guachán, Sigcho

En la figura 22, se visualiza el prototipo de pantalla en la que se muestra un menú con los datos del usuario y la única acción disponible que este puede realizar. Tomando en cuenta que, para el rol de usuario que registra visitas será solamente el formulario antes mencionado para registrar los datos solicitados.

Figura 22

Prototipo de información del usuario que está utilizando la aplicación y cierre de sesión.

Nota: Vista prototipo del menú de la aplicación móvil. Elaborado por: Guachán, Sigcho.

Prototipo de la interfaz del sistema: Usuario Conductor

En la figura 23, se visualiza el prototipo Splash Screen que mostrará a el usuario conductor, será la primera vista que se presentará al abrir la aplicación, una pantalla de bienvenida para el usuario.

Figura 23

Prototipo pantalla de bienvenida a la aplicación.

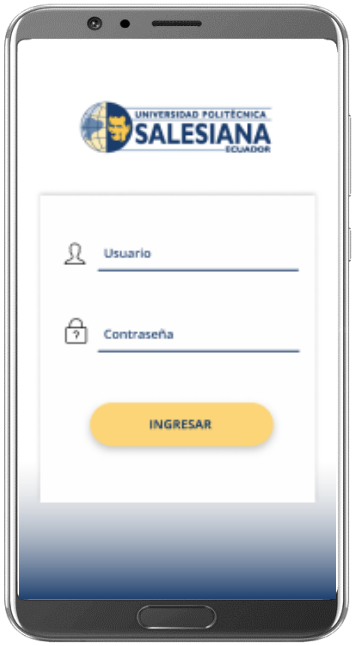
Nota: Vista prototipo de la pantalla de bienvenida que será mostrada al usuario conductor.

Elaborado por: Guachán, Sigcho

En la figura 24, se visualiza el prototipo de formulario de inicio de sesión para acceder a la aplicación móvil el cual recoge el correo electrónico del usuario, así como también su número de cédula que será su contraseña de acceso, vista mediante la cual se envían los datos de acceso cifrados a validar y comprar con los registros de la base de datos.

Figura 24

Prototipo de formulario de inicio de sesión



Nota: Vista prototipo del formulario para iniciar sesión dentro de la aplicación. Elaborado por: Guachán, Sigcho.

En la figura 25, se visualiza el prototipo de la pantalla que contiene todas las zonas de parqueo disponibles dentro del estacionamiento, para que el usuario pueda comprobar la disponibilidad dentro de la misma. Estos valores se obtendrán de la base de datos contando los registros por zona de parqueo totales y los que tengan el estado ocupado para presentar la cantidad restante, esto servirá como una vista previa para el usuario y que de este modo pueda anticipar que zona tiene espacios y en qué cantidad hay disponibilidad.

Figura 25

Prototipo de zonas de parqueo disponibles dentro del lugar

Nota: Vista prototipo de la pantalla para seleccionar la zona de parqueo a consultar. Elaborado por: Guachán, Sigcho.

En la figura 26, se visualiza el prototipo de la pantalla que muestra la zona de parqueo previamente seleccionada con los indicadores en cada espacio para representar el estado de cada uno, en este caso “Libre” y “Ocupado”. Estos datos se obtendrán de la base de datos, donde se valida la zona a mostrar, los espacios y el estado de cada uno de los registros que permitirán identificarlo.

Figura 26

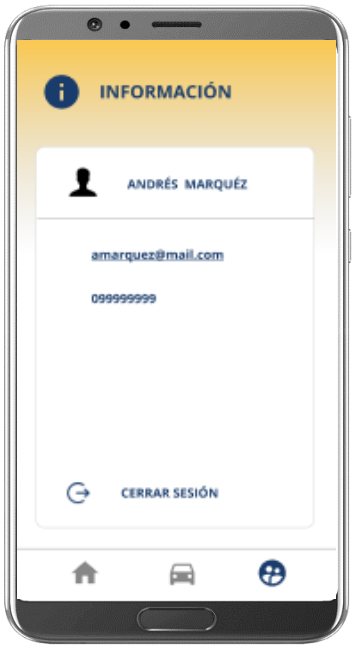
Prototipo de mapa de la zona de parqueo seleccionada.

Nota: Vista prototipo de la zona de parqueo con el estatus dentro de cada espacio. Elaborado por: Guachán, Sigcho.

En la figura 27, se visualiza el prototipo de la pantalla con los datos del usuario que inició sesión dentro de la aplicación móvil los cuales se obtuvieron al momento de validar el ingreso a la aplicación comparando los datos que se enviaron desde el formulario de inicio de sesión con la tabla de usuarios registrados dentro de la base de datos.

Figura 27

Prototipo de perfil con los datos del usuario que inicio sesión



Nota. Vista prototipo del perfil de usuario registrado dentro de la aplicación móvil. Elaborado por: Guachán, Sigcho

CAPÍTULO IV

DESARROLLO DEL SISTEMA

Métodos para gestionar la conexión con la base de datos:

Para la gestión de los datos que se utilizan dentro de la aplicación mediante la conexión a un servidor remoto, para lo cual, en este caso se hace uso de cuatro archivos en formato PHP que albergan los métodos para: realizar la conexión con la base de datos, validar el inicio de sesión de los usuarios, guardar los datos de los vehículos que ingresan a la zona de parqueo y consultar todos los espacios disponibles y las zonas de parqueo que se encuentran registradas en una base de datos de pruebas.

Conexión a la base de datos

En la siguiente sección de pseudocódigo, se puede observar el método para la conexión de la base de datos, en el mismo proceso se crean las operaciones para el control de errores en caso de que la conexión no sea exitosa.

```
Proceso Conexión

        Definir host,dbname,dbuser,dbpass como Caracter;
        Definir conn como logico;
        host<-'localhost';
        dbname<-'test_db';
        dbuser<-'test_user';
        dbpass<-'test_pwd';
        conn <-
ConexionProcess(host=host,dbname=dbname,dbuser=dbuser,dbpass=dbpass);
        Si conn Entonces
                retornar conn;
        SiNo
                Escribir 'Error en la conexión';
        FinSi

FinProceso
```

Validar el inicio de sesión de los usuarios:

En la siguiente sección de pseudocódigo, se puede apreciar el algoritmo que permite validar el inicio de sesión de los usuarios, cruzando los datos que el usuario proporciona en el formulario de ingreso, con los que se encuentran registrados en la base de datos. Primeramente, se hace uso del proceso definido en el punto anterior para obtener dentro de este nuevo algoritmo el acceso a la base de datos, luego se recogen los datos que llegan desde la aplicación en una variable y se decodifica ya que se hace uso de JSON un lenguaje usado para transferencia de datos.

Desde la aplicación se invoca un método de encriptación el cual permite enviar la información de inicio de sesión cifrada así que, el siguiente paso es crear un método que permita desencriptar los datos que recibe el servidor, este recibirá dos cadenas de texto, que serán usuario y contraseña, el método de encriptación, en este caso 'aes-128-ecb', y una clave de encriptación, la cual debe ser la misma que se utiliza en la aplicación para que la conversión sea correcta. Con los datos obtenidos, usuario y clave, se crea una consulta para la tabla personas en busca del usuario y clave proporcionados y el resultado de la consulta se envía hacia la aplicación, dependiendo de esto se concederá o denegará el ingreso a las funcionalidades de la aplicación móvil.

```
SubProceso output <- decrypt (cadena)

        Definir output Como Logico;
        Definir encrypt_method, key Como Caracter;
        output <- Falso;
        encrypt_method <- 'aes-128-ecb';
        key <-'clave_test';
        output = openssl_decrypt(cadena, encrypt_method, key);

    retornar output;

FinSubproceso
```

```
Proceso Ingreso

	Definir sql Como Carácter;
	Definir consulta Como Logico;
	Definir var Como EstructuraDeDatos;

	var <- ObtenerDatos();

	sql <- "Select * from tb_persona where correo= '".decrypt($var['username']).'" And
num_cedula='".decrypt($var['password'])."'";

	consulta <- EjecutarConsulta(sql);
	Si consulta Entonces
		Retornar Verdadero;
	SiNo
		Retornar Falso;
	FinSi;

FinProceso
```

Registrar datos de visitantes

En la siguiente sección de código se puede apreciar la funcionalidad que se desarrolló para insertar los datos del visitante que el usuario ingresa en la aplicación. De igual manera se hace uso del algoritmo para la conexión de la base de datos, posteriormente se captura los datos que llegan desde la aplicación móvil y al igual que en el anterior proceso se hace una conversión de tipo de datos ya que los datos viajan en un formato JSON.

Se define una consulta en lenguaje SQL para insertar los datos en la base de datos dentro de la tabla llamada “tb_registro_ingresos”, esta ejecución retorna una respuesta la cual genera una alerta en la aplicación móvil para indicar que el registro de los datos fue exitoso. El proceso se lo representa mediante el siguiente bloque de código.

```
Proceso registrar_ingreso

        Definir var Como EstructuraDeDatos;
        Definir SQL Como Caracter;
        Definir data_ingresada Como Logico;
        var <- ObtenerDatos();
        SQL = "INSERT INTO tb_registro_ingresos(placa, tipo, cantidad,
hora) VALUES ('".var['plate']."','".var['type']."',".var['quantity'].",'".var['time']."')";
    data_ingresada <- EjecutarConsulta(SQL);
    retornar data_ingresada

FinProceso
```

Consulta de zonas de parqueo y la disponibilidad de los espacios

En la siguiente sección de código se puede apreciar el proceso mediante el cual se envía a la aplicación móvil una estructura de datos, la cual contiene el identificador de la zona de parqueo y el estado del espacio asociado a dicha zona, con estos datos se consigue representar los vehículos dentro de la aplicación móvil.

Métodos implementados en la aplicación móvil

La siguiente descripción presenta el código utilizado mediante el cual la aplicación se comunica con la base de datos, además también se muestra cómo se realizaron los algoritmos para presentar las zonas con su correspondiente representación de los espacios disponibles.

```
    Proceso registrar_ingreso
        Definir sql Como Caracter;
        Definir data_parqueadero Como Logico;
        sql <- "Select * from tb_parqueadero";
        data_parqueadero <- EjecutarConsulta(sql);
        retornar data_parqueadero
    FinProceso
```

Método para validar el inicio de sesión de los usuarios:

Esta sección de código se usará del lado de la aplicación, tendrá el funcionamiento de establecer la conexión con la base de datos para enviar los datos de acceso del usuario encriptados hacia el servidor, la respuesta que se recibe por parte del servidor será evaluada para indicar si es un usuario y contraseña válido dentro del registro de datos.

```
Proceso AppIngreso
    Definir uri, encrypted_username, encrypted_password Como Carácter;
    Definir body Como EstructuraDatos;
    Definir response Como Logico;
    uri <- 'https://enlaceingreso//';
    encrypted_username = encrypt(username);
    encrypted_password = encrypt(password);
    body['username'] = encrypted_username;
    body['password'] = encrypted_password;
    response = post(uri,body = body);
    Si response Entonces
        Retornar Verdadero;
    SiNo
        Retornar Falso;
    FinSi
FinProceso
```

Obtener las zonas de parqueo con sus respectivos estados para cada espacio

Debido a la extensión de código utilizado en este apartado se mencionan los métodos o secciones más relevantes o que se utilizan más recurrentemente dentro de esta sección. Como se mencionó anteriormente, Flutter en su estructura hace uso de Widgets, para entenderlo mejor sería como definir un objeto dentro la programación orientada a objetos, estos widgets tienen distintos tipos y fines, por ejemplo, el widget Column permite definir un contenedor el cual tiene ciertas propiedades como dar estilos a sí mismo, contener otros widgets, definir márgenes entre otros.

Otro widget que se usa es Positioned, mediante el cual se puede dar posicionamiento a los objetos que se definan en su interior, mediante este widget se establece dónde debe ir un indicador dentro de la zona seleccionada ya que la solución más optima que se aplicó para resolver el problema de los indicadores fue trazar la zona mediante un programa de diseño exportarla como imagen y colocarla a la par con las posiciones mediante un Widget conocido como Stack, este permite superponer una imagen sobre otra, luego con el de posicionamiento se establece donde irá cada una.

Con este antecedente se empieza definiendo el método mediante el cual se obtienen los datos de las zonas y su disponibilidad desde la base de datos, esta sección de código se definirá dentro de un widget de tipo Future, en el cual se indica la url que se debe consultar la cual envía de regreso los datos que encontró en la base dentro de la tabla de parqueo como ya se mencionó anteriormente. Esto envía en formato JSON los datos de la zona a la que pertenece el parqueo y su estado. El tipo Future permite mantener los datos actualizados en la aplicación obteniendo los cambios realizados en la base de datos, así se obtiene un estado en tiempo real de cada zona.

```
Proceso obtenerStatusZona
        Definir uri Como Carácter;
        uri <- 'https://enlaceobtenerdatos//';
        response = get(uri);
        Retornar response;
FinProceso
```

Teniendo los valores de las zonas en un arreglo, se irá recorriendo mediante un bucle for, este bucle contendrá un widget de tipo Positioned, para cada propiedad de posicionamiento (sean: bottom, left, right, top) se definirá un método con el cual, mediante el índice del espacio de parqueo se irá calculando la posición que ocupará, además también se utiliza el Widget Transform.rotate para definir ángulos de giro y acoplar mejor los indicadores de disponibilidad, los cuales son imágenes cuyos nombres son iguales que su estado.

Para explicar como funcionan los métodos que calculan las posiciones, se toma como ejemplo la lógica de los que se pueden encontrar dentro de la primera zona, ya que es la más pequeña espacialmente y su atributo de forma no requiere muchas validaciones para poder hacer el posicionamiento de los indicadores. De tal forma que, la funcionalidad tiene la siguiente estructura.

- Se implementa un bucle for para listar los registros que forman parte de la zona 1.
- Se utiliza el incremento del mismo bucle para establecer una separación secuencial y uniforme mientras se van creando cada uno de los componentes, así se garantiza que entre cada imagen de un vehículo haya siempre 10 unidades (o pixeles) de distancia.
- Dependiendo del lugar que ocupen dentro del mapa, se deben definir puntos de quiebre para que a partir de estos cambie el comportamiento de los cálculos, así girar los

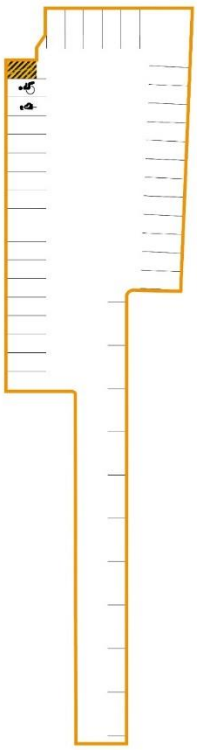
componentes, o establecer un incremento en los valores que posicionan los elementos lateralmente, esto para representar inclinaciones en la forma de la zona de parqueo.

Este procedimiento se realiza por cada zona de parqueo establecida, y la extensión de los cálculos que contiene cada método puede variar según la cantidad de espacios de estacionamiento y la forma que tenga la zona, es decir, si contiene una gran cantidad de espacios y la forma del contorno cambia en aspectos como inclinación, bordes o curvas habrá más puntos de quiebre, esto con el fin de adaptar el indicador a su espacio para ubicarlo donde corresponde según el mapa diagramado de la zona.

En la figura 28, se puede apreciar el trazo de la primera zona de parqueo que se identificó dentro del entorno de pruebas.

Figura 28

Trazo de la zona de parqueo número 1

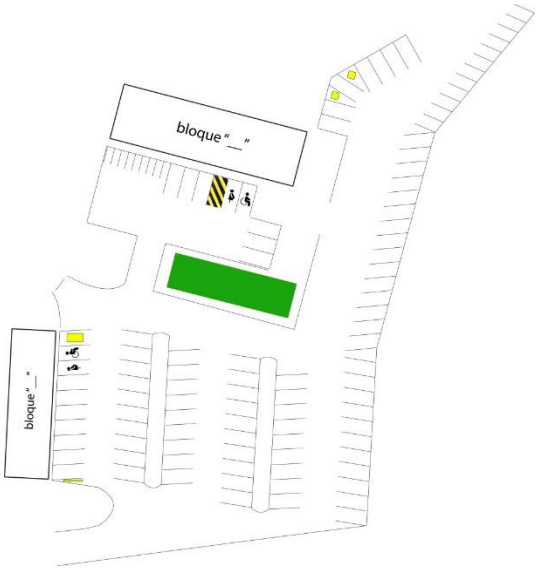


Nota. Zona de parqueo número 1. Elaborado por: Guachán, Sigcho.

En la figura 29, se puede apreciar el trazo de la segunda zona de parqueo que se identificó dentro del entorno de pruebas.

Figura 29

Trazo de la zona de parqueo número 2

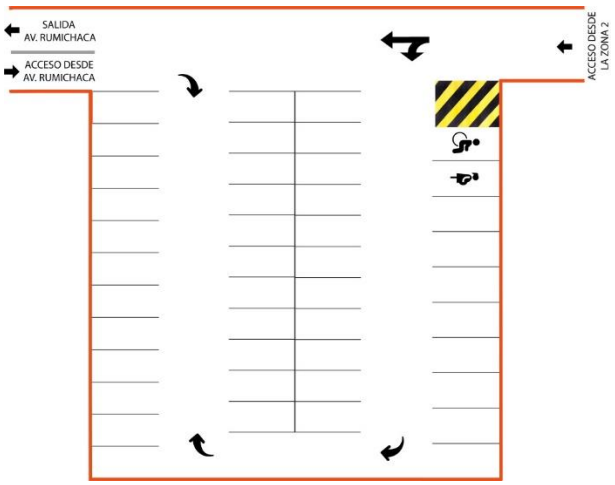


Nota. Zona de parqueo número 2. Elaborado por: Guachán, Sigcho.

En la figura 30, se puede apreciar el trazo de la tercera zona de parqueo que se identificó dentro del entorno de pruebas.

Figura 30

Trazo de la zona de parqueo numero 3

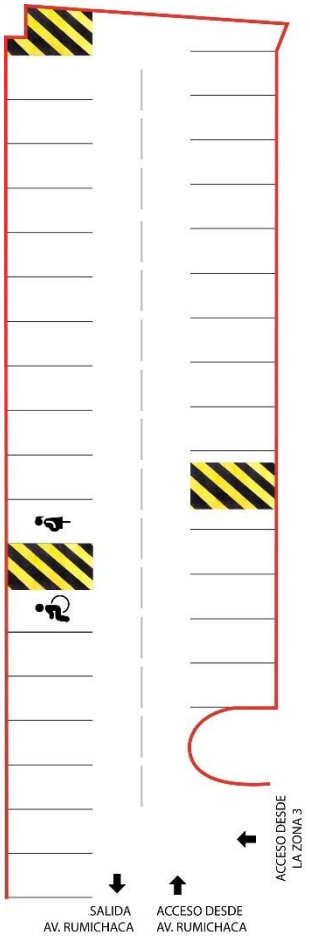


Nota. Zona de parqueo número 3. Elaborado por: Guachán, Sigcho.

En la figura 31, se puede apreciar el trazo de la cuarta zona de parqueo que se identificó dentro del entorno de pruebas.

Figura 31

Trazo de la zona de parqueo número 4



Nota. Zona de parqueo número 4. Elaborado por: Guachán, Sigcho.

En la figura 32, se puede apreciar el ícono del indicador de parqueo para el estado de "disponible" dentro de su espacio y zona.

Figura 32

Icono indicador de disponibilidad – Estado "disponible"

Nota. Ícono indicador de espacio disponible. Elaborado por: Guachán, Sigcho.

En la figura 33, se puede apreciar el ícono del indicador de parqueo para el estado de "ocupado" dentro de su espacio y zona.

Figura 33

Ícono indicador de disponibilidad – Estado "ocupado"

Nota. Ícono indicador de espacio disponible. Elaborado por: Guachán, Sigcho.

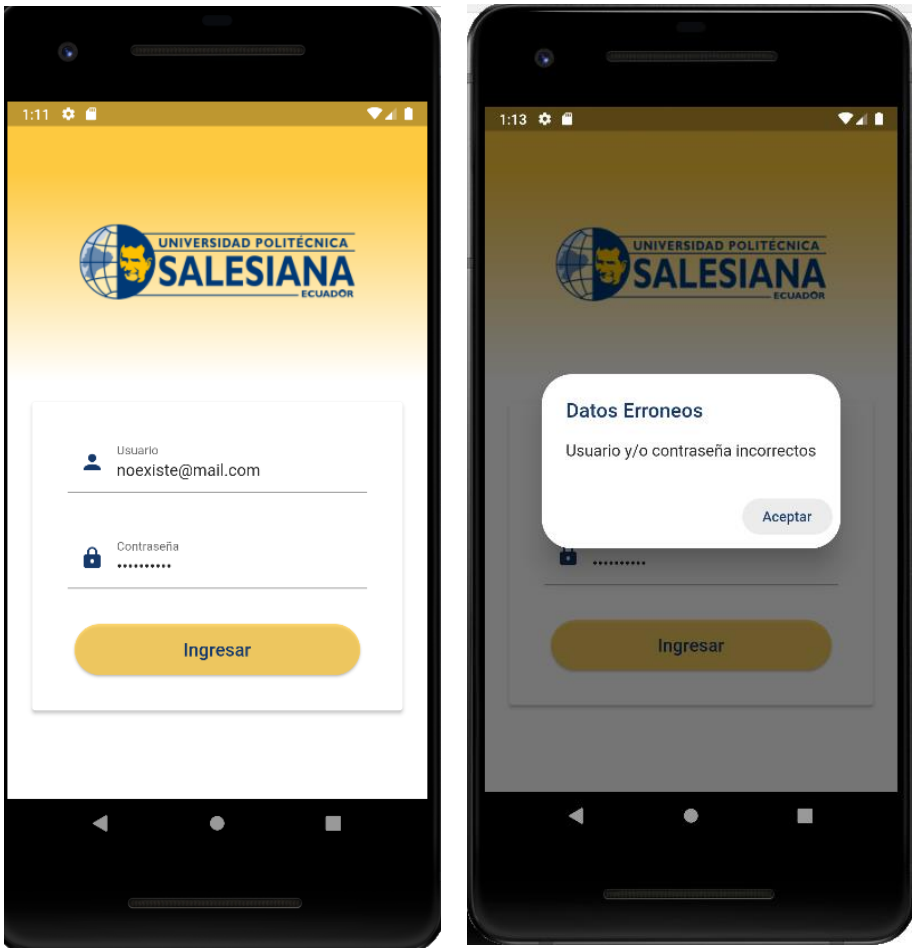
PRUEBAS DEL SISTEMA

Aplicación para registrar visitas

En la Figura 34 se comprueba que el formulario de ingreso cumpla con la función de validador de los datos que el usuario está intentando ingresar en este formulario, se capturan los datos de correo y contraseña, estos son cifrados y enviados hacia el servidor para ser comparados con los registros de la base de datos, al no encontrar datos coincidentes este presenta el mensaje de error que se puede apreciar en la presente figura.

Figura 34

Prueba de validación de datos incorrectos – vista de ingreso.

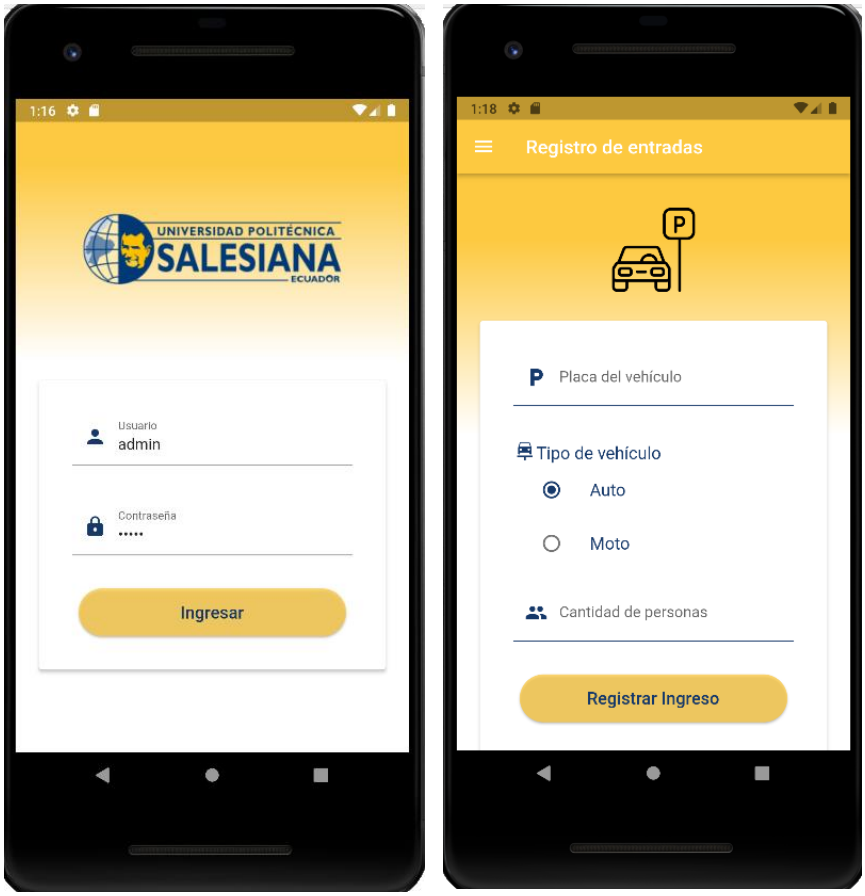


Nota. Prueba de funcionalidad para el formulario de inicio de sesión en la aplicación de registro de visitas (caso erróneo). Elaborado por Guachán, Sigcho

En la figura 35, se realiza una comprobación del formulario de inicio de sesión nuevamente, en este caso se hace una prueba con un usuario que, si se encuentra registrado dentro de la base de datos, funcionalmente el procedimiento para realizar la comprobación es el mismo descrito en el punto anterior, pero en este caso se puede apreciar que el acceso es exitoso y el siguiente paso es mostrar al usuario la vista con el formulario para que realizar el registro de la visita a la zona de parqueo.

Figura 35

Prueba de validación de datos incorrectos – vista de ingreso.

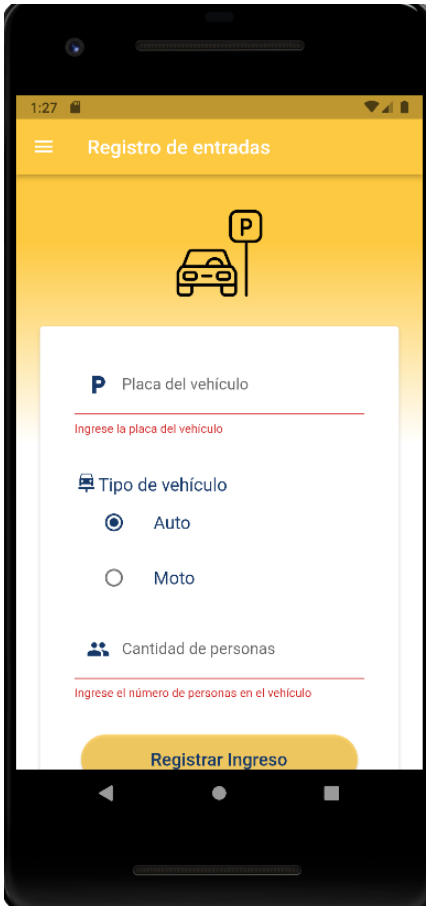


Nota. Prueba de funcionalidad para el formulario de inicio de sesión en la aplicación de registro de visitas (caso valido). Elaborado por Guachán, Sigcho.

En la figura 36, se puede apreciar la validación de los campos dentro del formulario de registro de visitas, se puede observar que en esta vista si el usuario envía los campos vacíos estos presentarán mensajes de error informativos para que el usuario tenga en cuenta que la funcionalidad de registro de visitas requiere que los 3 campos se encuentren llenos para poder avanzar hacia la vista de confirmación.

Figura 36

Prueba de validación de campos requeridos en formulario de registro

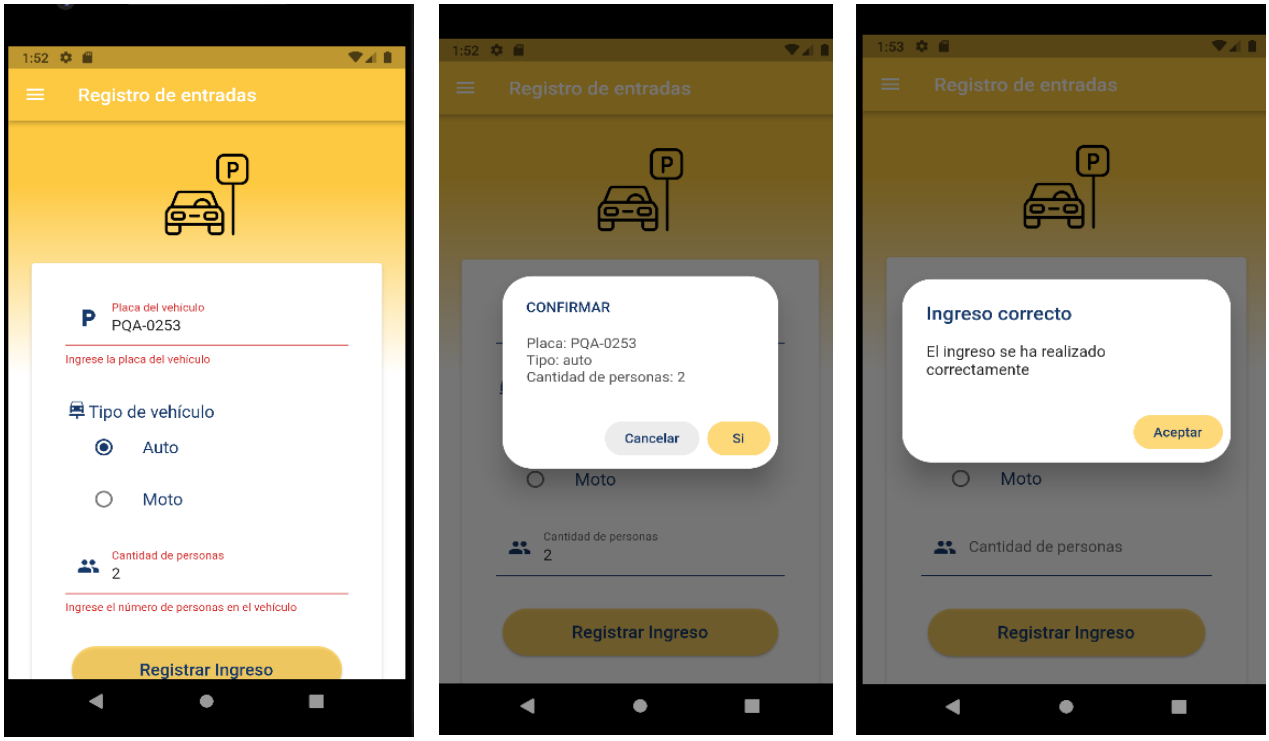


Nota. Prueba de funcionalidad para el formulario de registro de visitas para el caso en el que se intente registrar un ingreso con los campos requeridos vacíos. Elaborado por Guachán, Sigcho

En la figura 37, se evidencia el proceso para realizar un registro exitoso de una visita dentro de la base de datos, el usuario llena los campos solicitados en el formulario de registro, al presionar el botón de registrar ingreso se le presenta una vista que le solicita confirmar si los datos que va a guardar son correctos mostrándoselos nuevamente y una vez que este valida y confirma los datos son enviados a la base de datos y el usuario obtiene el mensaje correspondiente de registro exitoso.

Figura 37

Prueba de validación de registro de datos correctos



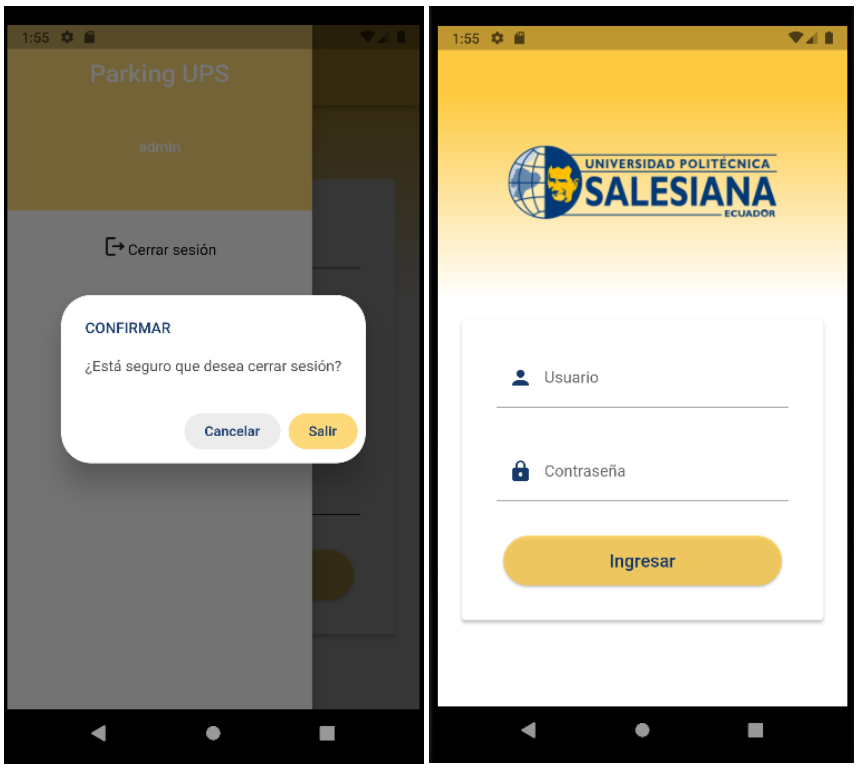
Nota. Prueba de funcionalidad para el registro de una visita, en este caso se presenta el proceso que sigue la aplicación cuando el registro de datos y la validación es correcta.

Elaborado por Guachán, Sigcho

En la figura 38, se puede apreciar el funcionamiento de la opción que le permite al usuario cerrar sesión dentro de la aplicación en el caso de que ya no desee utilizarla o si se da la situación de que el dispositivo sea de uso compartido otro usuario pueda iniciar sesión con sus datos para continuar con el proceso de registro.

Figura 38

Validación de cierre de sesión del usuario



Nota. Prueba de funcionalidad sobre la opción de finalizar la sesión del usuario que está utilizando la aplicación. Elaborado por Guachán, Sigcho

Tabla 7

Matriz de registro de pruebas de funcionalidad: Aplicación de registro de visitas

Id	Caso de Prueba	Descripción	Funcionalidad / Característica	Datos / Acciones de Entrada	Numero de pruebas / dispositivo	Resultado Esperado	**Resultado Obtenido**	**Estado**
1	Ingreso con credenciales incorrectas	Se realiza un ingreso con usuario y contraseña incorrectos, que no existen en la base de datos	Login	usuario: noexiste@mail.com contraseña: prueba	50 pruebas en 2 dispositivos	Mensaje de error	Resultado correcto. Mensaje: Usuario y/o contraseña incorrectos	Exitoso
2	Ingreso con credenciales correctas	Se realiza un ingreso con usuario y contraseña correctos	Login	usuario: admin contraseña: admin	50 pruebas en 2 dispositivos	Ingreso correcto	Resultado correcto: Muestra la pantalla principal	Exitoso
3	Registro de ingreso de vehículo	Se realizo un registro de ingreso con datos faltantes	Registro de vehículo	Placa del vehículo: Tipo de vehículo: Auto Número de personas: 3	50 pruebas en 2 dispositivos	Mensaje de datos faltantes	Mensaje: Ingrese la placa del vehículo	Exitoso
4	Registro de vehículo exitoso	Se realiza un ingreso de vehículo con los datos completos	Registro de vehículo	Placa del vehículo: PQA-0253 Tipo de vehículo: Auto Número de personas: 2	50 pruebas en 2 dispositivos	Mensaje de aprobación y registro exitoso	El sistema pidió la confirmación y el registro fue exitoso	Exitoso
5	Cierre de sesión	Se toca en el botón de cierre de sesión	Cierre de Sesión		50 pruebas en 2 dispositivos	Mensaje de confirmación de cierre de sesión	Mensaje de confirmación de cierre de sesión y salida exitosa	Exitoso

Nota. Registro de pruebas funcionales para la aplicación de registro de visitas que detalla las pruebas que se realizaron, cantidad, datos ingresas y los datos obtenidos como respuesta. Elaborado por Guachán, Sigcho.

Pruebas de carga y estrés sobre el host del web service

En las figuras 39, 40 y 41, se pueden visualizar los resultados de las pruebas de carga realizadas en la url para obtener los espacios disponibles de las zonas de parqueo. Para las siguientes pruebas se hizo uso de un grupo de 100 usuarios en un periodo de cinco segundos, esta prueba se la realizó en dos ocasiones obteniendo los mismos resultados.

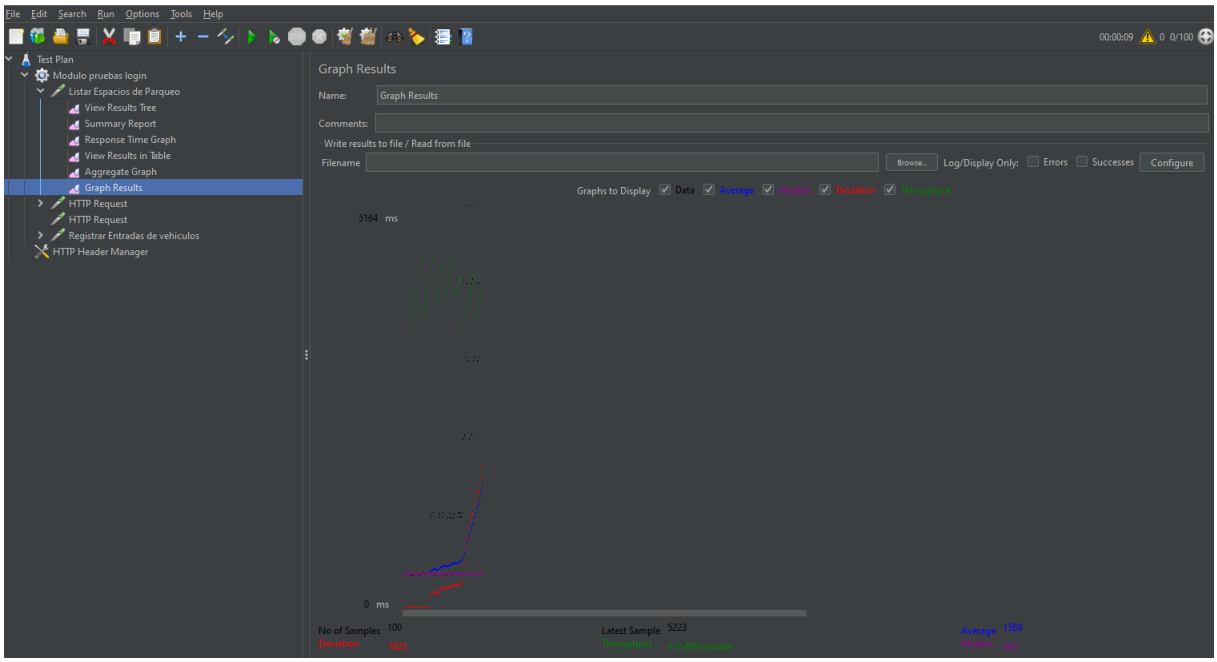
Como resultados, se puede observar que, con 100 usuarios intentando entrar en un intervalo de cinco segundos el sistema funcionará sin ningún problema y este brindará respuestas rápidas de como mínimo medio segundo de espera, esto es un excelente tiempo de carga, según estadísticas de Google el 50% de usuarios esperan que los datos carguen en menos de dos segundos.

Se puede observar en las figuras 42, 43 y 44 las pruebas realizadas en la url para el ingreso al sistema, en el que las muestras fueron 100 en un tiempo de cinco segundos, en estas pruebas el tiempo mínimo fue de 109 milisegundos y una media de 842 milisegundos, estas pruebas se ubican en un rango aceptable de tiempos de respuesta.

Para las pruebas realizadas en los ingresos de datos de vehículos que se muestran en las figuras 45, 46, 47 se usó la misma muestra de usuarios, 100 usuarios en cinco segundos y los resultados obtenidos fueron los siguientes, un promedio de respuesta de 1103 milisegundos, una media de 846 milisegundos y un tiempo mínimo de 370 milisegundos tomando en cuenta que se realizan ingresos en la base de datos, aun así, se encuentra en un tiempo de respuesta aceptable.

Figura 39

Resultado de la prueba de rendimiento para el web service de listar espacios (gráfica).

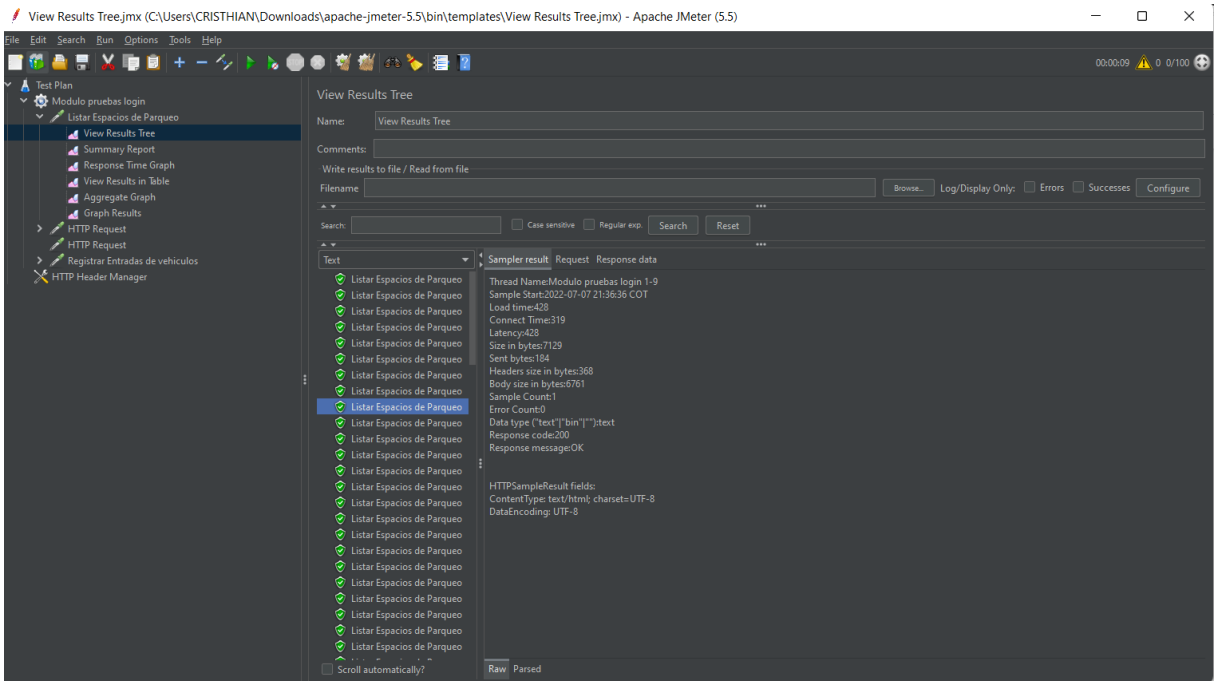


Nota. Gráfico que indica los resultados de las peticiones que realiza la aplicación de pruebas con los parámetros previamente indicados obteniendo una respuesta de 442 milisegundos.

Elaborado por Guachán, Sigcho.

Figura 40

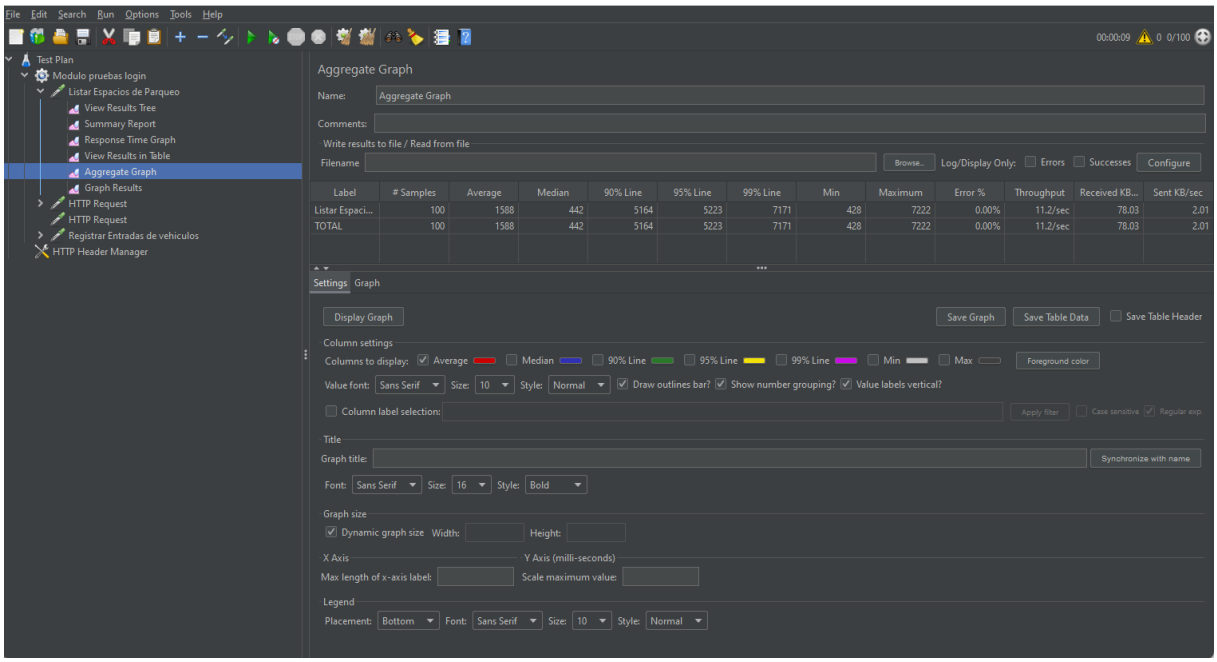
Resultado de la prueba de rendimiento para el web service de listar espacios (vista de resultados).



Nota. Resultado de las peticiones que realiza la aplicación de pruebas con los parámetros previamente indicados. En este listado se puede apreciar las peticiones que tuvieron fallas y aquellas que tuvieron éxito. Elaborado por Guachán, Sigcho.

Figura 41

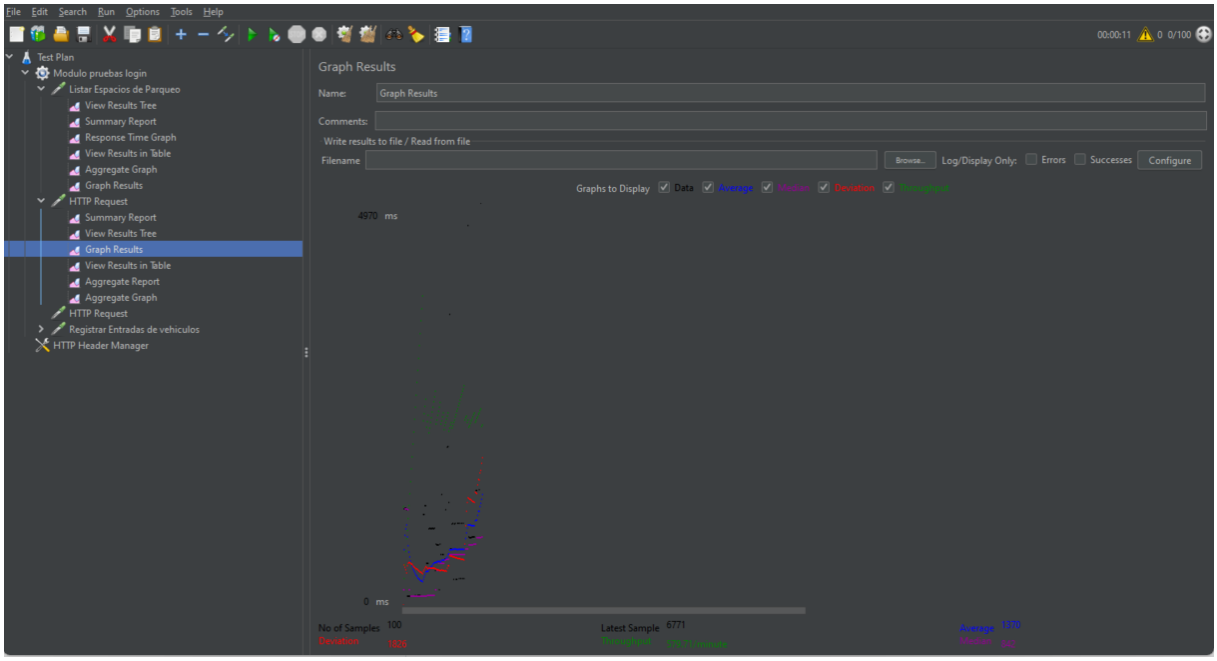
Resultado de la prueba de rendimiento para el web service de listar espacios (resumen).



Nota. Resultado de las peticiones que realiza la aplicación de pruebas con los parámetros previamente indicados. Este resumen muestra que para el caso de los 100 usuarios que se colocaron como parámetro de pruebas, el tiempo mínimo de espera fue de 0.428 segundos y su media 0.442 segundos. Elaborado por Guachán, Sigcho.

Figura 42

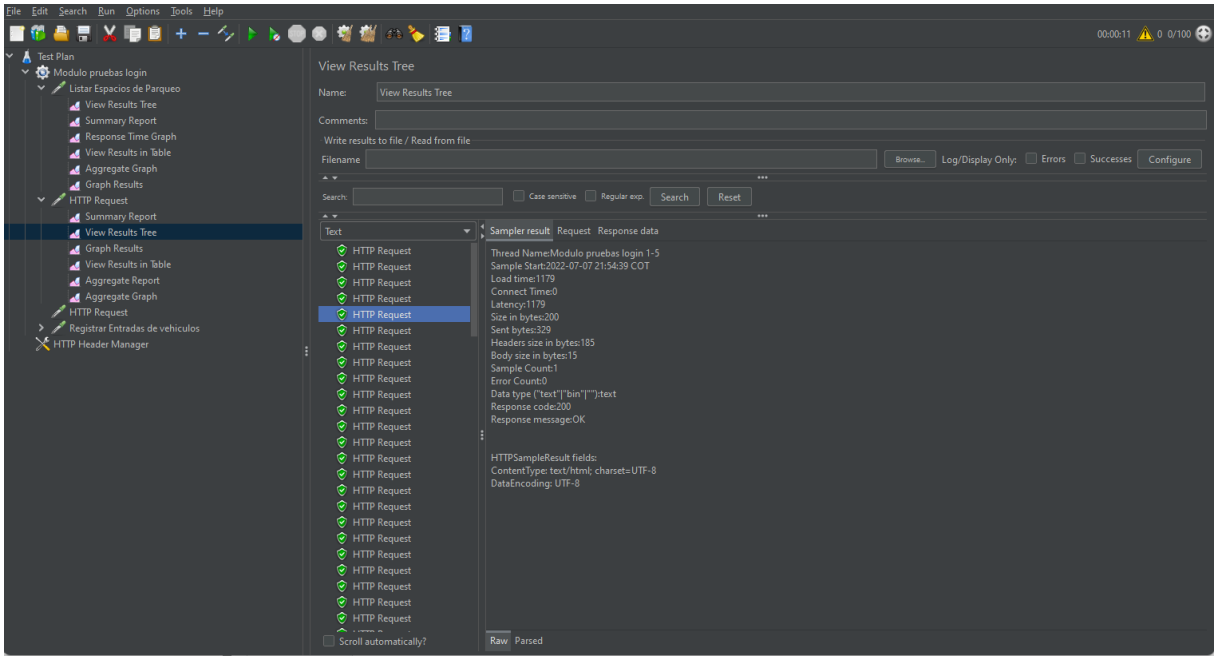
Resultado de la prueba de rendimiento para el ingreso al sistema (vista de gráfico).



Nota. Gráfica obtenida en las pruebas realizadas a la url de ingreso, con 100 ingresos en cinco segundos, dando como resultado una media de tiempo de respuesta de 842 milisegundos. Elaborado por Guachán, Sigcho.

Figura 43

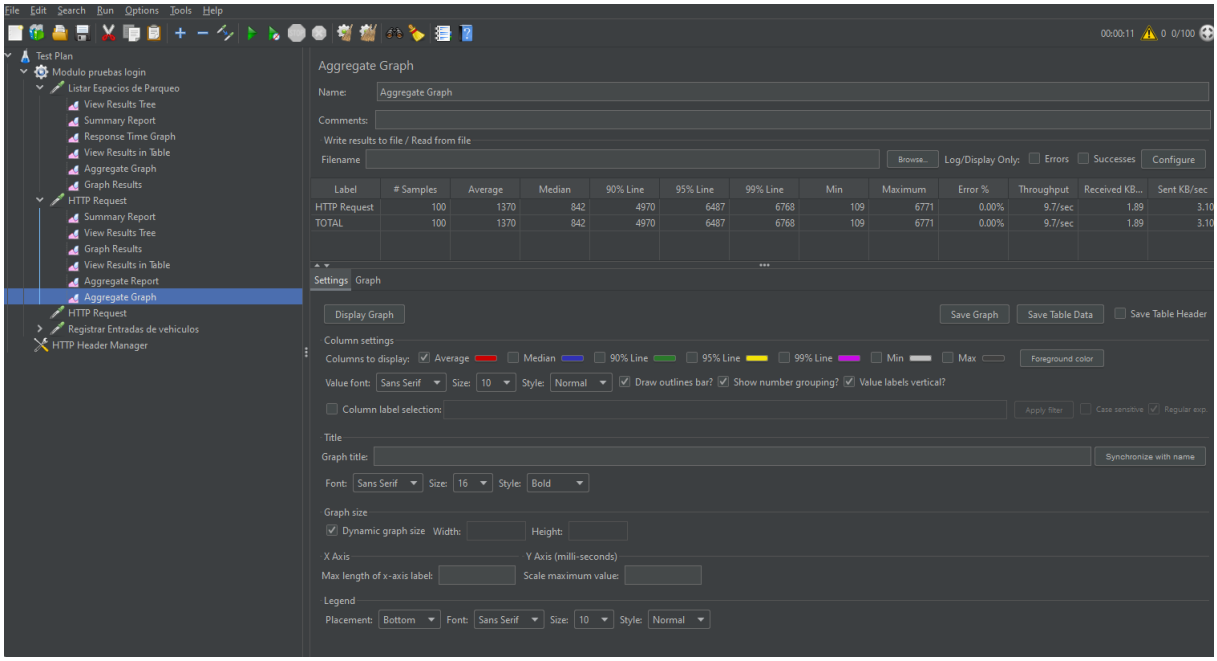
Resultado de la prueba de rendimiento para el ingreso al sistema (vista de resultados).



Nota. Resultado obtenido al realizar 100 peticiones de ingreso en un intervalo de 5 segundos, obteniendo una respuesta favorable en el 100% de los resultados. Elaborado por Guachán, Sigcho.

Figura 44

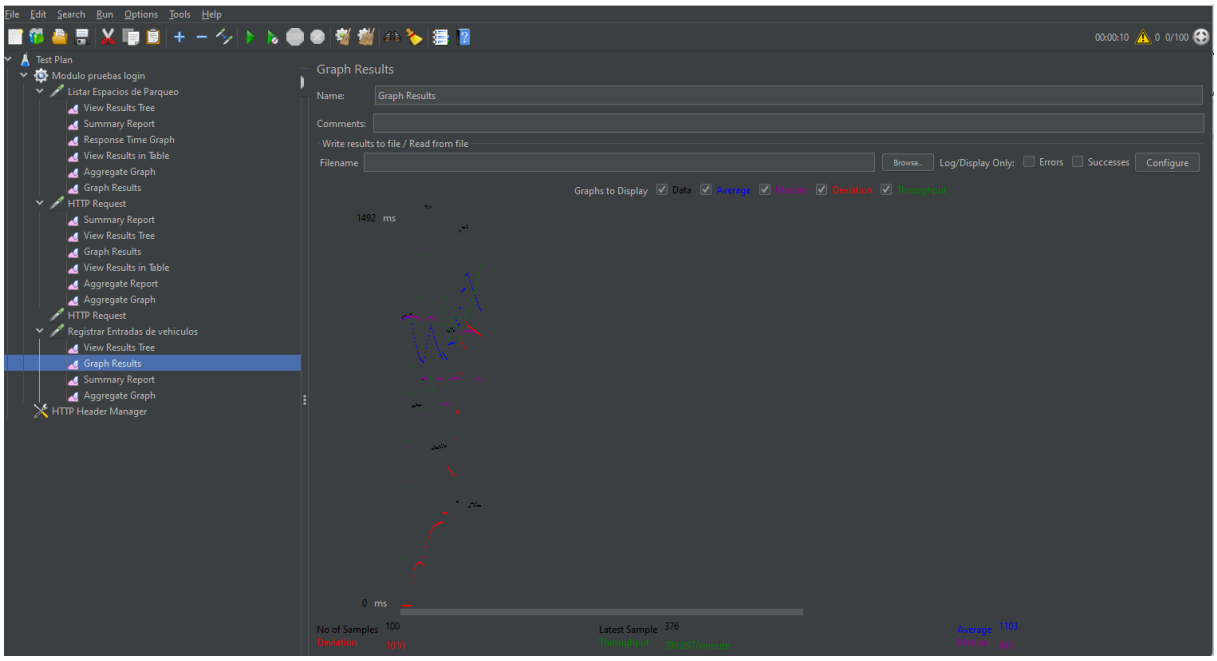
Resultado de la prueba de rendimiento para el ingreso al sistema (resumen de resultados).



Nota. Resumen de resultados obtenidos en las pruebas de ingreso al sistema, obteniendo un tiempo mínimo de 109 milisegundos y una media de 842 milisegundos y con un porcentaje de error de cero. Elaborado por Guachán, Sigcho.

Figura 45

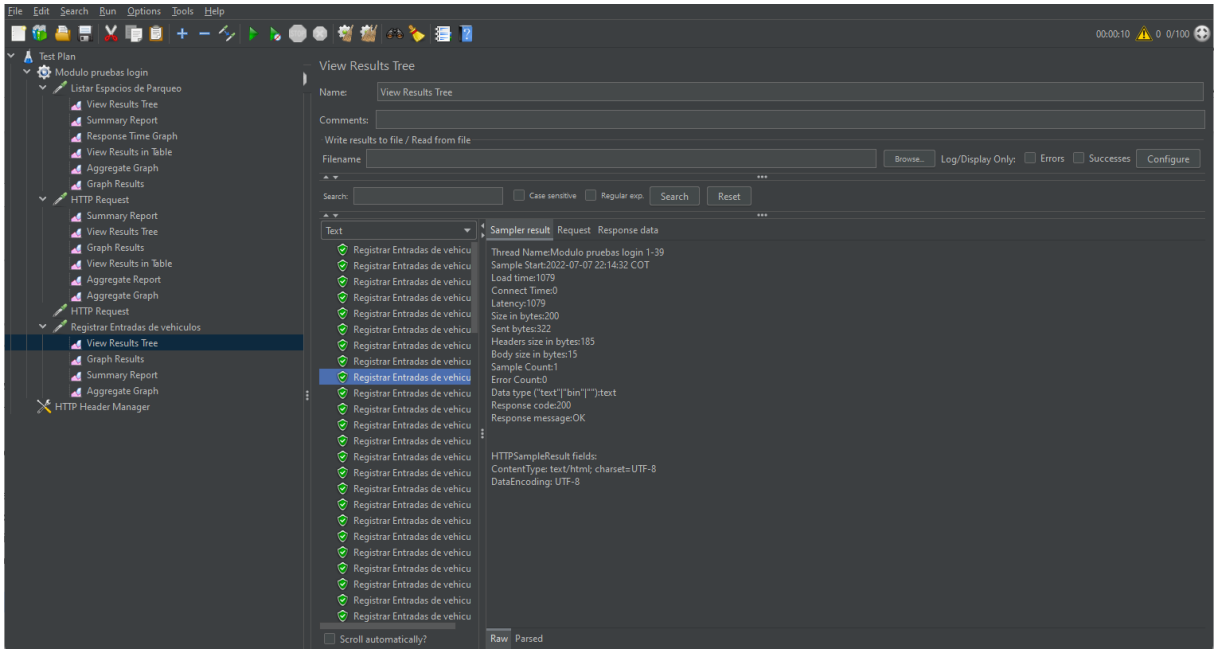
Resultado de la prueba de rendimiento para el ingreso de datos de vehículos (Gráfico).



Nota. Gráfica obtenida en las pruebas realizadas a la url de ingreso de datos de vehículos, con 100 ingresos en cinco segundos, dando como resultado un promedio de tiempo de respuesta de 1103 milisegundos. Elaborado por Guachán, Sigcho.

Figura 46

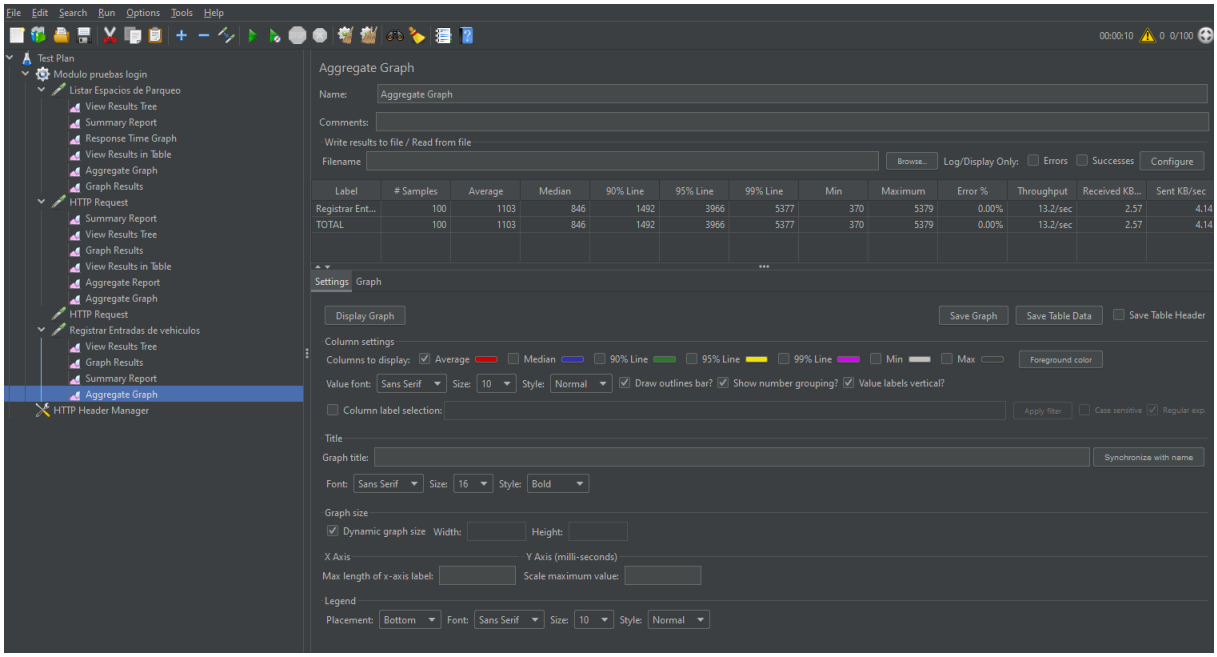
Resultado de la prueba de rendimiento para el ingreso de datos de vehículos (respuestas obtenidas).



Nota. Resultado obtenido al realizar 100 peticiones de ingreso de datos de vehículos en un tiempo de 5 segundos, obteniendo una respuesta favorable en el 100% de los resultados. Elaborado por Guachán, Sigcho.

Figura 47

Resultado de la prueba de rendimiento para el ingreso de datos de vehículos (resumen).



Nota. Resumen de resultados obtenidos en las pruebas de ingreso de datos de vehículos al sistema, obteniendo un tiempo mínimo de 370 milisegundos y una media de 846 milisegundos y con un porcentaje de error de cero. Elaborado por Guachán, Sigcho.

CONCLUSIONES

Una vez realizadas las pruebas de la aplicación en sistemas Android se pudo lograr una visualización clara y rápida de los espacios de parqueo disponibles en el sitio de implementación, ya que la interfaz muestra una primera pantalla con información sobre todas las zonas disponibles y sus lugares de parqueo libres evitando así elegir una zona de parqueo incorrecta.

Con el objetivo de mantener una base de datos con información de los vehículos que ingresan a un sitio de parqueo, el guardia hará uso de la aplicación podrá guardar datos importantes de los vehículos como lo son: el tipo del vehículo, el número de personas y la placa de dicho vehículo. Además, los datos de fecha y hora de ingreso se guardan de forma transparente para el usuario permitiendo así un registro de estos datos exacto al tiempo de ingreso.

El presentar la información al usuario en tiempo real permite agilitar y mejorar la experiencia del usuario, dándole una vista rápida e inmediata de los lugares disponibles para su uso, los cuales antes de ingresar al lugar con pocos pasos dentro de la aplicación podrán visualizar a que zona de parqueo deberán dirigirse para estacionarse, reduciendo así su tiempo de espera dentro de un estacionamiento.

El propósito de la aplicación dirigida al guardia consistía en mejorar la seguridad y control de los vehículos que ingresan al lugar y esto se consigue gracias a que el guardia al momento de realizar el ingreso dentro de la aplicación se guarda un histórico dentro de la base de datos con datos fundamentales.

RECOMENDACIONES

Para futuras implementaciones en caso de no ser un parqueadero público, mantener un repositorio controlado con accesos para la descarga de cada una de las aplicaciones para evitar así que personas ajenas a la institución donde se lo implemente tengan acceso a los mapas de parqueo del lugar.

Las aplicaciones tanto la de búsqueda de zonas de parqueo, como la aplicación dirigida al registro de ingreso de vehículos se podrían complementar con un panel de administración web o móvil que permita visualizar estadísticas de ingresos, información con la cual se podrían realizar informes y de ser el caso de un parqueadero publico informes financieros.

Para la aplicación de registro de ingresos de vehículos se podría aumentar un scanner mediante la cámara del celular permitiendo así un ingreso más rápido y agilizar los ingresos en horas pico, así como también conectar la aplicación a una base de datos del país desde la cual se puedan obtener más datos mediante la placa y alertar a administración en caso de tratarse de un vehículo con antecedentes policiales.

LISTA DE REFERENCIAS

Libros

- Lequerica, J. R. (2003). *Web Services (edición especial)*. ANAYA MULTIMEDIA.
- Robledo, D. (2016). *Desarrollo de aplicaciones para Android I*. Ministerio de Educación y Formación Profesional.

Páginas web

- Consejos para Ahorrar Gasolina - Maneje Más Eficientemente. (2001). fueleconomy. https://www.fueleconomy.gov/feg/esdriveHabits.shtml#:%7E:text=Mantener%20el%20auto%20encendido%20sin,segundos%20del%20uso%20de%20combustible
- Build apps for any screen. (s.f.). Flutter. https://flutter.dev/
- Herazo, L. (2020, 1 septiembre). ¿Qué es una aplicación móvil? Anincubator Website. https://anincubator.com/que-es-una-aplicacion-movil/
- *Los sitios web móviles que se cargan rápidamente atraen a más clientes: quere ayudarte a agilizar el tuyo*. (2017, 27 septiembre). Blog Oficial de Google España. Recuperado 7 de julio de 2022, de https://espana.googleblog.com/2017/09/los-sitios-web-moviles-que-se-cargan.html

Tesis

- Lozano, J. (2017, enero). *Implementación de una aplicación móvil, basado en XP, para mejorar el proceso de consulta de saldo de las tarjetas del metro de lima - línea 1*. [Tesis de grado, Universidad Autónoma del Perú]. Repositorio de la Universidad Autónoma del Perú. http://repositorio.autonoma.edu.pe/bitstream/handle/20.500.13067/391/LOZANO%20ANGULO%20JHAIR%20VINCENZO.pdf

Artículos

- Barriga, J. J., Sulca, J., León, J. L., Ulloa, A., Portero, D., Andrade, R., & Yoo, S. G. (2019). Smart parking: A literature review from the technological perspective. *Applied Sciences*, *9*(21), 4569.
- Baz, A., Ferreira, I., ÁLvarez, M., & García, R. (2011, febrero). *Dispositivos móviles*. http://isa.uniovi.es/docencia/SIGC/pdf/telefonia_movil.pdf
- Borman, R. I., Priandika, A. T., & Edison, A. R. (2020). Implementasi metode pengembangan sistem Extreme Programming (XP) pada aplikasi investasi peternakan. *JUSTIN (Jurnal Sistem Dan Teknologi Informasi)*, *8*(3), 272-277.
- Caicedo, F. (2010). Real-time parking information management to reduce search time, vehicle displacement and emissions. Transportation Research Part D: Transport and Environment, 15(4), 228–234. https://doi.org/10.1016/J.TRD.2010.02.008
- Carrera, M. (2022, 6 enero). Cada año se suman 17 539 vehículos nuevos en Quito. El Comercio. https://www.elcomercio.com/actualidad/quito/suman-vehiculos-nuevos-quito-2022.html#:%7E:text=En%20diciembre%20pasado%2C%20a%20prop%C3%B3sito,(AMT)%20matricul%C3%B3%20404%20327.
- Carrera, F. M., Govea, F. K., Hurtado, G. E., & Freire, C. E. (2019). Estudio correlacional de factores como desempleo e índices de delincuencia en Ecuador. *Información tecnológica*, *30*(3), 287-294.
- Censos, E. D. N. I. Y. (2020). *Transporte*. Instituto Nacional de Estadística y Censos. https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Economicas/Estadistica%20de%20Transporte/2020/2020_ANET_EVOLUCI%c3%93N%20HIST%c3%93RICA.pdf

- Chandrasekaran, S. S., Muthukumar, S., & Rajendran, S. (2013). Automated control system for air pollution detection in vehicles. Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS, 49–51. https://doi.org/10.1109/ISMS.2013.94
- Deepa, N., Prabadevi, B., Krithika, L. B., & Deepa, B. (2020, February). An analysis on version control systems. In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)* (pp. 1-9). IEEE.
- Faraji, S. J., & Naozar, M. J. (2019). Smart parking: an efficient approach to city's smart management and air pollution reduction. *Journal of Air Pollution and Health*, *4*(1), 53-72.
- González, J. F. (2013). Introducción a las metodologías ágiles. *Otras formas de analizar y desarrollar*.
- Kumar, P., Omidvarborna, H., Barwise, Y., & Tiwari, A. (2020). Mitigating exposure to traffic pollution in and around schools: Guidance for Children, Schools and Local Communities.
- M. P. Thakre, P. S. Borse, N. P. Matale and P. Sharma, "IOT Based Smart Vehicle Parking System Using RFID," 2021 International Conference on Computer Communication and Informatics (ICCCI), 2021, pp. 1-5, doi: 10.1109/ICCCI50826.2021.9402699.
- Maric, M., Gracanin, D., Zogovic, N., Ruskic, N., & Ivanovic, B. (2017). Parking search optimization in urban area. *International Journal of Simulation Modelling*, *16*(2), 2017.
- N. R. N. Zadeh and J. C. Dela Cruz, "Smart urban parking detection system," *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2016, pp. 370-373, doi: 10.1109/ICCSCE.2016.7893601.

- Rueda-López, J. J. (2007). La tecnología en la sociedad del siglo XXI: albores de una nueva revolución industrial. *Aposta. Revista de Ciencias Sociales*, (32), 1-28.
- Wu, W. (2018). React Native vs Flutter, Cross-platforms mobile application frameworks.