INDICE

[Información General. 4](#_Toc27186123)

[Propósito. 4](#_Toc27186124)

[Alcance. 5](#_Toc27186125)

[Personal involucrado. 5](#_Toc27186126)

[Convenciones, definiciones, acrónimos 5](#_Toc27186127)

[Referencias 6](#_Toc27186128)

[Resumen 6](#_Toc27186129)

[Perspectiva del producto 7](#_Toc27186130)

[Funcionalidad del producto 7](#_Toc27186131)

[Características del producto 8](#_Toc27186132)

[Restricciones 8](#_Toc27186133)

[Información funcional y no funcional. 9](#_Toc27186134)

[Proceso Funcional. 10](#_Toc27186135)

[Requerimientos Funcionales generales 11](#_Toc27186136)

[Requisitos comunes de las interfaces 14](#_Toc27186137)

[Interfaces de usuario 14](#_Toc27186138)

[Interfaces de hardware 14](#_Toc27186139)

[Interfaces de software 14](#_Toc27186140)

[Interfaces de comunicación 14](#_Toc27186141)

[Requerimientos Funcionales específicos 14](#_Toc27186142)

[Requisito funcional 1 14](#_Toc27186143)

[Requisito funcional 2 15](#_Toc27186144)

[Requisito funcional 3 15](#_Toc27186145)

[Requisito funcional 4 15](#_Toc27186146)

[Requerimientos No Funcionales 15](#_Toc27186147)

[Requisitos de rendimiento 15](#_Toc27186148)

[Seguridad 15](#_Toc27186149)

[Fiabilidad 15](#_Toc27186150)

[Disponibilidad 16](#_Toc27186151)

[Mantenibilidad 16](#_Toc27186152)

[Portabilidad 16](#_Toc27186153)

[Información técnica. 16](#_Toc27186154)

[Básico. 16](#_Toc27186155)

[Base Datos. 16](#_Toc27186156)

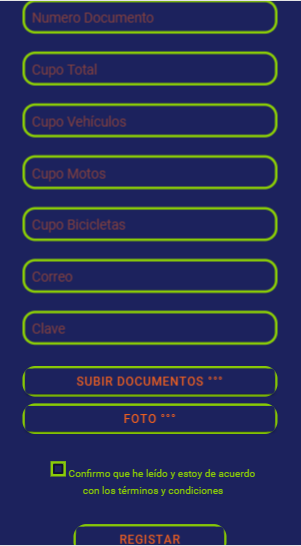
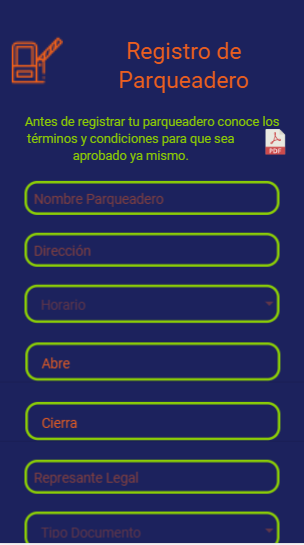
[Front End 17](#_Toc27186157)

[Vista “Login” 17](#_Toc27186158)

[Vista “selección categoría”. 18](#_Toc27186159)

[Vista “registro usuario”. 18](#_Toc27186160)

[Vista “registro parqueadero”. 19](#_Toc27186161)

[**** 20](#_Toc27186162)

[Vista “home”. 20](#_Toc27186163)

[Backend. 21](#_Toc27186164)

[Capa de negocio 21](#_Toc27186165)

[Capa de datos 21](#_Toc27186166)

[Capa de servicio 21](#_Toc27186167)

[Seguridad. 22](#_Toc27186168)

[Complementos. 22](#_Toc27186169)

[Pendientes de definición. 22](#_Toc27186170)

[Pendientes Técnicos. 22](#_Toc27186171)

[Links artículos de referencia. 22](#_Toc27186172)

# Información General.

## Propósito.

La razón principal para desarrollar soluciones tecnológicas al servicio de los ciudadanos es optimizar sus procesos ayudándolos a hacer más cosas en menos tiempo, optimizando así sus horas productivas o de ocio, dándole una mejor calidad de vida a la persona. Dentro de este contexto, una de las principales necesidades de los usuarios que tienen un vehículo (llámese moto, carro o bicicleta), es el poder encontrar un parqueadero cercano y de confianza donde puedan parquear con tranquilidad.

Y es que son muchos los problemas que aquejan a los ciudadanos en cuanto a parquear respecta, pues no solo son los largos minutos, que se convierten en horas perdidas buscando parqueadero, si no, también el aumento de la inseguridad en los parqueaderos, pues según varios artículos publicados en diferentes medios de información dicen que la inseguridad en los parqueaderos de Bogotá ha aumentado; las bandas criminales tienen nuevos métodos de robo, parqueaderos que no son legales, entre muchos otros problemas de seguridad.

De acuerdo con la información anterior, este proyecto pretende desarrollar una aplicación móvil que permita a los usuarios encontrar parqueaderos de confianza, reservarlos con anticipación y marcar rutas directas al parqueadero que haya reservado, evitando pérdidas de tiempo en el tráfico, pues este es uno de los problemas más grandes en Bogotá; un artículo publicado en febrero del 2019 por CNN en Español, demuestra que Bogotá es la tercera ciudad más congestionada del mundo, siendo Moscow la primera y Turquía la segunda, estos datos según el estudio INRIX.

Para responder al problema de la inseguridad y la confianza en los sitios de parqueo, los usuarios podrán calificar cada parqueadero después de cada servicio que tomen en el mismo, se podrán poner comentarios y calificaciones de 1 a 5 según criterios como: atención, espacio, accesibilidad, seguridad, confianza, entre otros.

Esta aplicación no solo ayudará al usuario que busca un parqueadero disponible, si no, también ayudará al dueño de parqueadero que quiere alcanzar más clientes, optimizar procesos de pago a través de la aplicación y gestionar sus espacios disponibles.

­

## Alcance.

Plantear, diseñar, desarrollar e implementar una aplicación en IONIC que muestre en un mapa parqueaderos cercanos y sus características como: espacios disponibles, opiniones, tarifa por minuto, opción de realizar reservas; con la finalidad de que los usuarios de vehículo (carro, moto, bicicleta) puedan optimizar sus tiempos de búsqueda en parqueaderos de confianza. Así como generar una plataforma de gestión de espacios disponibles y ocupados para los dueños del parqueadero, simplificar el proceso de pago a través de la aplicación. Esto con el objetivo de optimizar los procesos de gestión al dueño del parqueadero en cuanto espacios, pagos y generar más visualización y marketing en la plataforma y aumentar así la tasa de ocupamiento.

Como objetivo final de brindar una mejor calidad de vida a cada usuario brindándole más tiempo que estaba perdiendo debido a la búsqueda innecesaria de parqueaderos. Y ayudando a que el dueño de parqueadero genere más ingresos gracias a la simplificación y automatización de procesos.

## Personal involucrado.

|  |  |
| --- | --- |
| **Nombre** | Sara Estefania Camacho Albarracin |
| **Rol** | Analista, diseñador y programador |
| **Categoría Profesional** | Desarrolladora |
| **Responsabilidad** | Análisis de información, diseño y programación de **PARKIT** |
| **Información de contacto** | [Saritacamachoa77@gmail.com](mailto:Saritacamachoa77@gmail.com) |

## Convenciones, definiciones, acrónimos

|  |  |
| --- | --- |
| ***Nombre*** | ***Descripción*** |
| **Usuario** | Persona que usará el sistema para gestionar procesos |
| **Parqueadero** | Persona que brindará el servicio de parqueo |
| **Administrador** | Modificador de información y datos |

## Referencias

|  |  |
| --- | --- |
| **Título del Documento** | **Referencia** |
| Ranking de ciudades más congestionadas en el mundo | <https://cnnespanol.cnn.com/2019/02/14/estas-son-las-ciudades-con-peor-congestion-vehicular-y-movilidad-hay-4-latinoamericanas-en-el-top-10/> |
| Parqueaderos ilegales | <https://www.pulzo.com/nacion/estos-2-parqueaderos-norte-bogota-ilegales-estan-robando-gente-PP815010> |
| Estado del arte 1 | EstadoArte.doc |
| Diagrama de procesos | MapaProcesos.pdf |
| Robo de vehículos en parqueaderos | <https://www.pulzo.com/nacion/videos-robos-vehiculos-bogota-PP730923> |
| Aplicaciones existentes a fin | <https://noticias.canalrcn.com/tecnologia/cansado-de-no-encontrar-parqueadero-estas-apps-te-ubican-un-espacio-345853> |

## Resumen

Bogotá es un caos, una ciudad insegura y hasta el tope de trancones, siendo la tercera ciudad más congestionada del mundo, y la primera en la que sus ciudadanos pierden más horas en el tráfico siendo 272 las horas que pierden una persona atascos al año, teniendo en cuenta que a este problema de tráfico se suma el poder encontrar un parqueadero disponible y seguro para su vehículo.

Es necesario a través de los sistemas y las herramientas digitales el poder solucionar problemas como estos, tal vez no se puedan liberar los atascos, pero por lo menos un ciudadano de Bogotá no tendrá que perder horas en el tráfico buscando un parqueadero seguro, de confianza y sobre todo con disponibilidad en horas pico.

Considerando los anteriores problemas, el objetivo es diseñar, desarrollar e implementar un sistema que enlace a la comunidad con las diferentes ofertas de parqueaderos en toda la ciudad; de esta manera mejorando los tiempos que la persona pasa buscando parqueadero. Los parqueaderos ofertados en este sistema deben ser de total confianza y deberán poder ser valorados por la comunidad.

En la actualidad existen aplicaciones web y móviles que permiten al ciudadano encontrar un parqueadero, sin embargo, se pueden mejorar aspectos de cada una como los siguientes: algunos solo muestran parqueaderos para carros, otros solo muestran parqueos públicos, en otros solo se pude buscar y contactar con el parqueadero. Se debe diseñar, desarrollare implementar un sistema que integre todas las funcionalidades mencionadas anteriormente, permitiéndole al usuario conectar con parqueaderos públicos y privados, realizar pagos a través de la app, ofrecer parqueaderos para cualquier tipo de vehículo, implementar un sistema de gestión de espacios para los dueños de parqueaderos y desarrollar un sistema de seguridad para evitar robo de vehículos.

## Perspectiva del producto

**¿Qué herramientas practicas existen para encontrar parqueaderos, hacer pagos, gestionar procesos del parqueadero y facilitar la vida de los bogotanos?**

Actualmente en Bogotá existen ya varias aplicaciones web y móviles que cumplen con la función, sin embargo, tienen muchos aspectos a mejorar. Primero vamos a hacer un resumen de lo que la competencia ofrece a groso modo.

Hay aplicaciones que solo ofrecen parqueaderos públicos y no ofrecen un medio de pago, son más como un intermediario que establece la comunicación de las dos partes quien busca parqueadero y quien lo ofrece. Este tipo de plataformas son muy sencillas y no cumplen con todo lo que requiere un bogotano hoy en día.

Luego tenemos las aplicaciones que ofrecen parqueaderos públicos y pagos, pero no permiten ofrecer parqueos, es decir, son de una cadena de parqueaderos propios ya establecidos, en este caso en concreto hablaremos de Parking app. Esta aplicación es bastante completa, pues permite al usuario encontrar los parqueaderos de la cadena PARKING cercanos, establecer rutas, marcar favoritos, realizar reservas, realizar pagos y establecer mensualidades en caso de ser necesarias. Ahora el inconveniente de esta app es que no permite generar una oferta de parqueaderos, por lo que los usuarios muchas veces no podrán encontrar parqueaderos en todas las partes de la ciudad.

Por otra parte, tenemos las aplicaciones que permiten al usuario ofrecer parqueaderos públicos y privados, estás apps, están igual de completas con la misma funcionalidad que la de Parking y aparte de eso permite ofrecer parqueaderos, por lo que los usuarios podrán encontrar parqueos en casi cualquier parte de la ciudad, sin embargo, no ofrecen un sistema de gestión para los procesos de cada del parqueadero.

En base a toda esta información, es menester diseñar, desarrollar e implementar una aplicación que permita a los usuarios buscar parqueaderos tanto públicos como privados para cualquier tipo de vehículo, reservar, realizar pagos, establecer rutas rápidas para llegar al parqueadero, implementar sistemas de gestión de los parqueos, proveer visibilidad a los parqueaderos y de esta manera facilitar la vida de los usuarios.

## Funcionalidad del producto

Aplicación móvil para Smartphone con sistema operativo IONIC que permite realizar búsqueda de parqueaderos, reservación de espacios, establecimiento de rutas para llegar al destino, marcar favoritos, realizar pagos y establecer mensualidades, así como también informar de las características de cada parqueadero en puntuación de: servicio, calidad de espacio, atención, accesibilidad. Ofrecer un sistema de gestión de espacios para cada parqueadero y por ultimo dar visibilidad a los parqueaderos a través de la app.

## Características del producto

La aplicación PARKIT está diseñada para dispositivos móviles con sistema operativo Android a partir de su versión 7.0 (Nougat). El diseño de su interfaz fue desarrollado con la intención de ser fácil e intuitivo, con colores de alto contraste, como color principal el azul y su complementario, naranja, usado para acciones importantes en la aplicación. Esta corriente en el diseño de interfaces, permite la visualización de los elementos de manera óptima, facilitando la lectura, usabilidad y funcionalidad. La tipografía seleccionada, los colores de contraste a blanco y gris usados en combinación con sus aplicaciones regular y médium; permiten diferenciar jerárquicamente, la información principal de la secundaria, respecto a la distribución y visualización de los textos.

Los formularios están acompañados de texto e iconografía alusiva al tipo de dato que debe ingresar el usuario, permitiéndole guiarlo sobre la información que debe llenar en cada uno. Adicionalmente se generaron piezas gráficas distintivas para los marcadores de posición de acuerdo al tipo de indicador de seguridad que se visualiza en la pantalla.

El Icon App contienen el logo correspondiente a la aplicación, en colores verde, naranja, gris, y con fondos azul en opacidades para darle suavidad y simpleza.

La aplicación tiene tipo de navegación menú superior lateral izquierdo, que es accesible en la mayoría de los niveles de la aplicación y me permite navegar por todas las opciones del home principal, donde se accede directamente a las secciones de la app: home, mis servicios, pago, códigos promocionales, ayuda y soporte, configuración, notificaciones, términos y condiciones y cerrar sesión.

El home, requiere el uso del mapa/ geolocalización, por ende, el uso de la pantalla completa es fundamental, requiriendo limpieza visual al momento de su uso e implementación.

Todo ello desarrollado bajo el Framework desarrollo de aplicaciones móviles IONIC en su versión 4, a través del editor de código Visual Studio Code, construcción y conexión de la base de datos con Microsoft SQL Server y las diferentes librerías requeridas de acuerdo a las funcionalidades principales de la app.

## Restricciones

Para el funcionamiento correcto de la aplicación, esta debe ser instalada en un dispositivo móvil con sistema operativo Android desde su versión 7.0.1 (Nougat).

La orientación de su visualización es totalmente vertical, su funcionamiento requiere de conexión de datos a rede móvil o wifi, Utilización de Sistemas de Posicionamiento Global GPS. Registro de usuario y login dentro de la plataforma.

* Interfaz para ser usada con internet.
* Usa geolocalización GPS
* Lenguajes y tecnologías en uso: HTML, CSS, EXPRESS, NodeJS, Javascript, Typescript, Ionic Framework v4, SQL Transac.
* Los servidores deben ser capaces de atender consultas concurrentemente.
* El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

# Información funcional y no funcional.

## Proceso Funcional.

|  |
| --- |
| C:\Users\LENOVO\Desktop\amorshito\mapa proceso 1.PNG |
| C:\Users\LENOVO\Desktop\amorshito\mapa proceso 2.PNG |

## Requerimientos Funcionales generales

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF01 |
| **Nombre del Requerimiento:** | Autentificación de Usuario. |
| **Características:** | Los usuarios deberán identificarse para acceder al sistema. |
| **Descripción del requerimiento:** | El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad. |
| **Requerimiento NO funcional:** |  |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF02 |
| **Nombre del Requerimiento:** | Registrar Usuarios. |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al Usuario registrarse. Se deben suministrar datos como: nombre, apellidos, tipo de documento, número de documento, celular, correo electrónico, contraseña. |
| **Requerimiento NO funcional:** |  |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF03 |
| **Nombre del Requerimiento:** | Registro Parqueadero |
| **Características:** | Los usuarios deberán registrar el parqueadero para poder publicarlo. |
| **Descripción del requerimiento:** | El sistema permitirá al Usuario registrar el parqueadero. Se deben suministrar datos como: nombre parqueadero, dirección, horario de atención, abre, cierra, representante legal, tipo de documento, numero de documento, cupo total, cupo de vehículos, cupo de motos, cupo de bicicletas, correo, clave. |
| **Requerimiento NO funcional:** |  |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF04 |
| **Nombre del Requerimiento:** | Menú Home |
| **Características:** | Solicita confirmar la ubicación actual |
| **Descripción del requerimiento:** | El sistema solicitará confirmar su ubicación en tiempo real para mostrar parqueaderos cercanos |
| **Requerimiento NO funcional:** |  |
| **Prioridad del requerimiento:**  Alta | |

**Requerimientos No Funcionales.**

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RNF01 |
| **Nombre del Requerimiento:** | Interfaz de la aplicación. |
| **Características:** | La aplicación presentara una interfaz de usuario sencilla para que sea de fácil manejo a los usuario. |
| **Descripción del requerimiento:** | La aplicación debe tener una interfaz de uso intuitiva y sencilla. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RNF02 |
| **Nombre del Requerimiento:** | Desempeño |
| **Características:** | La aplicación garantizara a los usuarios un desempeño en cuanto a los datos almacenado en el sistema ofreciéndole una confiabilidad a esta misma. |
| **Descripción del requerimiento:** | Garantizar el desempeño de la aplicación a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  | |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  | |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  | |

## Requisitos comunes de las interfaces

### Interfaces de usuario

La interfaz con el usuario consistirá en un conjunto de vistas con botones, listas y campos de textos. Ésta deberá ser construida específicamente para el sistema propuesto y, será visualizada a través de Smartphone con sistema operativo Android a partir de su versión 7.0.1 (Nougat).

### Interfaces de hardware

Será necesario disponer de equipos de cómputos en perfecto estado con las siguientes características:

* Dispositivo con Android 7.0.1 (Nougat)

### Interfaces de software

* Dispositivo con Android 7.0.1 (Nougat)

### Interfaces de comunicación

Los servidores, clientes y aplicaciones se comunicarán entre sí, mediante protocolos estándares en internet, siempre que sea posible.

## Requerimientos Funcionales específicos

### Requisito funcional 1

**Autentificación de Usuarios:** los usuariosdeberán identificarse para acceder a la aplicación.

* El sistema podrá ser consultado por cualquier usuario dependiendo del rol en el cual se encuentre.

### Requisito funcional 2

**Registrar Usuarios:** El sistema permitirá al Usuario registrarse. Se deben suministrar datos como: nombre, apellidos, tipo de documento, número de documento, celular, correo electrónico, contraseña.

### Requisito funcional 3

**Registro Parqueadero:** El sistema permitirá al Usuario registrar el parqueadero. Se deben suministrar datos como: nombre parqueadero, dirección, horario de atención, abre, cierra, representante legal, tipo de documento, numero de documento, cupo total, cupo de vehículos, cupo de motos, cupo de bicicletas, correo, clave.

### Requisito funcional 4

**Menú Home:** El sistema solicitará confirmar su ubicación en tiempo real para mostrar parqueaderos cercanos

## Requerimientos No Funcionales

### Requisitos de rendimiento

* Garantizar que el diseño de las consultas u otro proceso no afecte el desempeño de la base de datos, ni considerablemente el tráfico de la red.

### Seguridad

* Garantizar la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
* Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales sean documentos, archivos y contraseñas.
* Facilidades y controles para permitir el acceso a la información al personal autorizado a través de Internet, con la intención de consultar y subir información pertinente para cada una de ellas.

### Fiabilidad

* El sistema debe tener una interfaz de uso intuitiva y sencilla

### Disponibilidad

* La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas, garantizando un esquema adecuado que permita la posible falla en cualquiera de sus componentes, contar con una contingencia, generación de alarmas.

### Mantenibilidad

* El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento con el menor esfuerzo posible
* La interfaz debe estar complementada con un buen sistema de ayuda (la administración puede recaer en personal con poca experiencia en el uso de aplicaciones informáticas).

### Portabilidad

* El sistema será implantado bajo la plataforma Ionic

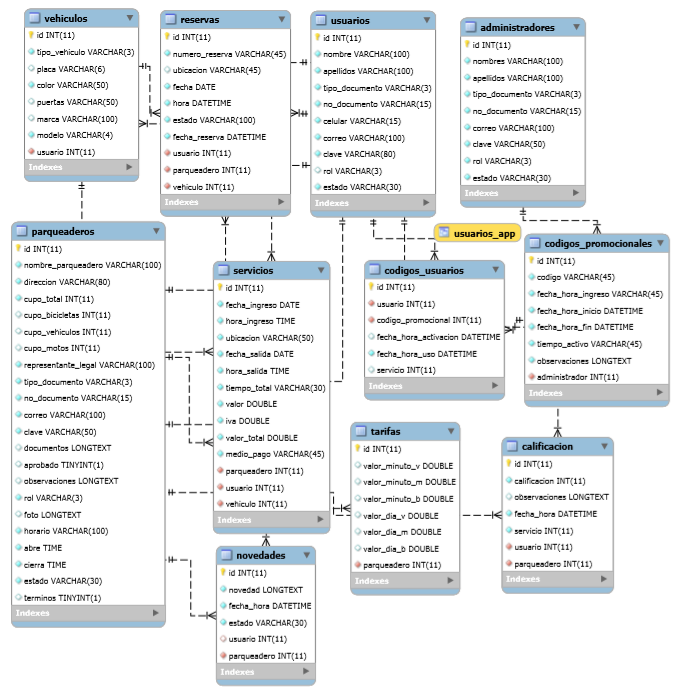
# Información técnica.

## Básico.

1. Las hojas tienen columnas escondidas. Las hojas con columnas escondidas tienen un objetivo de proceso y de formulación que ayuda al control del modelo. Es importante que cuando se requiera adicionar líneas o columnas se haga un análisis de impacto sobre el modelo y sobre las rutinas o macros que utilizan información o referencias de celdas y rangos fijos.

## Base Datos.

1. Esta base de datos es el repositorio donde se almacenan los valores comunes a los usuarios
2. El concepto general de tener una base de datos con la información centralizada es tener el control de adición y modificación de datos utilizados
3. Modelo MER:



1. Path: [**\\10.500.500.204\**](file:///\\10.500.500.204\AxedeBC\AxeBC.accdb)
2. Los perfiles administradores pueden modificar cualquier campo o tabla que reside en esta base datos.

Los perfiles usuarios pueden leer cualquier campo o tabla que reside en esta base datos.

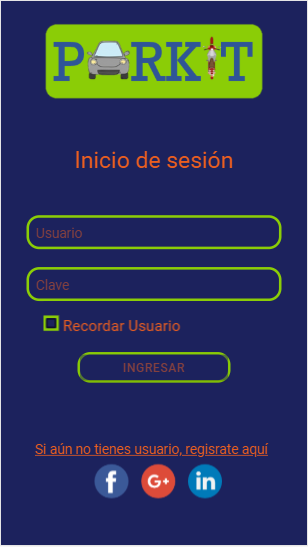
1. Script

## Front End

El front end se realizó en ANGULAR Y HTML

### Vista “Login”

Esta vista permite el ingreso al sistema de usuarios que están registrados en la base de datos, sino están registrados, encontrarán un link de acceso que les permitirá el registro a la aplicación. El formulario de login validará si la información ingresada por el usuario está correctamente diligenciada, al realizar esto el botón de iniciar sesión se activará y permitirá validar los datos en la base de datos.



### Vista “selección categoría”.

El usuario seleccionará la opción de registro que desea realizar si como usuario o como parqueadero, si selecciona usuarios es dirigido a la vista registro usuario. Si selecciona parqueadero es dirigido a la vista registro parqueadero.



### Vista “registro usuario”.

El usuario ingresará los datos correspondientes a: nombre, apellidos, tipo de documento, número de documento, celular, correo electrónico, contraseña.

El formulario validará que los datos ingresados como correo electrónico y contraseña coincidan entre sí, al realizar esto, el usuario debe aceptar los términos y condiciones del servicio. Y dar clic en registrarme.

Esta acción, enviará inmediatamente los datos a la base de datos y el usuario quedará registrado en la plataforma. Inmediatamente sus datos serán guardados y procederá al home principal.



### Vista “registro parqueadero”.

El usuario ingresará los datos correspondientes a: nombre parqueadero, dirección, horario de atención, abre, cierra, representante legal, tipo de documento, numero de documento, cupo total, cupo de vehículos, cupo de motos, cupo de bicicletas, correo, clave.

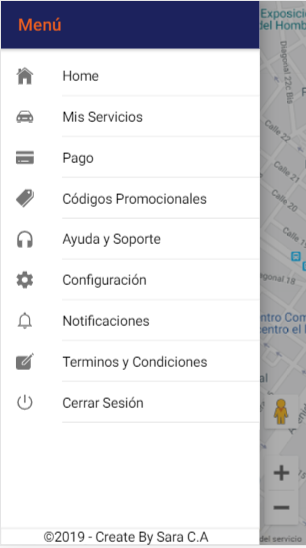
El formulario validará que los datos ingresados como correo electrónico y contraseña coincidan entre sí, al realizar esto, el usuario debe aceptar los términos y condiciones del servicio. Y dar clic en registrarme.

Esta acción, enviará inmediatamente los datos a la base de datos y el usuario quedará registrado en la plataforma. Inmediatamente sus datos serán guardados y procederá al home principal.

### **C:\Users\LENOVO\Downloads\registro parqueadero.pngC:\Users\LENOVO\Downloads\registro parqueaderro 2.png**

### Vista “home”.

El sistema solicitará confirmar su ubicación en tiempo real para mostrar parqueaderos cercanos. Una vez hecho, mostrara los parqueaderos cercanos a su ubicación.

 ****

## Backend.

El backend se realizó en NodeJS con una arquitectura de 3 capas, models, controlers y servicios.

### Capa de negocio

Se manejó la lógica de negocio de la app por medio de las clases A, B, y la clase C solo funciona como modelo de nuestro proyecto

### Capa de datos

Se creó un modelo que se conecta a nuestra base de datos que esta en el servidor 10.100.100.255:3306 por medio del usuario aaa y el password bbb.

### Capa de servicio

Se crearon las clases A, b, c, d y e que permite el uso de la api de nuestro servidor 10.100.100.255:3306 por medio del usuario aaa y el password bbb.

La clase A nos realiza los métodos get, post, put y delete de la tabla mama

La clase B nos realiza los métodos get, post y delete de la tabla papa

La clase B nos realiza los métodos get, de la tabla tio

## Seguridad.

1. La conexión con las api maneja un nivel de autenticación basado en auth 2.0 .
2. Autenticación básica con correo electrónico y contraseña proporcionado por el usuario

## Complementos.

Para compilar el programa se debe instalar el framework IONIC en su versión 4 y las siguientes dependencias para que pueda ser ejecutado correctamente a través del comando *npm i* en el editor de texto a través de la terminar. Sin embargo se debe asegurar de instalar node.js en primera instancia para ejecutar todas las instalaciones correspondientes (incluida IONIC)

El servidor de base de datos debe tener 1 Terabite de disco duro,16GB de RAM, Procesador AMD A8

# Pendientes de definición.

* Envío y recepción de Push notificación
* Alertas

# Pendientes Técnicos.

* Encriptación de información
* Conexión al servidor en la nube.
* Edición de perfil de usuario

# Links artículos de referencia.

\* <https://cnnespanol.cnn.com/2019/02/14/estas-son-las-ciudades-con-peor-congestion-vehicular-y-movilidad-hay-4-latinoamericanas-en-el-top-10/>

\*<https://www.pulzo.com/nacion/estos-2-parqueaderos-norte-bogota-ilegales-estan-robando-gente-PP815010>

\*<https://www.pulzo.com/nacion/videos-robos-vehiculos-bogota-PP730923>

\*<https://noticias.canalrcn.com/tecnologia/cansado-de-no-encontrar-parqueadero-estas-apps-te-ubican-un-espacio-345853>