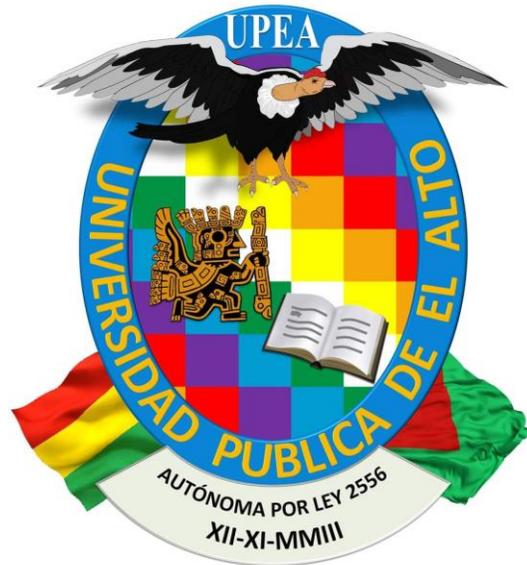


# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



## PROYECTO DE GRADO

### SISTEMA DE CONTROL Y SEGUIMIENTO DE PARQUEO APLICANDO LA TECNOLOGÍA RFID CASO: “PARQUEO PÚBLICO GONZALES”

Para Optar al Título de Licenciatura en Ingeniería de Sistemas  
MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Ruth Quispe Choque  
Tutor Metodológico: Ing. Maricel Yarari Mamani  
Tutor Especialista: Ing. Ramiro Kantuta Limachi  
Tutor Revisor: Lic. Katya Maricela Pérez Martínez

EL ALTO – BOLIVIA

2021

## **DEDICATORIA**

*A Dios por iluminarme y acompañarme en todos los momentos de mi vida,  
llenarme de bendición a mi vida,  
“A mis padres, Ángel Quispe Huanca y Celestina Choque de Quispe, que  
me dieron su apoyo incondicional desde el comienzo de mis estudios y  
darme la fuerza para seguir adelante con mis metas, siempre me apoyaron  
para no desistir, a quienes debo mi respeto y admiración a mis hermanas  
Zulema, Jhenny y mi hermano Giovany Alex por su apoyo en todo  
momento para seguir adelante.”*

## **AGRADECIMIENTOS**

*Al señor creador por darme le regalo de la vida y con ella darme la oportunidad de alcanzar mis sueños, e impulsándome para seguir adelante.*

*Agradezco a mis padres Ángel Quispe y Celestina Choque por su apoyo incondicional, su amor, su paciencia y la confianza en todos estos años de estudio.*

*Agradezco a mis amigos Carina Fabiola, Jorge Luis, Zulma Blanca por siempre incentivar y apoyarme incondicionalmente en todo momento.*

*Agradezco de manera especial a la Ing. Fanny Helen Perez Mamani por haberme aconsejado e incentivar a la culminación del presente proyecto.*

*Agradecer a mis tutores:*

*A la Ing. Maricel Yarari Mamani, mi Tutor Metodológica gracias por brindarme y guiarme en el desarrollo del presente proyecto, gracias por su conocimiento, paciencia y sin duda a su motivación para culminar este proyecto.*

*A la Ing. Ramiro Kantuta Limachi, mi Tutor Especialista que me brindo el apoyo incondicional y además de realizar sus valiosas observaciones.*

*A la Lic. Katya Maricela Pérez Martínez, mi Tutor Revisor que me brindo el apoyo incondicional a través de revisiones y además sus valiosas observaciones.*

*Fueron quienes me guiaron paso a paso durante el proceso con sus observaciones y sugerencias para el desarrollo y conclusión del presente proyecto de grado.*

## RESUMEN

En la actualidad las empresas públicas y privadas pretenden mejorar sus niveles de eficiencia en el manejo de las Tecnologías de Información, basándose en usar nuevas tendencias de medios tecnológicos, además de realizar en el control y seguimientos para una mejor administración con la ayuda de las tecnologías actuales.

El presente proyecto es un Sistema de Control y Seguimiento de Parqueo Aplicando la Tecnología RFID, con lo cual se pretende mejorar la administración del parqueo público “Gonzales”.

La metodología empleada en el proyecto es el OOHDM (*Object Oriented Hypermedia Design Methodology*), donde tiene cinco fases. Recolección de requerimientos u Obtención de requerimientos, diseño conceptual, diseño navegacional, diseño de interfaz abstracta e implementación. Para la evaluación de calidad de software se utilizó la ISO 25010 que nos permite conocer el nivel de calidad del software, en seguridad se recurrió la norma ISO 27001 y finalmente el costo se lo realizó con COCOMO II basado en el peso o líneas de código.

Para la implementación de software se utilizó el gestor de base de datos MySQL, además como lenguaje de programación PHP, con el framework Laravel, HTML, CSS, con el framework Bootstrap, JavaScript con VueJS.

## ÍNDICE GENERAL

### 1. MARCO PRELIMINAR

1.1.	Introducción.....	1
1.2.	Antecedentes Del Proyecto .....	2
1.3.	Planteamiento Del Problema.....	2
1.3.1.	Problema Principal.....	3
1.3.2.	Problemas Secundarios .....	4
1.4.	Objetivos.....	5
1.4.2.	Objetivo General.....	5
1.4.3.	Objetivos Específicos .....	5
1.5.	Justificación .....	5
1.5.2.	Justificación Técnica .....	5
1.5.3.	Justificación Económica.....	6
1.5.4.	Justificación Social.....	6
1.7.	Métricas De Calidad.....	7
1.7.2.	Estimación de Costo .....	8
1.7.3.	Seguridad de la información .....	8
1.8.	Herramientas .....	9
1.8.1.	Servidor Web Apache.....	9
1.8.2.	Gestor de Base de Datos MySQL.....	10
1.8.3.	Lenguaje de programación PHP.....	10
1.8.4.	Bootstrap.....	10
1.8.5.	Framework- Laravel .....	10
1.9.	Limites Y Alcances .....	11
1.9.1.	Limites.....	11
1.9.2.	Alcances .....	11

### 2. MARCO TEÒRICO

2.1.	Sistema .....	13
2.2.	Tecnología.....	13
2.3.	RFID ( <i>Radio Frequency Identification</i> ).....	14
2.3.1.	¿Qué es la identificación por radiofrecuencia? .....	14
2.3.2.	Arquitectura de un sistema RFID .....	15
2.3.3.	Etiquetas o tags .....	16

2.3.4.	Lectores de RFID .....	17
2.3.5.	Principio de funcionamiento .....	19
2.3.6.	Software de enlace.....	19
2.4.	Metodología de desarrollo .....	20
2.4.1.	Metodologías Web.....	20
2.4.2.	Metodología OOHDM .....	21
2.4.3.	Ventajas de OOHDM .....	26
2.5.	Herramientas .....	27
2.5.1.	HTML 5 .....	27
2.5.2.	Css .....	28
2.5.3.	JavaScript.....	29
2.5.4.	PHP 5 .....	29
2.5.5.	MySQL .....	31
2.5.6.	Laravel.....	32
2.5.7.	Boststrap .....	34
2.6.	Seguridad de la Información .....	35
2.7.1.	Norma de calidad Software ISO/IEC 27001 .....	35
2.8.	Métricas de calidad de software.....	36
2.8.1.	Calidad de software .....	36
2.8.2.	Estándar ISO/25000.....	36
2.9.	Modelo de Costos COMOMO II .....	42
2.9.1.	Fórmulas de estimación:.....	47
2.9.2.	Ventajas y Desventajas.....	49
2.10.	Pruebas De Funcionamiento .....	49
2.10.1.	Caja Blanca .....	49
2.10.2.	Caja Negra .....	50
<b>3.</b>	<b>MARCO APLICATIVO</b>	
3.1.	Desarrollo de la Metodología OOHDM .....	51
3.1.1.	Recopilación de Requerimientos .....	51
3.1.2.	Diseño Conceptual.....	58
3.1.3.	Diseño Navegacional.....	59
3.1.4.	Diseño de Interfaz Abstracta .....	61
3.1.5.	Implementación .....	63

3.2. Pruebas de software.....	84
3.2.1. Pruebas de caja blanca.....	84
3.2.2. Pruebas de caja negra .....	87
3.3. Métricas de Calidad .....	96
3.3.1. Factores de calidad ISO 25000 .....	96
3.4. Análisis de costos.....	108
3.4.1. Método de estimación COCOMO II .....	108
3.5. SEGURIDAD.....	110
3.5.1. Seguridad a nivel Base de Datos .....	110
3.5.2. Seguridad a nivel de Aplicación.....	110
3.6. Sistema de gestión de seguridad de la información ISO/27001 .....	111
3.6.1. Seguridad Lógica .....	111
3.6.2. Seguridad física.....	112
<b>4. CONCLUSIONES Y RECOMENDACIONES</b>	
4.1. Conclusiones.....	113
4.2. Recomendaciones.....	114
BIBLIOGRAFÍA .....	115
ANEXOS	

## ÍNDICE DE FIGURAS

Figura 1.	Podemos ver la estructura de la tecnología RFID .....	14
Figura 2.	Esquema de un sistema RFID.....	19
Figura 3.	Fases de la metodología OOADM.....	22
Figura 4.	Fase de implementación, integración con servidores .....	25
Figura 5.	La estructura el lenguaje de programación PHP .....	30
Figura 6.	Petición típica MVC en laravel.....	33
Figura 7.	División ISO/IEC 25000 .....	37
Figura 8.	Descripción de las características de la ISO/IEC 25010.....	37
Figura 9.	Diagrama general de caso de uso del parqueo .....	55
Figura 10.	Diagrama de caso de Uso: Gestión de Usuario.....	56
Figura 11.	Diagrama de caso de uso: Gestionar y Administrar Datos .....	57
Figura 12.	Diseño Conceptual .....	58
Figura 13.	Diagrama de navegación- Administrador.....	59
Figura 14.	Diagrama de navegación – Usuario Auxiliar.....	60
Figura 15.	Diagrama de navegación – Propietario (Cliente) .....	60
Figura 16.	Modelo de presentación Login.....	61
Figura 17.	Modelo de presentación página principal del Administrador.....	61
Figura 18.	Modelo de presentación página principal del Usuario Auxiliar.....	62
Figura 19.	Modelo de presentación en pantalla página principal del Propietario	62
Figura 20.	Inicio de Sesión .....	63
Figura 21.	Pantalla Principal del sistema .....	64
Figura 22.	Módulo de usuario .....	65
Figura 23.	Registro de Usuario .....	65
Figura 24.	Asignar Roles a los Usuario Auxiliares .....	66
Figura 25.	Módulo de tarifa .....	66
Figura 26.	Registro de tarifa .....	67
Figura 27.	Módulo de Propietario (Cliente) .....	67
Figura 28.	Registro de Propietario (Cliente).....	68
Figura 29.	Módulo de tipo de vehículo .....	68
Figura 30.	Registro de tipo de vehículo .....	69
Figura 31.	Módulo de Vehículo .....	69

Figura 32.	Registro Vehículo .....	70
Figura 33.	Módulo de Secciones .....	70
Figura 34.	Seguimiento de Mapeo de los Vehículos.....	71
Figura 35.	Nombre de los espacios .....	71
Figura 36.	Módulo de Cobros .....	72
Figura 37.	Visualiza la factura.....	72
Figura 38.	Módulo de Cobros .....	73
Figura 39.	Visualiza la entrada y salida .....	73
Figura 40.	Módulo de Cobros .....	74
Figura 41.	Visualiza la entrada y salida .....	74
Figura 42.	Visualiza el reporte de usuarios registrados .....	74
Figura 43.	Visualiza el reporte de lista de los vehículos .....	75
Figura 44.	Visualiza el reporte de lista de ingresos de los vehículos.....	76
Figura 45.	Visualiza el reporte de lista de salidas de los vehículos .....	77
Figura 46.	Visualiza el reporte de lista de tarifas .....	78
Figura 47.	Visualiza el reporte de lista de cobros .....	79
Figura 48.	Visualiza el reporte de lista de ubicaciones disponibles y ocupados .	80
Figura 49.	Visualiza las estadísticas de las ganancias .....	81
Figura 50.	Puede imprimir la estadística de las ganancias .....	81
Figura 51.	Visualiza las notificaciones de ingreso y salida .....	82
Figura 52.	Visualiza las el control con la tarjeta en pantalla .....	82
Figura 53.	Son los controles para salir del sistema .....	83
Figura 54.	Módulo de datos de la empresa.....	83
Figura 55.	Editar datos de la empresa.....	83
Figura 56.	Código del Módulo de propietarios pantalla principal .....	84
Figura 57.	Grafo de flujo del acceso al módulo de propietarios .....	85
Figura 58.	Prueba de Inicio de sesión .....	87
Figura 59.	Prueba de Acceso al Sistema.....	88
Figura 60.	Prueba Del Módulo De Tarifa .....	89
Figura 61.	Prueba Del Módulo De Propietario .....	90
Figura 62.	Prueba Del Módulo De Vehículo.....	91
Figura 63.	Prueba Del Módulo De Tipo de Vehículo.....	92

Figura 64. Prueba Del Módulo De Secciones .....	93
Figura 65. Prueba De Ingresos y Salidas .....	94
Figura 66. Prueba Del módulo de cobros .....	95

## ÍNDICE DE TABLAS

Tabla 1.	Descripción de ingresos de vehículos frecuentes .....	3
Tabla 2.	Descripción de los costos del parqueo .....	4
Tabla 3.	Productos y formalismos de la metodología OOHDM.....	26
Tabla 4.	Coeficiente del modelo COCOMO II .....	43
Tabla 5.	Variables factor de ajustes del esfuerzo .....	45
Tabla 6.	Tarea para obtención de requerimientos .....	51
Tabla 7.	Requerimiento de hardware.....	51
Tabla 8.	Requerimientos de software .....	52
Tabla 9.	Funciones del sistema .....	52
Tabla 10.	Requerimientos funcionales .....	53
Tabla 11.	Requerimientos no funcionales.....	53
Tabla 12.	Lista de Actores .....	54
Tabla 13.	Descripción de casos de uso: Gestión de usuario .....	56
Tabla 14.	Descripción de casos de uso: Gestión y Administrar Datos .....	57
Tabla 15.	Prueba del módulo de propietarios .....	87
Tabla 16.	Prueba de inicio de sesión.....	88
Tabla 17.	Prueba de Acceso al Sistema.....	88
Tabla 18.	Prueba Del Módulo De Tarifa .....	89
Tabla 19.	Prueba Del Módulo De Propietario .....	90
Tabla 20.	Prueba Del Módulo De Vehículo.....	91
Tabla 21.	Prueba Del Módulo De Tipo de Vehículo.....	92
Tabla 22.	Prueba Del Módulo De Secciones .....	93
Tabla 23.	Prueba Del Módulo De Ingresos y Salidas .....	94
Tabla 24.	Prueba Del Módulo De Ingresos y Reportes .....	95
Tabla 25.	Parámetros de Medida y su Cantidad.....	98
Tabla 26.	Parámetros del punto de función .....	98
Tabla 27.	Valores de ajustes de complejidad .....	99
Tabla 28.	Valores de fiabilidad de cada modulo .....	101
Tabla 29.	Parámetros de medición.....	103
Tabla 30.	Parámetros de medición sobre la usabilidad del sistema .....	104
Tabla 31.	Información requerida por el IMS.....	105

Tabla 32.	Fator de portabilidad.....	107
Tabla 33.	Calidad Global del Sistema .....	107
Tabla 34.	Cálculo de atributos <i><b>MXi</b></i> .....	108
Tabla 35.	Tabla general de costos .....	110
Tabla 36.	Copias de seguridad.....	111

# **CAPÍTULO I**

## **MARCO PRELIMINAR**



## 1.1. Introducción

En los últimos años la tecnología de Identificación por Radiofrecuencia RFID<sup>1</sup> logro un gran avance, aplicándose en las diferentes áreas administrativas de las empresas e industrias, La tecnología RFID está compuesta por un lector RFID y una etiqueta RFID, que es generalmente pequeña y se adhiere o incrusta dentro del objeto, animal o personas para su identificación.

La identificación por radiofrecuencia es una tecnología básicamente de captura e identificación automática de información contenida en etiquetas. Cuando estos transpondedores están en el área de cobertura de un lector RFID, este envía una señal para que la etiqueta o tags le transmite la información almacenada en su memoria. Desde este punto de vista, resulta claro que RFID ofrece interesantes potencialidades, siquiera como sustituto más versátil de las aplicaciones de identificación tradicionales basadas en el código de barras.

Además de hacer el seguimiento de los objetos también podemos hacer el uso de los tags para el control de inventario de los objetos de un lugar, este el área donde más se está utilizando la tecnología RFID.

Mediante el presente proyecto se dará una solución favorable para el parqueo público Gonzales, desarrollando un sistema de información que permita automatizar el control y seguimiento vehicular, de manera que al momento de realizar el cotejamiento e igualación de los datos registrados que pueda agilizar la toma de decisiones, para el personal de la empresa, donde cuente con información clara, precisa, actualizada e instantánea para, de esta forma, satisfacer la demanda. Además, el sistema agilizará la obtención de reportes diarios de todos los vehículos y usuarios, que accedan al servicio dentro del parqueo público Gonzales, aplicando la metodología OOHDM, lo que nos permite una planificación adecuada, además de herramientas tecnológicas para el desarrollo del software, este sistema será muy útil y de gran ayuda para la empresa.

---

<sup>1</sup> Identificador por Radiofrecuencia

## 1.2. Antecedentes Del Proyecto

El parqueo público GONZALES es una empresa dedicada a la atención de usuarios donde ofrece servicios de parqueo para sus vehículos, el cual cuenta con un personal capacitado para realizar sus actividades de servicio hacia los clientes.

- Jorge M. Rios Vidalon (2011) Universidad Católica del Perú, “Diseño de un Sistema de Control Vehicular Basado en el Acceso de Espacio Libres y Ubicación en estacionamientos Usando RFID”. El proyecto tiene como finalidad desarrollar un sistema que cumpla un eficiente control, que brinden facilidades y den buen servicio a los usuarios obteniendo así, una buena administración del estacionamiento con sistemas automáticos.
- Vladimir Quelca (2016), Universidad Mayor de San Andrés, “Sistema web de control, ventas e inventarios y verificación de temperatura de medicamentos usando RFID y alarmas tempranas caso: “farmacias la casa de salud””. La investigación plantea la implementación de la tecnología RFID en los vehículos públicos mediante el etiquetado, el cual se realizará el control de la información más eficiente. Aplicó la metodología XP (programación extrema), lenguaje de programación PHP y base de datos MySQL.
- Paul W. Landaeta. (2017) – Universidad Mayor de San Andrés, “sistema de control de inventarios móvil utilizando la tecnología RFID caso: CODEROAD”, el presente proyecto es diseñar una aplicación móvil capaz de hacer el control de inventario del área de tecnología de Información (IT) de la empresa Coderoad, aplicando la tecnología RFID. Aplicó la metodología ágil Mobile-D.

## 1.3. Planteamiento Del Problema

En la actualidad el control y seguimiento de ingresos y salidas vehicular dentro del parqueo público GONZALES, se lleva a cabo usando el control manual y la información que el cliente adquiere, no es al instante.

La lectura de cada vehículo se lo realiza de forma individual, provocando una pérdida de tiempo a la hora de realizar los registros de cada vehículo y el cliente queda desconforme. Otro problema es que los lugares asignados para el vehículo se encuentran ocupado, lo que causa una molestia del cliente, además no maneja siempre el registro, solamente realiza un acuerdo verbal con el propietario del vehiculó.

### 1.3.1. Problema Principal

El control vehicular de ingresos y salidas se realiza manualmente en el parqueo público GONZALES, tiene muchos problemas y limitaciones a la hora de realizar un control y seguimiento de registros, lo que provoca mucha pérdida de información y tiempo, por lo cual surge la siguiente pregunta:

¿De qué manera se puede automatizar el control vehicular de ingresos y salidas del parqueo público Gonzales?

En la siguiente *Tabla 1*. Se describe el ingreso de vehículos que son frecuentes al parqueo.

**Tabla 1. Descripción de ingresos de vehículos frecuentes**

---

**Horarios:** *lunes a viernes 07:00 a 22:00 y sábados 07:00 a 16:00*

<b>Tipo</b>	<b>Cantidad</b>
1 hora	18
021 día	20

---

**Fuente:** (Parqueo Público Gonzales)

En la siguiente *Tabla 2*. Se describe los costos establecidos en el parqueo para la elección de los clientes.

**Tabla 2. Descripción de los costos del parqueo**

**Horarios:** *lunes a viernes 07:00 a 22:00 y sábados 07:00 a 16:00*

Tiempo	Precio Bs.
1 hora	5
2 horas	8
3 horas	12
4 horas	15
5 horas	18
6 horas	20
7 horas	22
8 horas	25
9 horas	28
10 horas	30
11 horas	32
12 horas	35

**Fuente:** (Parqueo Público Gonzales)

### 1.3.2. Problemas Secundarios

- ✓ El registro de los vehículos, se lo realiza de forma manual, lo que genera pérdida de tiempo.
- ✓ Los reportes de ingresos y salidas de vehículos en el parqueo no existen.
- ✓ No se cuenta información al instante de los espacios disponibles.
- ✓ El control y seguimiento de ingresos, salidas del parqueo se lo realiza de forma verbal, por lo que no se controla bien.
- ✓ No obtienen información al instante y el usuario queda desconforme con la atención.

- ✓ La información de los diferentes vehículos no se puede obtener en forma oportuna, ya que no se tiene ningún registro físico de los mismos.

## **1.4. Objetivos**

### **1.4.2. Objetivo General**

Desarrollar un sistema de control y seguimiento de parqueo aplicando la tecnología RFID, que mejore, acelerare el manejo de la información de los usuarios, así brindar un mejor servicio a los usuarios que ofrece el parqueo público Gonzales.

### **1.4.3. Objetivos Específicos**

- ✓ Realizar el control de registros de ingresos y salidas de los vehículos con la tecnología RFID, para tener una información en tiempo real
- ✓ Obtener reportes de los registros para tener un mejor manejo y evitar perdida de información.
- ✓ Hacer la búsqueda en tiempo real de lugares disponibles para mejorar la calidad del servicio con el mapeo.
- ✓ Centralizar la información en una Base de Datos de los registros realizados dentro del parqueo.
- ✓ Obtener información de los usuarios y vehículos de forma rápida y eficiente con el sistema.

## **1.5. Justificación**

### **1.5.2. Justificación Técnica**

Para una buena interacción entre el usuario y el software se necesita un alto manejo de usabilidad, una interfaz limpia y un sistema rápido, por tal motivo la implementación del proyecto se realizará bajo las mismas condiciones, el desarrollo

de software del sistema de control y seguimiento de vehículos de parqueo aplicando la tecnología RFID.

### **1.5.3. Justificación Económica**

El desarrollo de sistema para el parqueo público GONZALES, se desarrolla bajo herramientas de software libre y la tecnología RFID, el cual no generará costo alguno para llevar a cabo su realización. Los procesos de información serán automatizados y permitirá reducir el tiempo de acceso a la información, reducir el exceso manejo de documentación y reducir gastos que representan el material del escritorio.

### **1.5.4. Justificación Social**

Se conoce que el control de registros, manualmente puede provocar muchas pérdidas de la información y tiempo, con lo que no se puede llegar a conocer correctamente, por ello surge la idea del desarrollo de un sistema, que facilite la lectura de etiquetas o tags para tal objetivo se usa la tecnología RFID.

El sistema beneficiará al control y seguimiento vehicular del parqueo público GONZALES, al momento de leer la etiqueta RFID este recibirá información exacta del vehículo, además obtendrá los datos de las personas que fue asignado, ayudando así tener un registro más exacto de todos los vehículos inventariados de la empresa.

## **1.6. Metodología OOHDM**

La metodología OOHDM u *Object Oriented Hypermedia Design Method*, para el diseño de las aplicaciones y para la web, fue de diseñado por D. Schwabe y G. Rossi , establece que el desarrollo de un Hiperdocumento es un proceso de cinco fases en el que se combinan diferentes estilos de desarrollo como el incremental, iterativo y prototipo. Las tres primeras fases son de diseño, en las que se obtiene un conjunto de modelos orientados a objeto que describen el documento que será construido en la última fase.

Las siguientes fases son:

- ✓ Fase 1: Obtención de requerimientos
- ✓ Fase 2: Diseño Conceptual
- ✓ Fase 3: Diseño Navegaciones
- ✓ Fase 4: Diseño De Interfaz Abstracta
- ✓ Fase 5: Implementación

Tiene actividades separadas que permiten obtener diseños modulares y reusables.

También se puede obtener un framework que contenga diseños específicos que se pueden usar en cada una de las etapas. Los desarrolladores podrán ir generando su propia biblioteca de recursos que podrán utilizar en futuros desarrollos, generando un léxico común que mejora la comunicación dentro del equipo. Finalmente se buscan esquemas que permitan mapear las primitivas de diseño (nodos, links, contextos) en ambientes de implementación. En el caso de implementación en ambientes o lenguajes no orientados a objetos, no es directa, pero al ser un método orientado a objetos, el mapeo con tecnologías orientadas a objetos se ve facilitada.

OOHDM es un método para modelar aplicaciones web de gran envergadura, que puede ser usado para modelar distintos tipos de aplicaciones navegables, sitios web, sistemas de información, presentaciones multimedia. (Lacarma Maria, 2013)

## **1.7. Métricas De Calidad**

### **1.7.1. Modelo de Calidad del Software ISO/IEC 25000**

Para la calidad el proyecto se utilizara la norma ISO/IEC 25000 constituye una serie basado en la ISO/IEC 9126 Y ISO/IEC14598 cuyo objetivo principal es guiar el desarrollo de los productos de software mediante la especificación de los requisitos y evaluación de características de calidad, la ISO/IEC 25000 es creación de un marco de trabajo común para evaluar la calidad del producto software, sustituyendo a las

anteriores ISO/IEC 9126 e ISO/IEC 14598 y convirtiéndose, de esta manera, en la piedra angular de esta área de ingeniería de software.

Para cumplir los requisitos de esta norma ISO 25000, se realiza un proceso de evaluación, mediante las siguientes actividades.

- ✓ Establecer requisitos de evaluación
- ✓ Especificar evaluación
- ✓ Diseño de la evaluación
- ✓ Ejecutar evaluación

(Fuente: <http://iso25000.com/>)

## **1.7.2. Estimación de Costo**

Para el análisis de costos del sistema, haremos uso del modelo de estimación empírica de:

### **1.7.2.1. Modelo COCOMO II**

El Modelo Constructivo de Costes COCOMO II u Constructive Cost Model, es un modelo matemático de base empírica utilizado para la estimación de costos de software, permite realizar estimaciones en función del tamaño del software y de un conjunto de factores de costo y escala. COCOMO II posee tres modelos denominados de aplicación, diseño temprana y post- arquitectura (COCOMO II, 2016).

## **1.7.3. Seguridad de la información**

### **1.7.3.1. Estándar ISO/IEC 27000**

Se utilizará para la seguridad la ISO/IEC 27000, sistemas de gestión de la seguridad de la información.

Las normas ISO se crearon con la finalidad de ofrecer orientación, coordinación, simplificación y unificación de criterios a las empresas y organizaciones con el objeto de reducir costes y aumentar la efectividad, así como estandarizar las normas de productos y servicios para las organizaciones internacionales.

Nos permitirá realizar lo siguiente:

- ✓ Analizar y ordenar la estructura de los sistemas de información.
- ✓ Establecer los procedimientos de trabajo para mantener su seguridad.
- ✓ Disponer de controles para medir la eficacia de lo establecido en el punto anterior.

La idea es alcanzar un nivel de riesgo menor que el soportado por la institución, para preservar la confidencialidad, integridad y disponibilidad de la información, ([www.iso27000.es](http://www.iso27000.es)).

## **1.8. Herramientas**

Las herramientas que se utilizaran para el desarrollo de software son:

### **1.8.1. Servidor Web Apache**

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Apache es altamente configurable, admite bases de datos de autenticación y negociado de contenido, aunque carece de una interfaz gráfica que ayude en su configuración.

Apache es una aplicación que permite montar un servidor web en cualquier equipo y casi cualquier sistema operativo. Al contrario que IIS (Internet Information Server) que sólo funciona en sistemas operativos de Microsoft.

Apache soporta PHP como lenguaje de programación. Con los módulos adecuados, Apache puede soportar también ASP. (Pérez Díaz, 2007)

### **1.8.2. Gestor de Base de Datos MySQL**

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que, de otra manera, se vulneraría la licencia GPL. (Pérez García, 2007).

### **1.8.3. Lenguaje de programación PHP**

Es un lenguaje Script interpretado del lado del servidor para la generación de las páginas web Dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. Este lenguaje no necesita ser compilado para ejecutarse. Necesita tener instalado un servidor web como apache, Nginx o IIS con librerías de PHP para poder funcionar señala (Pérez Valdés 2007).

### **1.8.4. Bootstrap**

Es un conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales. (Mark Otto, 2012)

### **1.8.5. Framework- Laravel**

Laravel es un framework web de código abierto creado por Taylor Otwell en 2011. Además, es utilizado para desarrollar aplicaciones web y sistemas web. Laravel está programado en PHP, lo que nos permite realizarlo de una forma elegante y simple. Además, nos permite el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP.

Su principal característica es que se basa en el paradigma de diseño MVC, el cual organiza el código de la aplicación en tres grandes bloques: Modelo, Vista y Controlador. (Martínez Perelló, 2017)

- ✓ **Modelo:** Permite abstraer el código de nuestra aplicación de la forma en la que se almacenan los datos. Además, se encarga también de implementar los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación.
- ✓ **Vista:** Se encarga de mostrar los datos que ha solicitado el usuario, permitiéndole interactuar con ellos a través de diferentes eventos.
- ✓ **Controlador:** Se encarga de gestionar los eventos que produce el usuario a través de la vista, así como de interactuar con la información contenida en el modelo.

## 1.9. Límites Y Alcances

### 1.9.1. Límites

El sistema se limitará a las siguientes condiciones:

- ✓ El sistema no emitirá facturación vinculada a impuestos nacionales, porque no cuenta actualmente con el NIT.
- ✓ El sistema será utilizado únicamente por el personal de la empresa.
- ✓ El usuario podrá ver información de los vehículos mediante reportes y también el detalle de cada uno de ellos.

### 1.9.2. Alcances

En cuanto los alcances del proyecto son los siguientes:

- ✓ El sistema contará con reportes para poder acceder a la información del inventario en tiempo real de los módulos correspondientes.

- ✓ Tendrá un módulo para un mejor control de ingresos y salidas, donde pueda tener reportes diarios.
- ✓ Tendrá una base de datos actualizada donde realizará el almacenamiento de la información.
- ✓ Tendrá un módulo de vehículos donde contará con la información de todos los vehículos de manera eficaz.
- ✓ El sistema será capaz de realizar registros con el lector RFID de manera directa.
- ✓ Los módulos correspondientes tendrán para realizar el registro, actualizar y eliminar usuarios correspondientes.
- ✓ Almacenar los datos modificados, en cualquiera de los atributos excepto los códigos.
- ✓ Tendrá un manual para el uso respectivo de los módulos.

# **CAPÍTULO II**

## **MARCO TEÓRICO**



## **2.1. Sistema**

Un sistema está conformado por un conjunto de entes u objetos componentes que interactúan entre sí para el logro de objetivos. De allí que la teoría general de sistema no solo estudia la estructura del sistema sino su comportamiento, su funcionamiento, dependiendo esta última de su estructura.

Un sistema tiene la propiedad de que toda acción que produce cambios en una de las partes de los sistemas, también estos cambios se dan en el resto del sistema. El Sistema también relacionara ante cualquier evento o estímulo producido en cualquier parte de la unidad. (Ricardo, 2012, pág. 33)

El concepto de sistemas ha sido utilizado por dos líneas de pensamiento diferentes.

La primera es la teoría de sistemas generales, corriente iniciada por von Bertalanffy y continua por Boulding y otros. El esfuerzo central de este movimiento es llegar a la integración de las ciencias. El segundo movimiento es bastante más práctico y se conoce con el nombre de “ingeniería de sistemas” o “ciencias de sistemas” iniciada por la Investigación de Operaciones y seguida por la administración científica Management Sciences y finalmente por el Análisis de Sistemas. (Oscar Bertoglio, 2015, págs. 53-54)

## **2.2. Tecnología**

Tecnología significa aplicación sistemática del conocimiento científico (u otro conocimiento organizado) a tareas prácticas. Su consecuencia más importante es una función de la división y subdivisión de cada una de esas tareas en partes o fases componentes”. (J.K. Galbrait, 1984, pág. 4)

“La Tecnología es una rama del saber, constituida por el conjunto de conocimientos y de competencias necesarias en la utilización, mejora y creación de las técnicas. Y una Técnica, está compuesta por el conjunto de operaciones que deben ser efectivamente realizadas para la fabricación de un bien dado”. (Bertrand Nezeys, 1985, pág. 3)

## 2.3. RFID (*Radio Frequency Identification*)

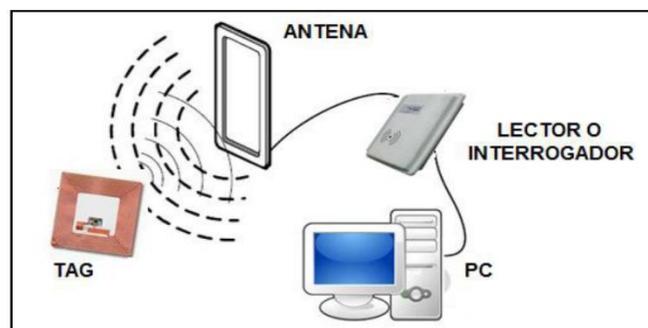
### 2.3.1. ¿Qué es la identificación por radiofrecuencia?

El sistema de identificación por radiofrecuencia o RFID (*Radio Frequency Identification*) es un sistema inalámbrico de almacenamiento y recuperación de datos que usa ondas de radio para determinar la identificación de pequeños dispositivos denominados etiquetas o tags RFID. (Vargas V, 2013, pág. 5)

Existen 3 componentes básicos en un sistema de RFID.

- ✓ **El tag o etiqueta de RFID**, consiste en un pequeño circuito integrado con una pequeña antena, capaz de transmitir un número de serie único hasta un dispositivo de lectura, como respuesta a una petición. Algunas veces puede incluir una batería.
- ✓ **El lector**, (el cual puede ser de lectura o lectura/escritura) esté compuesto por una antena, un módulo electrónico de radiofrecuencia y un módulo electrónico de control.
- ✓ **Un controlador o un equipo anfitrión**, comúnmente una PC o Workstation, en la cual corre una base de datos y algún software de control y seguimiento.

**Figura 1. Podemos ver la estructura de la tecnología RFID**



**Fuente:** (Vargas V, 2013, pág. 5)

### 2.3.2. Arquitectura de un sistema RFID

Himanshu Bhatt y Bill Glover, autores de “RFID Essentials”, uno de los libros más usados a nivel mundial por personas que desean informarse y/o documentarse sobre la tecnología RFID, definen arquitectura como: “la descomposición de un sistema particular en componentes individuales para demostrar como dichos componentes trabajan juntos para lograr los requerimientos exigidos por el sistema complejo”. Tomando en cuenta esto, se puede deducir que no existe una arquitectura RFID de carácter universal que encaje con todos los requerimientos de cualquier técnico, ingeniero o trabajador en área para satisfacer las necesidades de cualquier trabajo. (Nava J, 2013, pág. 17)

Por lo tanto, se puede afirmar que dependiendo de la implementación que se le quiera dar al sistema RFID, su arquitectura será distinta. Sin embargo, un sistema RFID debería poder satisfacer o cumplir con algunas de las siguientes características y capacidades;

- ✓ La capacidad de codificar etiquetas o tags RFID
- ✓ La capacidad de adjuntar etiquetas codificadas a algún artículo
- ✓ La capacidad de rastrear el movimiento de los artículos etiquetados
- ✓ La capacidad de integrar información RFID en aplicaciones de empresas
- ✓ La capacidad de producir información que puede ser compartida entre empresas.
- ✓ La capacidad para desarrollar la auto-organización de dispositivos inteligentes.

El sistema RFID que se desarrollara, incluirá los siguientes componentes, los cuales se estudiarán a profundidad a lo largo del trabajo:

- ✓ Etiqueta o tags

- Pasivas
  - Activas
  - Semi- pasivas
- ✓ Lector de RFID
  - ✓ RFID Middleware

Básicamente, las etiquetas pueden ser programadas para llevar cualquier información dentro de ellas, ya sea el tipo de artículo a que este adherida, un historial de los puntos por los que ha pasado, la localización del mismo y mucho más, todo depende del tipo de uso que se le quiera dar. El lector se encarga de recibir la información de la etiqueta mediante un proceso de llamado y escucha recibiendo ayuda de una antena. Esta información enviada será posteriormente enviada a un sistema intermedio llamado RFID Middleware.

Ambos dispositivos previamente mencionados tienen antenas, estas son las que realizan mediante radio frecuencia el proceso de intercambio de información entre ellos, por medio de un acoplamiento.

Cuando se planea la arquitectura de un sistema RFID, el proceso clave incluye seleccionar un esquema de numeración de objetos, codificados de etiquetas, adjuntar las etiquetas a los objetos que se desea rastrear, leer los objetos mientras se mueve a través de un ambiente físico, enviar dicha información a una interfaz lógica comprendida por un usuario final y gestionar los datos de la mejor manera para el desarrollo de la empresa en cuestión. (Nava J, 2013, pág. 18)

### **2.3.3. Etiquetas o tags**

Una etiqueta o Tag RFID, cumple con varios propósitos, pero el principal de todos ellos es adjuntar físicamente a un objeto, datos relevantes sobre el mismo objeto. Debido a que las etiquetas llevan dicha información dentro de ellas, deben tener la capacidad para transmitírsela a algún dispositivo interesado en recibirla, por lo que

todas tienen algún mecanismo interno para almacenar estos datos y algún otro mecanismo para poder comunicarlos. Estas etiquetas pueden estar contenidas dentro de pequeños botones plásticos, capsulas de vidrio, etiquetas de papel, e incluso cajas metálicas; también pudieran estar adjuntadas a un paquete, integrados a un animal o ser humano escondido en la cabecera de una llave o perforadas en alguna prenda.

Están compuestas por dos partes básicas:

- ✓ El Chip o Circuito Integrado: Almacena una serie de números únicos para ese chip. También contiene lógica y algoritmos sobre cómo reaccionar si se encuentra en frente de un lector.
- ✓ La antena: Permite al chip recibir poder y comunicarse, logrando de esta manera el intercambio de información entre el lector y la etiqueta.

Las etiquetas RFID se comunican con el mundo enviando y recibiendo Señales Electromagnéticas. Debido a que existen varios tipos de tags, debe conocerse el mecanismo que utiliza cada una parte para comunicarse. Los tags activos utilizan baterías para dar poder a sus comunicaciones, las pasivas reflejan la señal enviada por el lector y la semi- pasiva, tienen un mecanismo interno de poder, el cual les permite desempeñar acciones de etiquetas activas, y al mismo tiempo, utilizar la energía generada por el lector para comunicarse. (Nava J, 2013, pág. 19)

#### **2.3.4. Lectores de RFID**

El lector de RFID es un dispositivo que puede leer y escribir datos hacia tags RFID compatibles. El lector es el componente central del hardware en un sistema de RFID y tiene los siguientes componentes:

- ✓ **Transmisor:** El transmisor emite potencia y envía el ciclo de reloj a través de sus antenas hacia los tags que se encuentran dentro de su rango de lectura.

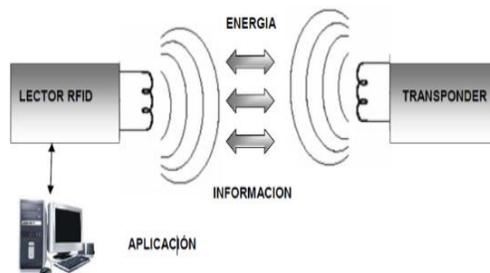
- ✓ **Receptor:** Este componente recibe las señales analógicas provenientes del tag a través de la antena y envía estos datos al microprocesador, donde esta información es convertida en su equivalente digital.
- ✓ **Antena:** Esta antena va conectada directamente al transmisor y al receptor. Existen lectores con múltiples puertos para antenas, lo que les permite tener múltiples antenas y extender su cobertura.
- ✓ **Microprocesador:** Este componente es responsable de implementar el protocolo de lectura enviada empleado para comunicarse con tag compatibles. Decodifica y realiza verificación de errores a las señales recibidas. Adicionalmente, puede contener cierta lógica para realizar filtrado y procesamiento de bajo nivel de los datos leídos, esto es, eliminar lecturas duplicadas o erróneas.
- ✓ **Memoria:** La memoria es utilizada para almacenar información como los parámetros de configuración del lector, además de una lista de las últimas lecturas realizadas, de modo tal que, si se pierde comunicación con la PC, no se pierdan todos los datos.
- ✓ **Controlador:** El controlador es el componente que permite a una entidad externa, sea un humano o un software de computadora, comunicarse y controlar las funciones del lector.
- ✓ **Interfaz de Comunicación:** Esta Interfaz provee las instrucciones de comunicación, que permiten la interacción con entidades externas, mediante el controlador, para transferir datos y recibir comandos. Un lector puede tener distintos tipos de interfaz como se discute más adelante.
- ✓ **Fuente de Alimentación:** Este componente provee de alimentación eléctrica a los componentes del lector y regularmente consiste en un cable con un adaptador de voltaje, conectado hacia la toma de corriente. Pero en los últimos años se ha incrementado el número de lectores de tipo pistola, los cuales son móviles y su fuente de alimentación es una batería recargable.

### 2.3.5. Principio de funcionamiento

Como hemos visto, existe una gran diversidad de sistemas RFID, los cuales pueden satisfacer un amplio abanico de aplicaciones para los que pueden ser utilizados. Sin embargo, a pesar de que los aspectos tecnológicos pueden varias, todos se basan en el mismo principio de funcionamiento, que se describe a continuación.

- ✓ Se equipa a todos los objetos a identifica, controlar o seguir, con una etiqueta RFID.
- ✓ La antena del lector o interrogador emite un campo de radiofrecuencia que las activa las etiquetas.
- ✓ Cuando una etiqueta ingresa en dicho campo utiliza la energía y la referencia temporal recibidas para realizar la transmisión de los datos almacenados en su memoria. En el caso de etiquetas activas la energía necesaria para la transmisión proviene de la batería de la propia etiqueta.
- ✓ El lector recibe los datos y los envía al ordenador de control para su procesamiento.

**Figura 2. Esquema de un sistema RFID**



**Fuente:** (Rodríguez H., 2009, pág. 19)

### 2.3.6. Software de enlace

Para el sistema RFID, es necesario una plataforma de software para la captura y gestión inteligente de datos. Una vez que el lector captura la información brindada por los tags, estos datos son enviados a un programa que produce interpretar u traducir a un lenguaje entendible para el hombre.

## **2.4. Metodología de desarrollo**

Cabe comentar que no es siempre conveniente escoger y aplicar una metodología de forma estricta. Es importante entenderla y conocer que nos puede aportar a nuestro proyecto, para aplicarla en las fases o procesos en los que nuestro equipo o nuestros usuarios estén más cómodos con ella y no al revés.

Sin embargo, valoran enormemente las facilidades de gestión que ofrezca el proyecto, el acceso a su documentación, la cantidad de la misma, etc., por lo que estarán mucho más motivados y pasaran más tiempo colaborando en el proyecto que entendiéndolo. De la misma manera, determinados colaboradores o clientes rechazarán una metodología demasiado moderna o flexible por desconocimiento. (Aycart Pérez, Golbert G., Hernández M. & Mas Hernández, 2007, pág. 28)

### **2.4.1. Metodologías Web**

El desarrollo de aplicaciones Web abarca decisiones no triviales de diseño e implementación que inevitablemente influyen en todo el proceso de desarrollo, afectando la división de tareas. Los problemas involucrados, como el diseño del modelo del dominio, modelo navegacional y la construcción de la interfaz de usuario, tienen requerimientos disjuntos que deben ser tratados por separado.

A partir de esta separación de intereses, surgen las metodologías de desarrollo de aplicaciones Web que permiten especificar los requerimientos atacando cada uno de sus aspectos más importantes: el modelo conceptual, navegacional y de interfaz de usuario. El modelo conceptual define cuáles serán los conceptos/objetos del negocio que serían manipulados en la aplicación.

El modelo navegacional permite describir que información será presentada usualmente agrupada en un Nodo y de qué forma se interactúa con esta información a partir de las relaciones conceptuales; un nodo, por ejemplo, indica el criterio con el que se mostrarán los objetos de negocio. Finalmente, el modelo de interfaz de usuario especifica de qué forma se presentará la información al usuario y como este percibirá en términos de eventos visuales.

Las metodologías de Web maduras tal como HDM, OOHDM, UWE, entre otras son ejemplos de metodologías que faciliten el diseño de una aplicación Web abarcando los aspectos (conceptual, navegacional e interfaz de usuario).

#### **2.4.2. Metodología OOHDM**

OOHDM (Modelo de Diseño de Hipermedia Orientado a Objetos – Object Oriented Hypermedia Desing Model), fue creado en 1996 por Daniel Schwabe (Pontificia Universidad de Católica de Río de Janeiro) y Gustavo Rossi (Universidad Nacional de la Plata). En OOHDM se modela la a navegación a través del diagrama de clases navegacionales y del diagrama de contextos.

Según los autores la metodología OOHDM, tiene similitud en sus características con la HDM con la única diferencia de que tiene un proceso que indica las actividades a ejecutar y el producto o entregable que debe hacerse al finalizar una fase. Este método toma como punto de partida el modelo de clases obtenido durante la primera fase del desarrollo de software denominado modelo conceptual, además permite modelar aplicaciones de grandes tamaños o con grandes volúmenes de información y pueden ser usados en diversos tipos de aplicaciones navegables, sitios Web, sistemas de información o presentaciones multimedia.

OOHDM que propone un proceso de desarrollo de cinco fases donde se combinan notaciones graficas UML con las otras propias de la metodología. Además, permite desarrollar aplicaciones Web a partir de la utilización de modelos especializados como: conceptual, navegación e interfaz de usuario teniendo como objetivo simplificar y hacer más eficaz el diseño de aplicaciones. (Jimmy R. Molina, Mariuxi P. Zea, María J. Contenido y Fabricio G. García, 2018, pag.8)

**Figura 3. Fases de la metodología OOHDM**



**Fuente:** (*Object-Oriented Design Structures in Web Application Models* (2002)).

#### **2.4.2.1. Recolección de Requerimientos**

La herramienta en la cual se fundamenta esta fase son los diagramas de casos de usos, los cuales diseñados por escenarios con la finalidad de obtener de manera clara los requerimientos y acciones del sistema. (German, 2003, pág. 15)

Lo primero que todo es necesario la recopilación de requerimientos. En este punto, se hacen identificar los actores y las tareas que ellos deben realizar. Luego, se determinan escenarios para cada tarea y tipo de actor.

Los casos de uso que surgen a partir de aquí, serán luego representados mediante los Diagramas de Interacción de Usuario (UIDs), los cuales proveen de una representación gráfica concisa de la interacción entre el usuario y el sistema durante la ejecución de alguna tarea. Con este tipo de diagramas se capturan los requisitos de la aplicación de manera independiente de la implementación. Esta es una de las fases más importantes, debido a que es aquí donde se realiza la recogida de datos. (Díaz Pulido, José Arturo, 2017, pág. 28)

#### **2.4.2.2. Diseño conceptual**

Se construye un modelo orientado a objetos que represente el dominio de la aplicación usando las técnicas propias de la orientación a objetos. La finalidad principal durante esta fase es capturar el dominio semántico de la aplicación teniendo en cuenta el papel de los usuarios y las tareas que desarrollan.

OOHDM utiliza el meta-modelo de clases de UML, con pequeñas extensiones, para expresar el diseño conceptual (Koch, 2002).

### 2.4.2.3. Diseño Navegacional

La estructura de navegación de una aplicación hipermedia está definida por un esquema de clases de navegación específica, que refleja una posible vista elegida.

En OOHDM hay una serie de clases especiales predefinidas, que se conocen como clases navegacionales:

- ✓ **Nodos:** Se definen como vistas orientadas a objeto de las clases definidas durante el diseño conceptual usando un lenguaje predefinido y muy intuitivo, permitiendo así que un nodo sea definido mediante la combinación de atributos de clases diferentes relacionadas en el modelo de diseño conceptual. Los nodos contendrán atributos de tipos básicos y enlaces.
- ✓ **Enlaces:** Los enlaces reflejan la relación de navegación que puede explorar el usuario.
- ✓ **Estructura de acceso:** Las estructuras de acceso actúan como índices o diccionarios que pueden al usuario encontrar de forma rápida y eficiente la información deseada. Los menús, los índices o las guías de rutas son ejemplos de esta estructura. Las estructuras de acceso también se modelan como clases, compuestas por un conjunto de referencias a objetos que son accesibles desde allá y una serie de criterios de clasificación de las mismas.
- ✓ **Contexto Navegacional:** Para diseñar bien una aplicación hipermedia hay que prever los caminos que el usuario puede elegir, así es como únicamente podremos evitar información redundante o que el usuario pierda en la navegación. En OOHDM un contexto navegacional está compuesto por un conjunto de nodos, de enlaces, de clases de contexto y de otros contextos navegacionales. Estos son introducidos desde clases de Navegación (enlaces,

nodos o estructuras de acceso)., pudiendo ser definidas por extensión o de forma implícita.

- ✓ **Clase de Contexto:** Es otra clase especial que sirve para complementar la definición de una clase navegación. Por ejemplo, sirve para indicar que información esta accesible desde un enlace y desde donde se puede llegar a él. (Díaz Pulido, José Arturo, 2017, pág. 30-31)

La navegación no se encontraría definida sin el otro modelo que propone OOHDM el contexto navegaciones. Esto es la estructura de la presentación dentro de un determinado contexto. Los contextos navegaciones son uno de los puntos más criticados a OOHDM debido a su complejidad de expresión (Carrillo, 2010, pag.11).

#### **2.4.2.4. Diseño de interfaz Abstracta**

Definir la forma en la cual los objetos navegacionales pueden aparecer, de cómo los objetos de interfaz activaran la navegación y el resto de la funcionalidad de la aplicación, que transformaciones de la interfaz son pertinentes y cuando es necesario realizarlas. Consiste en definir: Qué objetos de interfaz va a percibir el usuario.

- ✓ El camino en el cuál aparecerán los diferentes objetos de navegación
- ✓ Qué objeto de interfaz actuarán en la navegación
- ✓ La forma de sincronización de los objetos multimedia y el interfaz de transformaciones

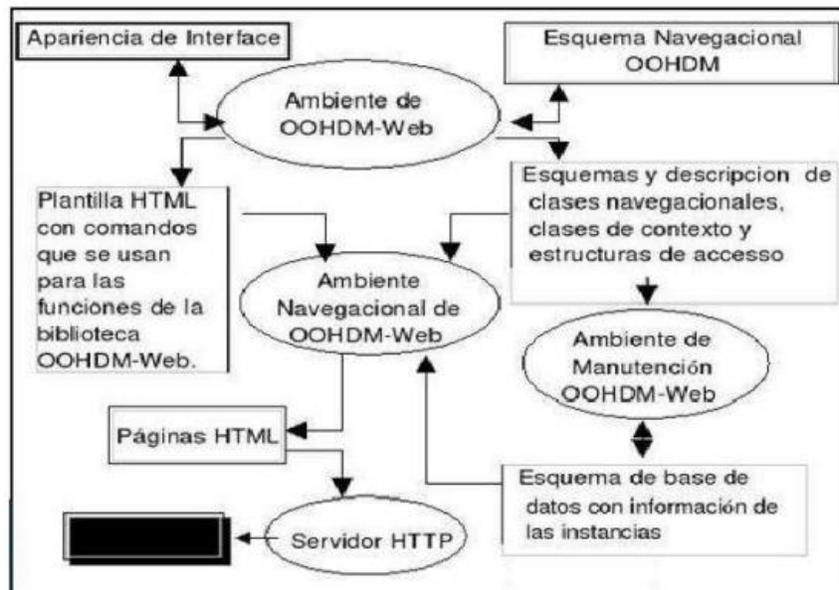
#### **2.4.2.5. Implementación**

En esta fase se tiene en cuenta el entorno particular en el cual se va a correr la aplicación. El diseñador ya tiene definido los artículos de información, componentes del sistema, que son parte del dominio del problema.

Un gran número de aplicaciones Web han sido desarrollados utilizando OOHDM como metodología y empleando una diversidad de tecnologías tal como Java (J2EE), ASPX (.NET), Lua (CGILua), Cold Fusion y Ruby (Ruby on Rails) para llevar a cabo la implementación (Carrillo, 2010, pág. 31).

En el comienzo de la implementación, el desarrollador de la plataforma determina cómo será almacenada la información, tanto de las etapas de modelado conceptual y estructura navegacional, definiendo también como trabajará la interfaz de la plataforma y el comportamiento que tendrá la plataforma en si al usar los lenguajes de programación como PHP y HTML; contemplando la posibilidad de necesitar extensiones o librerías que faciliten o mejoren el desarrollo de la aplicación, y de ser necesaria la asesoría de un profesional de diseño gráfico deberá documentarse adecuadamente su participación en el proyecto.

**Figura 4. Fase de implementación, integración con servidores**



**Fuente:** (*Object-Oriented Design Structures in Web Application Models* (2002))

Es importante mencionar que el desarrollador y diseñador son los encargados del lado técnico del sistema y su apariencia final, mientras que el cliente verifica que funcione correctamente como lo ha solicitado en la primera fase o etapa.

**Tabla 3. Productos y formalismos de la metodología OOHDM.**

ETAPAS	PRODUCTOS	FORMALISMO
Obtención de requerimientos	Casos de Uso (actores, escenarios)	Plantillas del formato del documento, Diagrama de Interacción de Usuario (UIDs).
Diseño conceptual	Clase, subsistemas, relaciones, atributos.	Modelos Orientados a Objetos.
Diseño navegacional	Nodos, enlaces, estructuras de acceso, contextos, navegacionales, transformaciones de navegación.	Vistas Orientados a Objetos, Cartas de navegación orientados a objetos, Clases de Contexto.
Diseño de interfaz Abstracta	Objetos de la interfaz abstracta respuestas a eventos externos, transformaciones de la interfaz.	Vistas abstractas de Datos (ADV), diagramas de Configuración Cartas de navegación de los ADVs.
Implementación	Aplicación en funcionamiento.	Los soportados por el entorno.

**Fuente:** (Jimmy R. Molina, Mariuxi P. Zea, María J. Contenido y Fabricio G. García)

Para finalizar se resalta que el método OOHDM aplica los principios de la programación orientada a objetos sin embargo no necesita de un entorno de implementación del tipo mencionado anteriormente.

### 2.4.3. Ventajas de OOHDM

La metodología OOHDM es practicada puesto que nos permite mirar de manera más amplia y precisa los diagramas y mediante estos se puede representar en forma

precisa elementos propios de las aplicaciones hipermedia, tales como nodos, vínculos, imágenes, estructuras de acceso y contexto.

OOHDM propone un conjunto de tareas que en principio pueden involucrar mayores costos de diseño, pero que a mediano y largo plazo reducen notablemente los tiempos de desarrollo al tener como objetivo principal la reusabilidad de diseño, y así simplificar la evolución y el mantenimiento.

OOHDM crea una cantidad más o menos grande de documentación a través de sus distintas etapas de desarrollo, lo que nos permite llevar un control del desarrollo de las etapas y tener la posibilidad real de realizar una rápida detección, corrección de errores y mantención. (Echeverría, 2009, pag.12).

## **2.5. Herramientas**

### **2.5.1. HTML 5**

HTML es el lenguaje de marcaje predominante para la creación de páginas web. Se utiliza para descubrir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos como imágenes. El HTML se escribe en forma de etiquetas entre corchetes angulares (<,>). El HTML también puede describir, hasta cierto punto, la apariencia de un documento y puede incluir scripts (por ejemplo, Java Script), lo cual puede afectar al comportamiento de navegadores web y otros procesadores de HTML.

Por convención, los archivos en formato HTML utilizan la extensión .htm o html.

El HTML se desarrolló como un subconjunto del SGML (un lenguaje de marcas más complejo) y es el lenguaje de marcas que ha hecho posible la web tal como la conocemos actualmente. El HTML 5 es la última versión de HTML recomendada por el W3C.

Como se vio, el HTML se definió dentro del marco de SGML y fue de lejos la aplicación más conocida de este estándar. De todos modos, los navegadores nunca han puesto muchas exigencias al código HTML que interpretan, las

páginas web pueden llegar a ser muy caóticos si no cumplen la sintaxis. Sin embargo, hay que decir que no todos los navegadores siguen estándares, lo cual significa que una misma web se puede visualizar de manera diferente según que navegador la interprete. (Barlona Auge, 2010, pág. 6)

### **2.5.2. Css**

Las hojas de estilo en cascada (cascading style sheets) son un mecanismo simple que nos describe como se muestra un documento en la pantalla del ordenador, o como se imprime o incluso como se pronuncia la información del documento en un dispositivo de lectura. Con las CCS, tendremos un control total sobre el estilo y el formato de nuestros documentos. Cualquier cambio realizado el estilo establecido para un elemento afectara a todas las páginas vinculadas a la CSS en las cuales aparezca este elemento.

La CCS funciona con declaraciones sobre el estilo de uno o más elementos. Por lo tanto, las hojas de estilo están compuestas por una o más declaraciones referidas a elementos de un XHTML (lenguaje de marcado de hipertexto entendido). Una declaración tiene dos partes: una propiedad y el valor de la propiedad. Hay diferentes maneras de dar estilo a un documento mediante y el valor de la propiedad. Hay diferentes maneras de dar estilos a un documento mediante CSS, aunque hay algunas que rompen con la premisa de separar el contenido de la presentación. La mejor manera de hacerlo y la más recomendable, es utilizando una hoja de estilo externo que se vinculara al documento mediante el elemento <link>, el cual tiene que estar situado dentro de la sección <head>. En este caso, la hoja de estilo estará en un archivo con extensión “.css” totalmente separado del “xhtml”. Lo cual nos permitirá que, al modificar el archivo. ccs, se modifiquen los estilos de todos los documentos donde está vinculado. (Barlona Auge, 2010, pág. 7-8)

### **2.5.3. JavaScript**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. (Javier Eguiluz, 2009, pág. 5)

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor. Sus usos en aplicaciones externas a la web.

JavaScript es un lenguaje con muchas posibilidades, utilizados para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos muchos más complejos. Con JavaScript podemos crear diferentes efectos e interactuar con nuestro usuario (Cárdenas, 2001, pág. 2-4)

### **2.5.4. PHP 5**

PHP corresponde a las siguientes iniciales de personal home page tools (herramientas para páginas iniciales personales). Es un lenguaje de programación tipo script para entornos web con unas funciones muy semejantes a las de ASP y JSP, utilizado, sobre todo, en servidores Linux para personalizar la información enviada a los usuarios que acceden a un sitio web. Desde un punto de vista técnico, es un

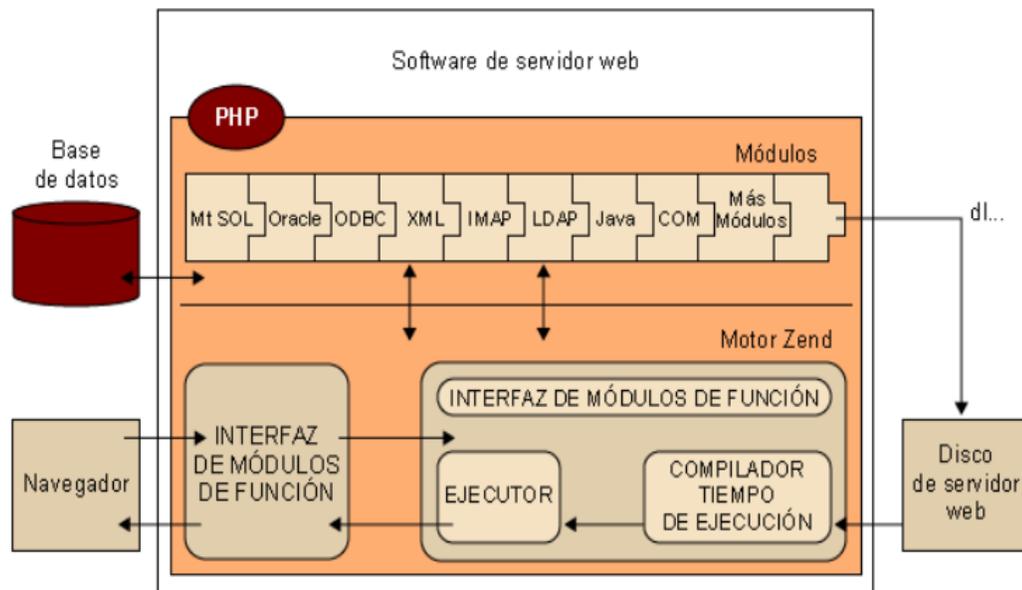
lenguaje interpretado de alto nivel, similar en construcciones léxicas y sintácticas a C, C++, Java y Perl, por lo que a quienes ya conozcan estos lenguajes les resultara muy fácil comenzar a escribir código PHP.

PHP es un lenguaje incrustado (embedded) en páginas HTML, es decir, es un lenguaje de programación que se introduce dentro de las páginas HTML. El código PHP se interpreta en el lado del servidor de web, desde donde se genera la página HTML solicitada antes de llevar a cabo su transmisión al navegador.

De esta forma, podemos programar aplicaciones asociadas al servidor de web, aumentando, así, la funcionalidad de dicho servidor y convirtiéndolo en sistema de desarrollo de aplicaciones cliente/servidor mucho más completo. Su principal objetivo es hacer que desarrolladores de aplicaciones basadas en la web puedan escribir páginas que se generan dinámicamente de una forma sencilla y rápida.

En la presente figura se aprecia la estructura interna con lo que se trabaja, el lenguaje de programación PHP, interactuando con el navegador, los servidores web y la base de datos.

**Figura 5. La estructura el lenguaje de programación PHP**



**Fuente:** (Berni Millet & Gil de la Iglesias, 2010)

PHP proporciona, por tanto, una gran facilidad para acceder a diferentes tipos de base de datos como Oracle, Sybase, MySQL, PostgreSQL, Adabas, etc. De hecho, es bastante sencillo portar una aplicación escrita con PHP para MySQL a cualquier otro servidor de base de datos, ya que las funciones de acceso que ofrece PHP son, en muchos casos, de sintaxis compartida. (Berni Millet & Gil de la Iglesias, 2010, pág. 7-8)

### **2.5.5. MySQL**

MySQL es un sistema de gestión de bases de datos (SGBD) SQL que en algunos aspectos es aproximadamente tan potente como Oracle (<http://www.oracle.com/>). Cabe mencionar que, a mediados del año 2009, Oracle, ha adquirido MySQL.

Sus principales objetivos han sido la velocidad y la robustez. Es un SGBD sencillo y rápido que se adapta perfectamente a entornos en los que el volumen de datos sea del orden de megabytes (en la documentación se habla de sus usos con base de datos de 50 millones de registros). En la versión 5 de MySQL ha incluido el control de transacciones, procedimientos almacenados y triggers, por lo que ha rellenado el gran hueco que lo diferenciaba de los grandes SGBD como Oracle. Si bien existe la posibilidad de comprar su soporte, Oracle sigue teniendo más aceptación en el mundo empresarial.

MySQL también puede verse como un conjunto de aplicaciones o pluggins funcionando en conjunto. También se encuentra, igual que PHP, en fases de pleno desarrollo: se están publicando con regularidad nuevas versiones del sistema, así como herramientas que son básicas en cualquier SGBD actual. (Berni Millet & Gil de la Iglesias, 2010, pág. 9-10)

- ✓ Dispositivo JDBC para acceder desde Java
- ✓ Dispositivo ODBC para acceder utilizando la API ODBC
- ✓ API de programación para C, Perl, C++, Python y TCL.
- ✓ Acceso desde PHP

- ✓ Entornos visuales de gestión de la base de datos
- ✓ Control de acceso basado en una base de datos de administración.

### 2.5.6. Laravel

Laravel es un Framework de código abierto para desarrollar aplicaciones y servicios web con PHP5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el “código espagueti”. Tiene como objetivo ser un Framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros Frameworks y aprovechar las características de las últimas versiones de PHP. (Otwell, 2003, pág. 15)

Laravel se enfoca en el modelo-vista-controlador (MVC) patrón de arquitectura, que impone una separación entre “lógica de negocio” de la lógica de entrada y presentación asociado con una interfaz gráfica de usuario (GUI) (Laravelbook, 2014, pag. 45).

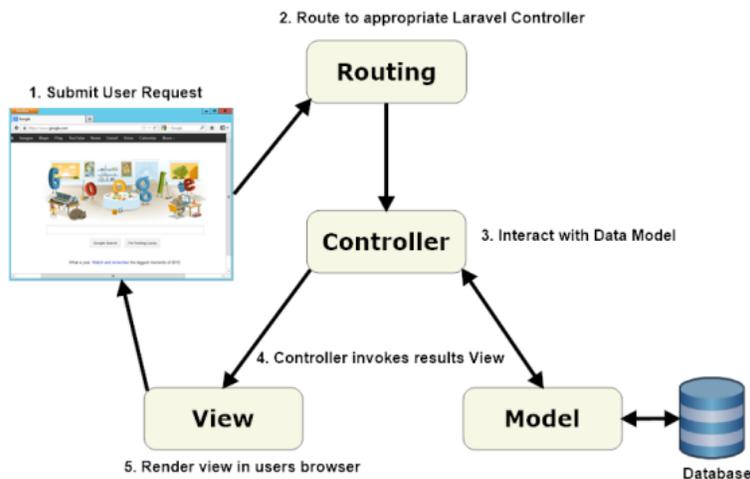
- ✓ **Capa Modelo:** El modelo es el componente responsable de la comunicación de la base de datos. Es aquí donde interviene Eloquent. Obviamente el uso de Eloquent es opcional, ya que también este framework dispone de algunos recursos que ayuda interactuar con los datos, o específicamente la creación de modelos.
- ✓ **Capa Vista:** Laravel, usa su propio motor de plantillas llamado Blade, se enfoca en dar un código limpio y fácil de comprender en las Vistas, además incluye un sistema de Caché esto ayuda a realizar las tareas mucho más rápido. Para la creación de estas plantillas se crea un archivo dentro de la carpeta /views con el *nombrevista.blade.php*.
- ✓ **Capa Controlador:** Es donde la lógica de negocio se encuentra y realiza las funcionalidades como: recuperar todas las entradas de la base de datos para enumerarlas, actualizar, eliminar o realizar búsquedas de la información en las

tablas de la base de datos dado un identificador, añadir nuevos registros a la base de datos y crear mensajes de confirmación de las funciones básicas. (Cogneau, 2012, pág. 11-36).

### ✓ Forma de petición típica Laravel

El navegador envía una solicitud, la cual es recibida por un servidor web y se transmite al motor de enrutamiento laravel. El router laravel recibe la petición y redirige al método de la clase controlador apropiado basado en el patrón de URL de enrutamiento.

**Figura 6. Petición típica MVC en laravel**



**Fuente:** (Laravelbook, 2014).

### ✓ Eloquent ORM

Un ORM (Object Relational Mapper) en PHP es un software que permite tratar la capa de persistencia de los datos, como simples accesos a métodos de una Clase u Objeto en PHP. La funcionalidad interna del ORM es mapear los objetos de PHP a las tablas en la base de datos (Patricio, 2013).

### 2.5.7. Boststrap

Bootstrap se creó en el 2011 como una ayuda interna para solucionar las inconsistencias en el desarrollo dentro del equipo de ingeniería de Twitter. Además, fue originalmente desarrollada por ingenieros en Twitter: Mark Otto y Jacob. (Anónimo, 2014)

Permite diseñar interfaces web utilizando CSS y JavaScript, su principal función es lade adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo, se conoce como “responsive design(diseño responsivo)” o diseño adaptativo (Solis, 2014).

El diseño responsivo es un método para tomar todo el contenido existente que está en la página y optimizarlo para el dispositivo que lo está viendo. Por ejemplo, en la PC está adaptada para visualizar en pantalla panorámica ya que optimiza para las pantallas más grandes. Las tablets consiguen un diseño optimizado, aprovechando sus diseños de retrato o paisaje. Y teléfonos, puede orientar su ancho mucho más estrecho (Spurlock, 2013).

#### 2.5.7.1. Características de Bootstrap

Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tablets y móviles a distintas escalas y resoluciones.

- ✓ Se integra perfectamente con las principales librerías JavaScript, por ejemplo, JQuery.
- ✓ Ofrece un diseño sólido usando estándares como CSS3/HTML5.
- ✓ Es un framework ligero que se integra de forma limpia en nuestro proyecto actual.
- ✓ Funciona con todos los navegadores, incluido Internet Explorer usando HTML Shimpara que reconozca los tags HTML5.

- ✓ Dispone de distintos layout predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.

## **2.6. Seguridad de la Información**

### **2.7.1. Norma de calidad Software ISO/IEC 27001**

ISO 27001 es una internacional que permite el aseguramiento, la confiabilidad e integridad de los datos y de la información, así como de los sistemas que la procesan.

El estándar ISO 27001:2013 para los Sistemas Gestión de la Seguridad de la Información permite a las organizaciones la evaluación del riesgo y la aplicación de los controles necesarios para mitigarlos o eliminarlos.

La gestión de seguridad de la información se complementa con las buenas prácticas o controles establecidos en la norma ISO 27002.

#### **2.7.1.1. Seguridad Física**

La seguridad física es uno de los aspectos más olvidados a la hora del diseño de un sistema informático. Si bien algunos de los aspectos tratados a continuación que intenta a acceder físicamente a una sala de operaciones de la misma (Borghello, 2001)

Esto puede derivar en que para un atacante sea más fácil lograr tomar y copiar una cinta de la sala, que intentar acceder vía lógica a la misma.

#### **2.7.1.2. Seguridad Lógica**

La seguridad lógica consiste en la “aplicación de barreras y procedimientos que resguarden el acceso a los datos y solo se permita acceder a ellos a las personas autorizadas para hacerlo. “Existe un viejo dicho en la seguridad informática que dicta que “todo lo que no está permitido debe estar prohibido” y esto es lo que debe asegurar la Seguridad Logica (Borghello, 2001).

## **2.8. Métricas de calidad de software**

El objetivo principal de la ingeniería de Software es producir un producto de alta calidad. Para lograr este objetivo, los ingenieros de software deben utilizar mediciones que evalúen la calidad de análisis y los modelos de desafío, el código fuente y los casos de prueba que se han creado al aplicar la ingeniería de software, para lograr esta evaluación de calidad con objetividad, no con subjetividad.

En general, la medición persigue tres objetivos fundamentales: ayudarnos a entender que ocurre durante el desarrollo y el mantenimiento, permitirnos controlar que es lo que ocurre en nuestros proyectos y poder mejorar nuestros productos. (Fenton y Pfeegeer, 1997)

### **2.8.1. Calidad de software**

Incluso los desarrolladores de software más experimentados estarán de acuerdo en obtener software de alta calidad es una meta importante. Pero ¿Cómo se define la calidad de software? En el sentido más general se define como: Proceso eficaz de software que se aplica d manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan. (Roger Pressman, 2005, pág. 340)

### **2.8.2. Estándar ISO/25000**

Los aspectos más importantes en el desarrollo de software son la calidad del producto y del proceso. ISO/IEC 25000, proporciona una guía para el uso de las nuevas series de estándares internacionales, llamados Requisitos y Evaluación de Calidad de Productos de software (SQuaRE). Constituyen una serie de normas basadas en la ISO/IEC 9126 y la ISO/IEC 14598 y su objetivo principal es guiar el desarrollo de los productos de software con la especificación de requisitos de calidad. (ISO/25000, 2021).

**Figura 7. División ISO/IEC 25000**



Fuente: (ISO/25000, 2021)

**2.8.2.1. ISO/IEC 25010**

La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas.

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura:

**Figura 8. Descripción de las características de la ISO/IEC 25010**



Fuente: (ISO/25000, 2021)

## ✓ **Adecuación Funcional**

Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Completitud funcional. Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
- Corrección funcional. Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
- Pertinencia funcional. Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

## ✓ **Eficiencia de desempeño**

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Comportamiento temporal. Los tiempos de respuesta y procesamiento y los ratios de throughput de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas (benchmark) establecido.
- Utilización de recursos. Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
- Capacidad. Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

## ✓ **Compatibilidad**

Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Coexistencia. Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.
- Interoperabilidad. Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

## ✓ **Usabilidad**

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Capacidad para reconocer su adecuación. Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- Capacidad de aprendizaje. Capacidad del producto que permite al usuario aprender su aplicación.
- Capacidad para ser usado. Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- Protección contra errores de usuario. Capacidad del sistema para proteger a los usuarios de hacer errores.
- Estética de la interfaz de usuario. Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.

- Accesibilidad. Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

✓ **Fiabilidad**

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Madurez. Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- Disponibilidad. Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- Tolerancia a fallos. Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.
- Capacidad de recuperación. Capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

✓ **Seguridad**

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Confidencialidad. Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
- Integridad. Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.

- No repudio. Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
- Responsabilidad. Capacidad de rastrear de forma inequívoca las acciones de una entidad.
- Autenticidad. Capacidad de demostrar la identidad de un sujeto o un recurso.

✓ **Mantenibilidad**

Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Modularidad. Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- Reusabilidad. Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.
- Analizabilidad. Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
- Capacidad para ser modificado. Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- Capacidad para ser probado. Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

## ✓ **Portabilidad**

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Adaptabilidad. Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- Capacidad para ser instalado. Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
- Capacidad para ser reemplazado. Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno. Fuente: (ISO/25000, 2021, pág. 1-3)

## **2.9. Modelo de Costos COMOMO II**

Una de las tareas de mayor importancia en la planificación de proyectos de software es la estimación, la cual consiste en determinar, con cierto grado de certeza, los recursos de hardware y software, costo, tiempo y esfuerzo necesarios para el desarrollo del mismo.

COCOMO II, este modelo permite realizar estimaciones en función del tamaño del software y de un conjunto de factores de costo y de escala, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto. (SOMMERVILLE, 2005)

COCOMO II define tres modos de desarrollo:

- ✓ **Orgánico:** Proyectos relativamente sencillos, menores de 50 KDCL líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.

- ✓ Semi-acoplado: Proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en serie tipo de proyectos es variable y las restricciones intermedias.
- ✓ Empotrado: Proyectos bastantes complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad

**Tabla 4. Coeficiente del modelo COCOMO II**

PROYECTO DE SOFTWARE	a	b	c	d
Orgánico	2.4	1.05	2.5	0.38
Semi-acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

**Fuente:** (Pressman: 2005)

✓ **Factores Multiplicadores de Esfuerzo**

Cada factor se cuantifica para un entorno de proyecto. La escala es muy baja - bajo – nominal – alto – muy alto – extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula (por ejemplo, si para un proyecto el atributo *DATA* es calificado como *muy alto*, el resultado de la fórmula debe ser multiplicado por 1000).

El significado de los atributos es el siguiente, según su tipo:

- **De software**

- **RELY:** garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto. Va desde la sola inconveniencia de corregir un fallo (*muy bajo*) hasta la posible pérdida de vidas humanas (*extremadamente alto*, software de alta criticidad).

- **DATA:** tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación:  $D/K$  donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código.
  - **CPLX:** representa la complejidad del producto.
- **De hardware**
- **TIME:** limitaciones en el porcentaje del uso de la CPU.
  - **STOR:** limitaciones en el porcentaje del uso de la memoria.
  - **VIRT:** volatilidad de la máquina virtual.
  - **TURN:** tiempo de respuesta requerido.
- **De personal**
- **ACAP:** calificación de los analistas.
  - **AEXP:** experiencia del personal en aplicaciones similares.
  - **PCAP:** calificación de los programadores.
  - **VEXP:** experiencia del personal en la máquina virtual.
  - **LEXP:** experiencia en el lenguaje de programación a usar.
- **De proyecto**
- **MODP:** uso de prácticas modernas de programación.
  - **TOOL:** uso de herramientas de desarrollo de software.
  - **SCED:** limitaciones en el cumplimiento de la planificación.

El valor de cada atributo, de acuerdo a su calificación:

**Tabla 5. Variables factor de ajustes del esfuerzo**

Factores (cost-drivers)	Valor de los factores					
	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra bajo
<b>Atributo del producto</b>						
Fiabilidad requerida	0,75	0,88	1,00	1,15	1,40	
Tamaño de la base de datos		0,94	1,00	1,08	1,16	
Complejidad del software	0,70	0,85	1,00	1,15	1,30	1,65
<b>Atributos del ordenador</b>						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria			1,00	1,06	1,30	1,66
Volatilidad del hardware		0,87	1,00	1,15	1,30	
Restricciones de tiempo de respuesta		0,87	1,00	1,07	1,15	
<b>Atributos del personal</b>						
Calidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia con el tipo de aplicación	1,29	1,13	1,00	0,91	0,82	
Experiencia con el hardware	1,21	1,10	1,00	0,90		
Experiencia con el lenguaje de programación	1,14	1,17	1,00	0,95		
Calidad de programadores	1,42	1,17	1,00	0,86	0,70	
<b>Atributos del proyecto</b>						
Técnicas modernas de programación	1,24	1,10	1,00	0,91	0,82	
Empleo de herramientas	1,24	1,10	1,00	0,91	0,83	
Restricciones a la duración del proyecto	1,23	1,08	1,00	1,04	1,10	
<b>TOTAL, <math>M(X_i)</math></b>						

**Fuente:** (IMB, Estimación del costo del software)

Y por otro lado existen diferentes modelos que define COCOMO:

- ✓ **MODELO 1** (COCOMO básico) calcula el esfuerzo y el coste del desarrollo en función del tamaño estimado del programa (LDC).

Se utiliza para una aproximación rápida al principio del ciclo de vida.

ESFUERZO:

$$E = a * KLDC^b$$

TIEMPO:

$$D = c E^d$$

Donde E es el esfuerzo aplicado en personas-mes, D es el tiempo de desarrollo en meses cronológicos y KLDC es el número estimado de Líneas de Código (en miles) distribuidas para el proyecto

- ✓ **MODELO 2** (COCOMO intermedio) calcula el esfuerzo y el coste en función del Tamaño estimado del programa y de un conjunto de “guías de coste” que incluyen una evaluación subjetiva del producto, hardware, personal y atributos del producto.

***Para el caso el modelo intermedio se usa en las fórmulas siguientes.***

Formula de esfuerzo:

$$E = a * kLDC^b * \prod_{i=1}^{n=15} M(Xi)$$

Donde:

**a y b** = representan los coeficientes COCOMO nivel intermedio

**E**= E s el esfuerzo aplicado en personas – mes

**KLDC** = Es el número estimación de Líneas de Código expresadas en miles distribuidas para el proyecto

$\prod_{i=1}^{n=15} M(Xi)$ = representa la productoria de los factores de costos (Cost-Drives) evaluados para el proyecto.

Donde E es el esfuerzo aplicado en personas–mes, KLDC es el número estimado de Líneas de Código (en miles) distribuidas para el proyecto.

- ✓ **MODELO 3** (COCOMO avanzado) incorpora las características del modelo 2 y evalúa el impacto de los  $\prod_{i=1}^{n=15} M(Xi)$  en cada fase del desarrollo.

### 2.9.1. Fórmulas de estimación:

#### i. Estimación de esfuerzo

$$E = a * kLDC^b * \prod_{i=1}^{n=15} M(Xi)$$

Donde:

**a y b** = representan los coeficientes COCOMO nivel intermedio

**E**= E s el esfuerzo aplicado en personas – mes

**KLDC** = Es el número estimación de Líneas de Código expresadas en miles distribuidas para el proyecto

$\prod_{i=1}^{n=15} M(Xi)$ = representa la productoria de los factores de costos (Cost-Drives) evaluados para el proyecto.

Donde E es el esfuerzo aplicado en personas–mes, KLDC es el número estimado de Líneas de Código (en miles) distribuidas para el proyecto.

**ii. Estimación del tiempo**

$$T = c * E^d \text{ Mes}$$

Donde:

C y d, representa los coeficientes COCOMO nivel intermedio

E, representa el esfuerzo.

**iii. Otras estimaciones**

**Número de personas requerido para el desarrollo**

$$NP = \frac{E}{T} \text{ Personas}$$

Donde:

*E*, es el esfuerzo, T es el tiempo

**Costo total del proyecto**

$$CT = NP * T * \text{Sueldo mes Bs/\$us}$$

Donde:

*NP*, es el número de personas, T es el tiempo

**Costo por línea de código**

$$LDC = \frac{CT}{kLDC} \text{ Bs/\$us}$$

Donde:

*CT*, es el costo total del proyecto

**KLDC**, líneas de código expresadas en miles

## **2.9.2. Ventajas y Desventajas**

Se han desarrollado varias técnicas de estimación para el desarrollo de software como establecer de antemano el ámbito del proyecto, usar las métricas del software (mediciones del pasado) como base para la realización de estimaciones y desglosar el proyecto en partes más pequeñas que se estiman individualmente.

## **2.10. Pruebas De Funcionamiento**

### **2.10.1. Caja Blanca**

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- ✓ Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- ✓ Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- ✓ Ejecuten todos los bucles en sus límites operacionales.
- ✓ Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican al software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

Cálculo de la complejidad Climática

$$V(G)=A-N+2$$

Donde:

A es el número de aristas y N es el número de nodos

En nuestro caso tenemos que A y N Entonces, se tiene que:

$$V(G) = A - N + 2 = *$$

### **2.10.2. Caja Negra**

Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca.

Muchos autores consideran que estas pruebas permiten encontrar:

- a.** Funciones incorrectas o ausentes.
- b.** Errores de interfaz.
- c.** Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- d.** Errores de rendimiento.
- e.** Errores de inicialización y terminación.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

# **CAPÍTULO III**

## **MARCO APLICATIVO**

---

### 3.1. Desarrollo de la Metodología OOHDM

#### 3.1.1. Recopilación de Requerimientos

La tarea de la recopilación de requerimientos es fundamental para que un sistema sea exitoso, para el siguiente proyecto se realizó las actividades que indica la siguiente tabla:

**Tabla 6. Tarea para obtención de requerimientos**

TAREA	CARACTERISTICA
Entrevista	Se realizó entrevistas en diferentes parques de la ciudad de La Paz, a los gerentes.
Observación	Se observó que en los diferentes parques se presentan problemas a la hora de brindar un mejor servicio para sus clientes.
Documentación	Encuestas.

**Fuente:** (Elaboración propia)

##### 3.1.1.1. Requerimiento de Hardware

PG: Parqueo Publico Gonzales

En esta tabla 7. están los requerimientos de hardware que son realmente necesarios para el parqueo.

**Tabla 7. Requerimiento de hardware**

ID REQUERIMIENTO	DESCRIPCION
PG -RH01	Servidor (opcional)
PG-RH02	Computadora de escritorio de Administración
PG-RH03	Impresora de Caja
PG-RH04	Lector y Tarjetas RFID (de escritura y lectura)
PG-RH05	Pantalla de control

**Fuente:** (Elaboración propia)

### 3.1.1.2. Requerimiento de Software

**Tabla 8. Requerimientos de software**

ID REQUERIMINETO	DESCRIPCION
PG-RS01	Sistema Operativo
PG-RS02	Gestor de base de datos
PG-RS03	2 navegador Web

**Fuente:** (Elaboración propia)

### 3.1.1.3. Funciones del Sistema

**Tabla 9. Funciones del sistema**

FUNCION DEL SISTEMA	FUNCION
Sistema de control y seguimiento	Brindar una mejor atención
Control	Brindar seguridad
Base de datos	Almacenar datos
Reportes	Generar reportes sobre las acciones, registros y estadistas.

**Fuente:** (Elaboración propia)

### 3.1.1.4. Modelo de Requerimientos del Sistema

#### 3.1.1.4.1. Requerimientos funcionales

Los requerimientos funcionales del sistema se enfocan principalmente en lo que realizara el sistema, en la siguiente tabla, se mostrara las características que necesita el sistema a partir de la información obtenida.

**Tabla 10. Requerimientos funcionales**

ROL	DESCRIPCION	FUNCION
R1	Gestión de usuarios	Se encarga de crear, editar usuarios, asignar roles.
R2	Registro de Propietarios	Permite registrar un nuevo propietario del vehículo, editar y eliminar.
R3	Registro de vehículos	Permite registrar un nuevo propietario del vehículo, editar y eliminar.
R4	Registro de ubicaciones	Permite mostrar si las ubicaciones están disponibles o no. (Mapeo)
R5	Lista de registros	Los registros se encuentran en cada formulario y se muestra.
R6	Registro de tarifa	Permite registrar una nueva tarifa, editar y eliminar.
R7	Registro de cobros	Se registra los cobros, mostrar y imprime factura.
R8	Reportes	Muestra los registros por fecha, día, tarifa y generar estadísticas.
R9	Control1	Visualiza en pantalla el ingreso, salida y ubicación para el vehículo

**Fuente:** (Elaboración propia)

#### 3.1.1.4.2. Requerimientos no funcionales

**Tabla 11. Requerimientos no funcionales**

ROL	FUNCIONES
R2-1	El sistema debe visualizarse y funcionar correctamente en cualquier navegador.
R2-2	Mantenimiento de los equipos.
R2-3	Tener lector y tarjetas RFID.
R2-4	Respaldo del servidor, para asegurar la disponibilidad.
R2-5	Soporte y mantenimiento periódico para asegurar un buen rendimiento del sistema

**Fuente:** (Elaboración propia)

### 3.1.1.5. Definición de actores

La definición de actores nos permite conocer a las personas que estarán involucradas en el parqueo público Gonzales, donde nos ayudara a formar los casos de uso. La siguiente tabla muestra la lista de actores, junto con una descripción de sus actividades relacionadas con el sistema.

**Tabla 12. Lista de Actores**

ACTOR	DESCRIPCION
 <b>Administrador</b>	<p>Tiene las siguientes funciones:</p> <p>Es el encargado de supervisar los usuarios auxiliares del parqueo también puede asignar o quitar roles.</p> <p>Este actor es el que está informado de todo, conoce y puede realizar todas las tareas, además tiene la responsabilidad de tener el control y seguimiento de las actividades del parqueo.</p>
 <b>Usuario Auxiliar</b>	<p>Tiene las siguientes funciones:</p> <p>Es el encargado del seguimiento de entradas y salidas de los vehículos, además puede registrar y generar reportes.</p>
 <b>Propietario</b>	<p>Tiene las siguientes funciones:</p> <p>El propietario del vehiculó debe traer sus documentos, para poder registrase y optar por su tarjeta RFID, para acceder al servicio del parqueo.</p>

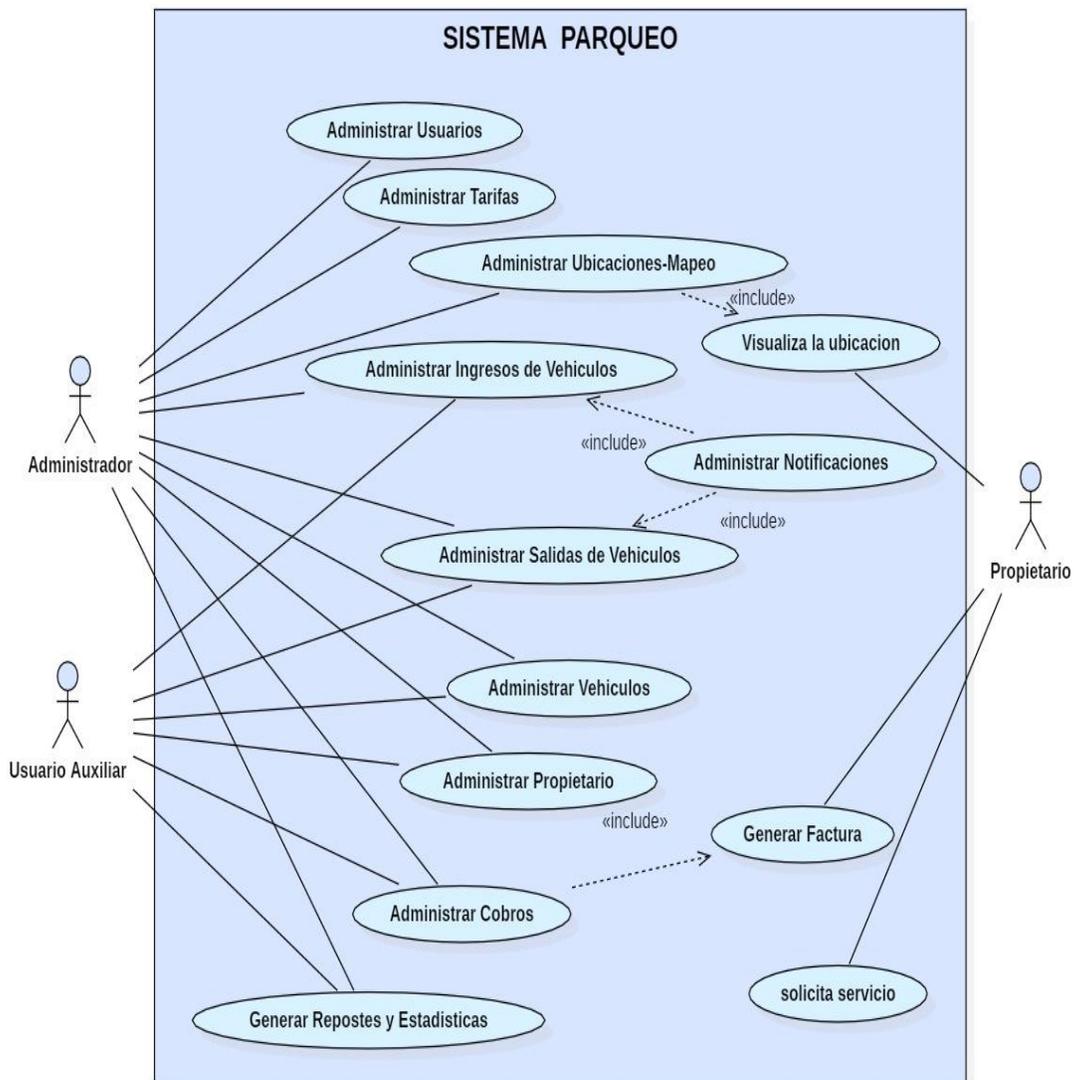
**Fuente:** (Elaboración propia)

### 3.1.1.6. Modelo de casos de uso

Para el modelado de casos de usos se realiza en base a los requerimientos del sistema mediante casos de uso, donde se describe el comportamiento del sistema frente a las acciones de los actores del mismo, así como el funcionamiento del sistema.

#### 3.1.1.6.1. Diagrama general de caso de uso del parqueo

**Figura 9. Diagrama general de caso de uso del parqueo**



**Fuente:** (Elaboración propia)

### 3.1.1.6.2. Diagrama casos de uso extendido

En el siguiente caso de uso se muestra los usuarios donde interactúan con el sistema y sus roles.

**Figura 10. Diagrama de caso de Uso: Gestión de Usuario**



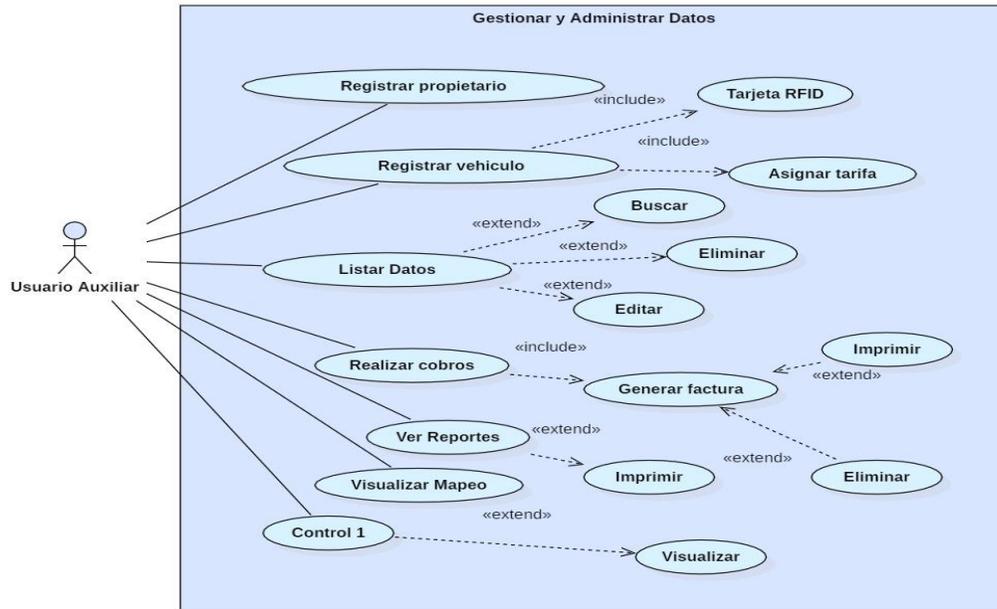
Fuente: (Elaboración propia)

**Tabla 13. Descripción de casos de uso: Gestión de usuario**

Gestión de usuario	
Actor	Administrador
Tipo	Primario Esencial
Descripción	El administrador registra a los usuarios y designa roles del sistema. Restringe el acceso al sistema habilitando y deshabilitando al usuario y restablece la contraseña. Además de encarga de todas las funciones del sistema de parqueo, donde se encuentran los menús de los módulos.

Fuente: (Elaboración propia)

**Figura 11. Diagrama de caso de uso: Gestionar y Administrar Datos**



Fuente: (Elaboración propia)

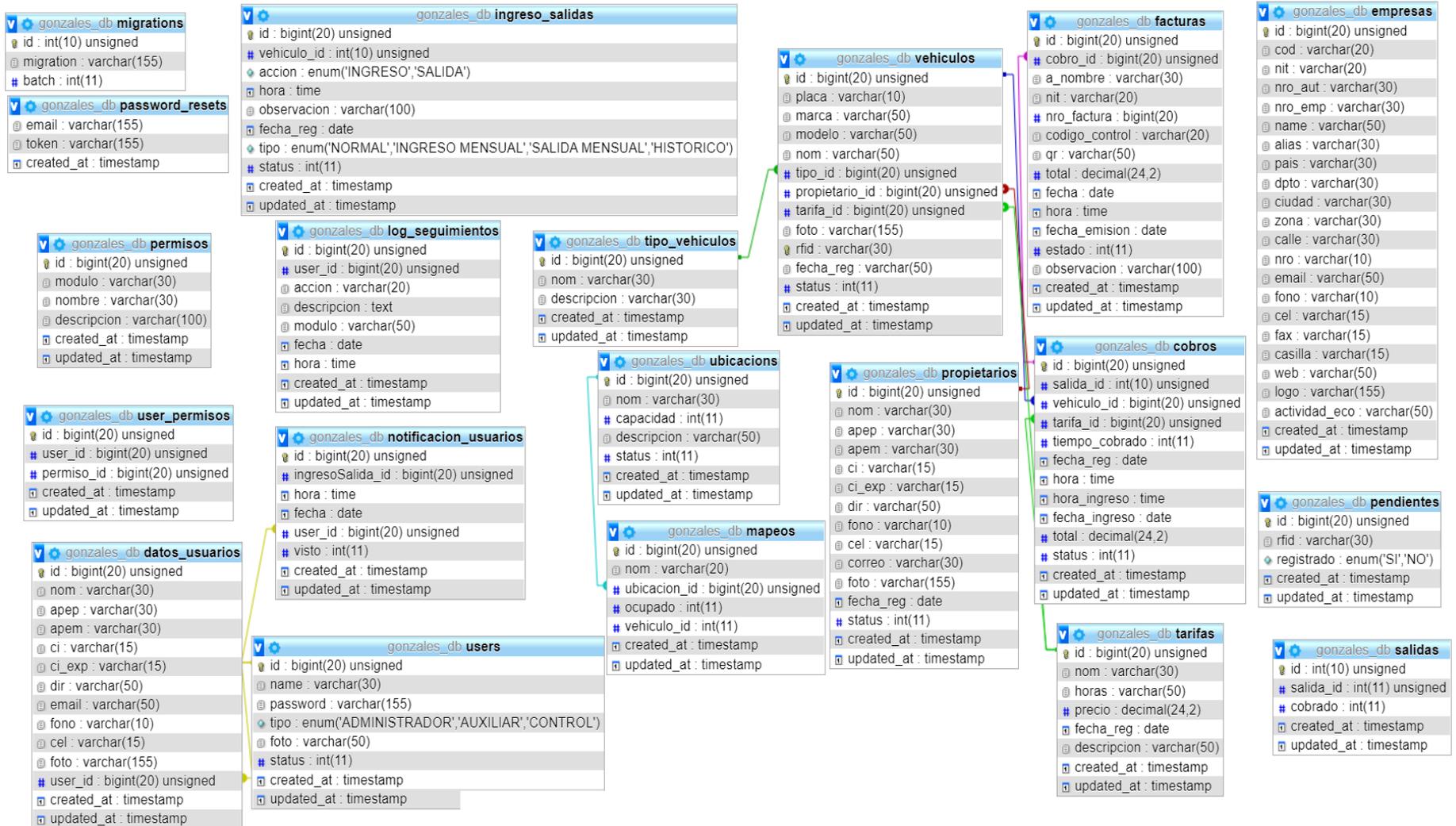
**Tabla 14. Descripción de casos de uso: Gestión y Administrar Datos**

Gestión y Administrar Datos	
Actor	Usuario Auxiliar
Tipo	Primario Esencial
Descripción	<p>El auxiliar se encarga de los registros de los datos de los propietarios, vehículos donde se puede eliminar.</p> <p>Realiza los cobros, genera reportes por registro, día, fecha y año.</p> <p>También visualiza el control para ver la notificación para el cliente y la búsqueda de información es rápida.</p>

Fuente: (Elaboración propia)

### 3.1.2. Diseño Conceptual

Figura 12. Diseño Conceptual



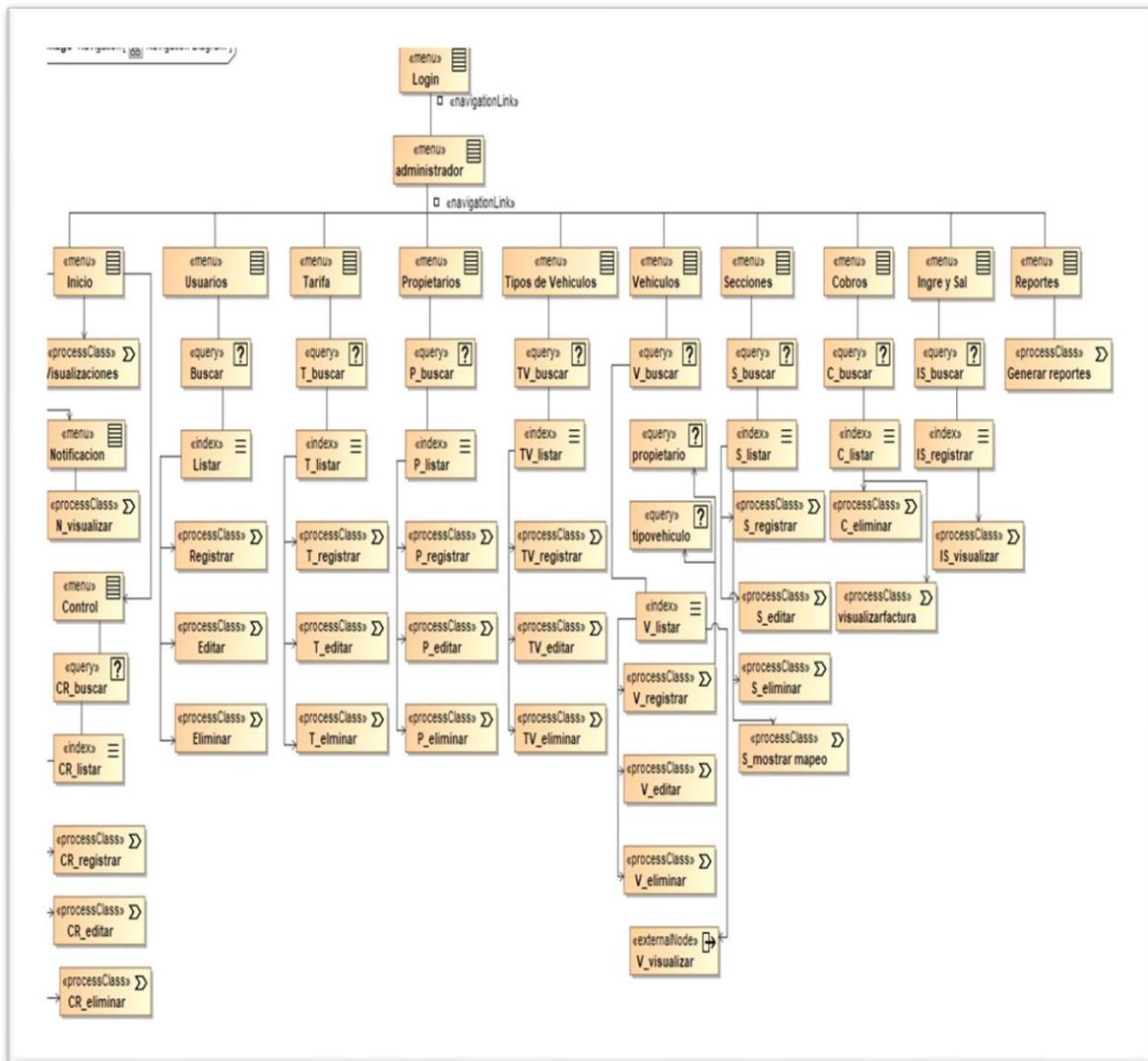
Fuente: (Elaboración propia)

### 3.1.3. Diseño Navegacional

En esta fase se realiza el modelo donde se visualiza la interacción de los usuarios en la navegación de sistema.

En la *figura 13*. Se puede ver el diagrama de navegación con el rol de usuario del administrador.

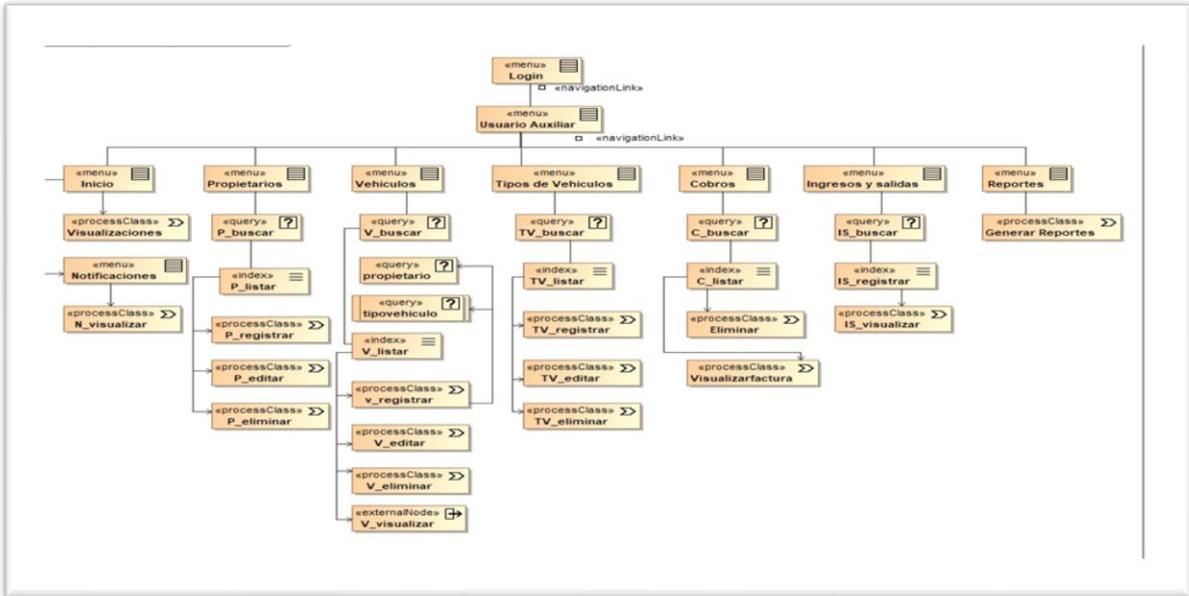
**Figura 13. Diagrama de navegación- Administrador**



Fuente: (Elaboración propia)

En la *figura 14*. Se puede ver el diagrama de navegación con el rol de usuario asignado por el administrador.

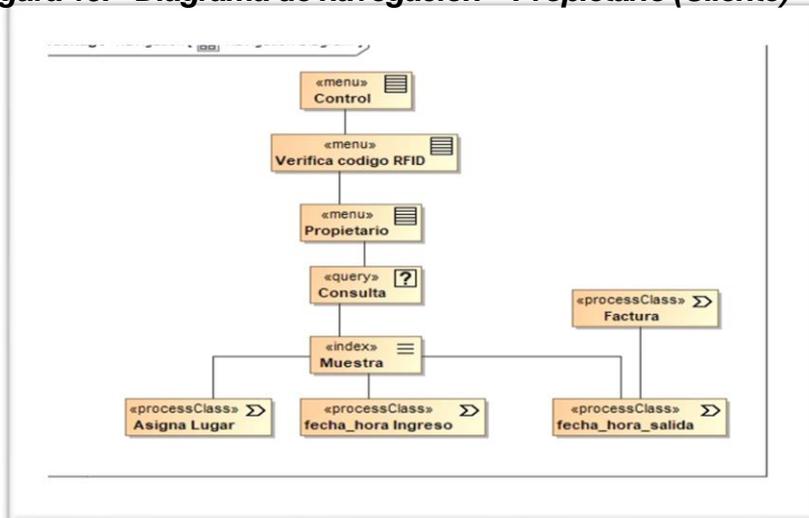
**Figura 14. Diagrama de navegación – Usuario Auxiliar**



Fuente: (Elaboración propia)

En la *figura 15*. Se puede ver el diagrama de navegación donde el propietario (Cliente) solo puede visualizar en pantalla.

**Figura 15. Diagrama de navegación – Propietario (Cliente)**

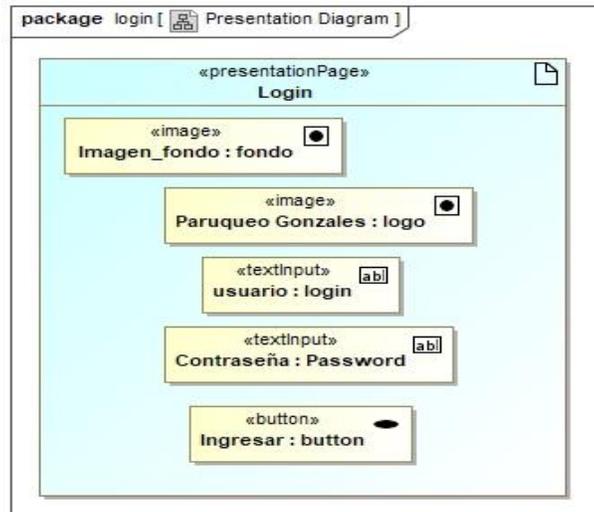


Fuente: (Elaboración propia)

### 3.1.4. Diseño de Interfaz Abstracta

En la *figura 16*. Se puede ver el diseño de Interfaz Abstracta del modelo de presentación del Login.

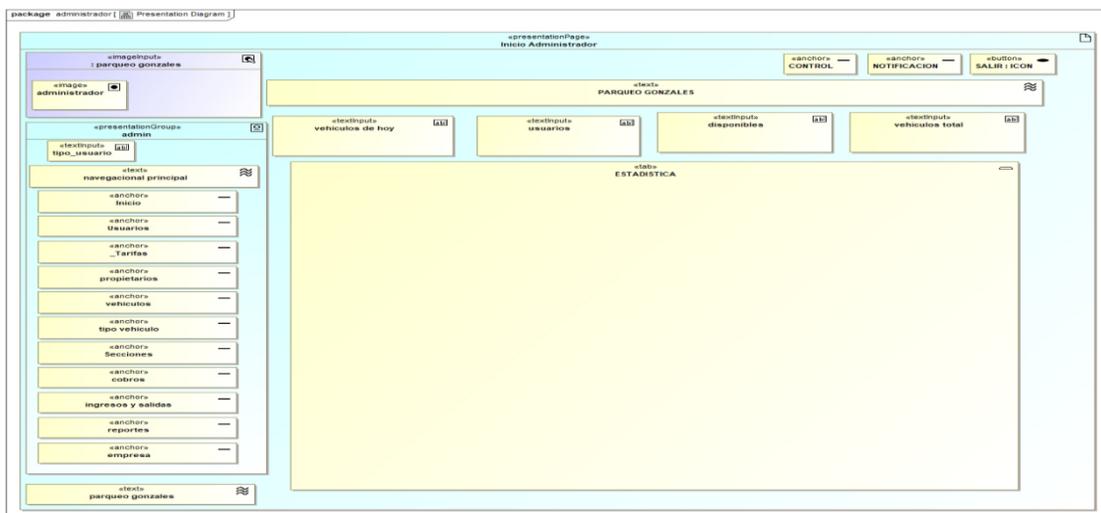
**Figura 16. Modelo de presentación Login**



Fuente: (Elaboración propia)

En la *figura 17*. Se puede ver el diseño de Interfaz Abstracta del Modelo de presentación página principal del Administrador.

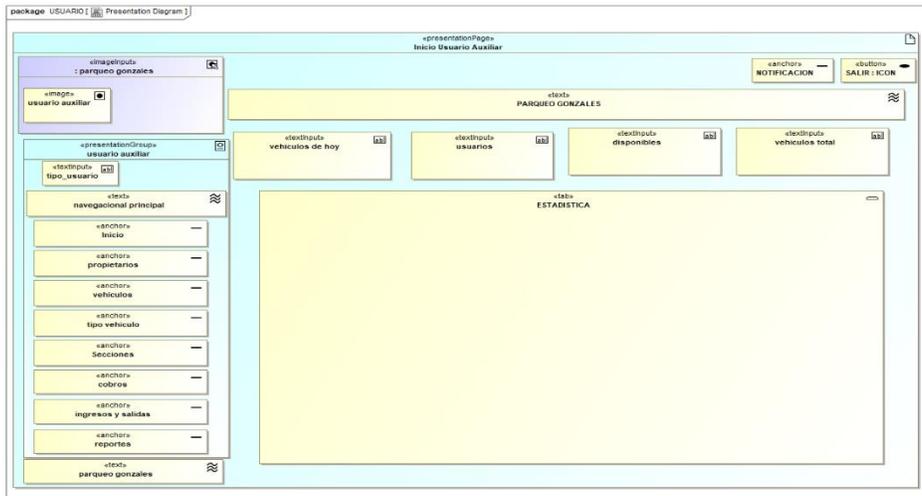
**Figura 17. Modelo de presentación página principal del Administrador**



Fuente: (Elaboración propia)

En la *figura 18*. Se puede ver el diseño de Interfaz Abstracta del Modelo de presentación página principal del Usuario Auxiliar con los permisos otorgados del administrador.

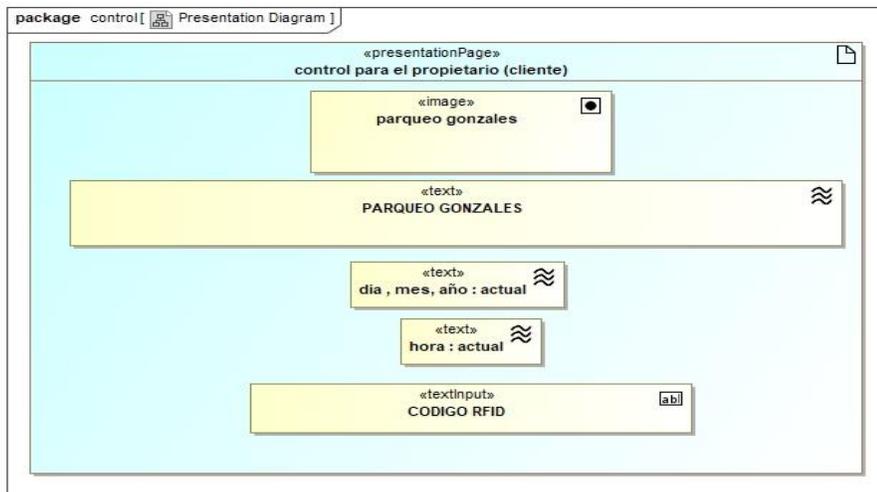
**Figura 18. Modelo de presentación página principal del Usuario Auxiliar**



**Fuente:** (Elaboración propia)

En la *figura 19*. Se puede ver el diseño de Interfaz Abstracta del Modelo de presentación en pantalla página principal del Propietario (Cliente).

**Figura 19. Modelo de presentación en pantalla página principal del Propietario**



**Fuente:** (Elaboración propia)

### 3.1.5. Implementación

En esta fase de implementación consiste en el funcionamiento del sistema, aplicando a partir del diseño del modelo de presentación OOADM.

En la *figura 20*. Se muestra en fondo de Área del Parqueo, en el centro está el formulario de inicio de sesión del sistema, como campos principales es el nombre del usuario y su contraseña respectiva.

**Figura 20. Inicio de Sesión**



```
@extends('layouts.login')
@section('css')
<link rel="stylesheet" href="{{asset('css/login.css')}}">
@endsection
@section('content')
<div class="login-box">
<div class="logo">
<a href="javascript:void(0);">{{$empresa->name}}</a>

</div>
<div class="card">
<div class="body">
<form id="sign_in" method="POST" action="{{ route('login') }}">
@csrf
<div class="msg">Escribe tu usuario y contraseña</div>
<div class="input-group">
<span class="input-group-addon">
<i class="material-icons">person</i>
</span>
<div class="form-line">
<input type="text" class="form-control" name="name" value="{{ old('name') }}" placeholder="Usuario" required autofocus>
</div>
</div>
<div class="input-group">
<span class="input-group-addon">
<i class="material-icons">lock</i>
</span>
<div class="form-line">
<input type="password" class="form-control" name="password" placeholder="Contraseña" required>
</div>
</div>
<div class="row">
<div class="col-xs-8 col-xs-offset-2">
<button class="btn btn-block bg-blue waves-effect" type="submit">ACCEDER</button>
</div>
</div>
</form>
</div>
</div>
</div>
@endsection
```

Fuente: (Elaboración propia)



En la *figura 22* En el módulo usuario se, editar, eliminar, también se podrá ver la lista de los usuarios registrados y realizar búsquedas y realizar nuevos registros *figura 23*. Además, se puede realizar asignaciones de roles a los módulos del sistema en la *figura. 24*.

**Figura 22. Módulo de usuario**

The screenshot shows a web interface for user management. At the top left, there is a blue button labeled 'Nuevo usuario'. Below it, the heading 'LISTA DE USUARIOS' is displayed. A search bar on the right contains the text 'Buscar:'. Below the search bar, there is a table with columns: 'Código', 'Apellidos y nombre(s)', 'C.I.', 'Teléfono/Celular', 'Tipo Usuario', 'Foto', and 'Acción'. The first row of data shows 'JVALDEZ' as the code, 'JOEL VALDEZ' as the name, '12345678 LP' as the C.I., 'S/N - 78945635' as the phone number, 'AUXILIAR' as the user type, and a circular profile picture of a man with a beard. The 'Acción' column contains icons for edit and delete. Below the table, it says 'Mostrando 1 a 1 de 1 registros' and navigation buttons for 'Anterior' and 'Siguiente' with a '1' in a blue box.

Código	Apellidos y nombre(s)	C.I.	Teléfono/Celular	Tipo Usuario	Foto	Acción
JVALDEZ	JOEL VALDEZ	12345678 LP	S/N - 78945635	AUXILIAR		

**Fuente:** (Elaboración propia)

**Figura 23. Registro de Usuario**

The screenshot shows a registration form titled 'REGISTRAR USUARIO' and 'NUEVO USUARIO'. A note states 'Los campos con \* son obligatorios.' The form is divided into sections: 'Información:' with fields for 'Nombre(s)\*', 'Apellido paterno\*', and 'Apellido materno'; 'Cédula de identidad\*' and 'Expedido\*' with a dropdown arrow; a profile picture placeholder with a red button labeled 'Elegir foto'; and 'Dirección:' with a 'Dirección\*' field.

**Fuente:** (Elaboración propia)

**Figura 24. Asignar Roles a los Usuario Auxiliares**

**LISTA DE PERMISOS**

<p><b>USUARIOS</b></p> <p>VER LA LISTA DE USUARIOS</p> <p>CREAR USUARIOS</p> <p>EDITAR USUARIOS</p> <p>ELIMINAR USUARIOS</p>	<p><b>TARIFAS</b></p> <p>VER LA LISTA DE TARIFAS</p> <p>CREAR TARIFAS</p> <p>EDITAR TARIFAS</p> <p>ELIMINAR TARIFAS</p>	<p><b>PROPIETARIOS</b></p> <p>VER LA LISTA DE PROPIETARIOS</p> <p>CREAR PROPIETARIOS</p> <p>EDITAR PROPIETARIOS</p> <p>ELIMINAR PROPIETARIOS</p>	<p><b>VEHICULOS</b></p> <p>VER LA LISTA DE VEHICULOS</p> <p>CREAR VEHICULOS</p> <p>EDITAR VEHICULOS</p> <p>ELIMINAR VEHICULOS</p>
<p><b>TIPO DE VEHICULOS</b></p> <p>VER LA LISTA DE TIPO DE VEHICULOS</p> <p>CREAR TIPO DE VEHICULOS</p> <p>EDITAR TIPO DE VEHICULOS</p> <p>ELIMINAR TIPO DE VEHICULOS</p>	<p><b>SECCIONES</b></p> <p>VER LA LISTA DE SECCIONES</p> <p>CREAR SECCIONES - UBICACIONES</p> <p>EDITAR SECCIONES - UBICACIONES</p> <p>ELIMINAR SECCIONES - UBICACIONES</p>	<p><b>COBROS</b></p> <p>VER LA LISTA DE COBROS</p> <p>REGISTRAR COBROS</p> <p>ELIMINAR COBROS</p>	<p><b>INGRESOS Y SALIDAS</b></p> <p>VER LA LISTA DE INGRESOS Y SALIDAS</p> <p>REGISTRAR INGRESOS Y SALIDAS</p>
<p><b>REPORTES</b></p> <p>VER Y GENERAR REPORTES</p>	<p><b>EMPRESA</b></p> <p>VER INFORMACIÓN DE LA EMPRESA</p> <p>EDITAR INFORMACIÓN DE LA EMPRESA</p>	<p><b>USUARIOS PERMISOS</b></p> <p>VER LA LISTA DE PERMISOS DE UN USUARIO</p>	

**Fuente:** (Elaboración propia)

En la *figura 25*. En el módulo Tarifa, solo el administrador podrá ver este formulario, donde podrá registrar una nueva tarifa (*figura 26.*), eliminar, editar y se mostrará un listado de los registros.

**Figura 25. Módulo de tarifa**

[+ Nueva tarifa](#)

**TARIFAS**

Mostrar 10 registros Buscar:

Nombre	Horas	Precio	Descripción	Acción
TARIFA 1	1	5.00		
TARIFA 2	2	10.00		
PAGO MENSUAL	730	500.00		
Nombre	Horas	Precio	Descripción	Acción

Mostrando 1 a 3 de 3 registros Anterior **1** Siguiente

**Fuente:** (Elaboración propia)

**Figura 26. Registro de tarifa**

CREAR TARIFA

NUEVA TARIFA  
Los campos con \* son obligatorios.

**Información:**

Nombre\*

Horas\* Precio (Bs.)\*

Descripción

 GUARDAR

**Fuente:** (Elaboración propia)

En la *figura 27*. En el módulo Propietario, donde podrá registrar un nuevo propietario (*figura 28.*), eliminar, editar y se mostrará un listado de los registros.

**Figura 27. Módulo de Propietario (Cliente)**

 Nuevo propietario

PROPIETARIOS

Mostrar 10 registros Buscar: \_\_\_\_\_

Nombre	Foto	C.I.	Dirección	Teléfono/Celular	Correo	Acción
CARLOS CHOQUE		12345678 LP		S/N - S/N		 
FEDERICO PACO		5244858 LP		S/N - S/N		 
GERARDO MAMANI QUISPE		2625154 CB	RIO SECO	S/N - 6986158		 
GONZALO SILVESTRE		12345678 LP		S/N - S/N		 

**Fuente:** (Elaboración propia)

**Figura 28. Registro de Propietario (Cliente)**

NUEVO PROPIETARIO  
Los campos con \* son obligatorios.

**Información:**

Nombre(s)\*  Apellido paterno\*  Apellido materno

Cédula de identidad\*  Expedido\*

  
Elegir foto

**Dirección:**

Dirección

**Contacto:**

**Fuente:** (Elaboración propia)

En la *figura 29*. En el módulo tipo de vehículo, se podrá registrar (*figura 30*.), eliminar, editar y se mostrará un listado de los registros.

**Figura 29. Módulo de tipo de vehículo**

+  
Nuevo tipo

TIPOS DE VEHICULOS

Mostrar 10 registros  Buscar:

Nombre	Descripción	Acción
AUTOMOVIL	NISSAN 2020	 
AUTOMÓVIL	AUTOMÓVIL	 
SUZUQI	AUTOMOVIL	 
VAGONETA	VAGONETA	 
Nombre	Descripción	Acción

Mostrando 1 a 4 de 4 registros Anterior **1** Siguiente

**Fuente:** (Elaboración propia)

**Figura 30. Registro de tipo de vehículo**

NUEVO TIPO DE VEHICULO  
Los campos con \* son obligatorios.

**Información:**

Nombre\* Descripción

 GUARDAR

**Fuente:** (Elaboración propia)

En la *figura 31*. En el módulo Propietario, donde podrá registrar un nuevo propietario, eliminar, editar y se mostrará un listado de los registros.

**Figura 31. Módulo de Vehículo**

+  
Nuevo vehículo

LISTA DE VEHICULOS

Mostrar 10 registros Buscar: \_\_\_\_\_

Nombre	Foto	Placa	Marca	Modelo	Propietario	Tarifa	RFID	Acción
DEZIRE		4528-GFD	NISSAN	2018	GONZALO SILVESTRE	TARIFA 1	1234567890	  
FORD XLS		5012-SAR	FORD	2020	HECTOR PERES	PAGO MENSUAL	123450102	  
MAZDA CX3		4256-DDD	MAZDA	2020	CARLOS CHOQUE	PAGO MENSUAL	70446661	  
NISSAN KICKS		4567-HYG	NISSAN	2020	ROLANDO VARGAS	TARIFA 2	0844928464	  

**Fuente:** (Elaboración propia)

En la *figura 32*. Se realizará un nuevo registro, donde se debe pasar por el lector la nueva tarjeta de **RFID** y se registrara automáticamente donde aparecerá el código y se registrara.

**Figura 32. Registro Vehículo**

**Información:**

Nombre del vehículo\*  Placa

Marca  Modelo

Tipo\*  Propietario\*

RFID\*  Tarifa\*

Seleccione esta caja de texto y pase la tarjeta RFID.



[Elegir foto](#)

[GUARDAR](#)

**Fuente:** (Elaboración propia)

En la *figura 33*. En el módulo de Secciones, donde solo el administrador podrá registrar un nuevo propietario, eliminar, editar y se mostrará un listado de los registros y se puede ver mapeos.

**Figura 33. Módulo de Secciones**

[+ Nueva sección](#)

LISTA DE SECCIONES - UBICACIONES

Mostrar 10 registros

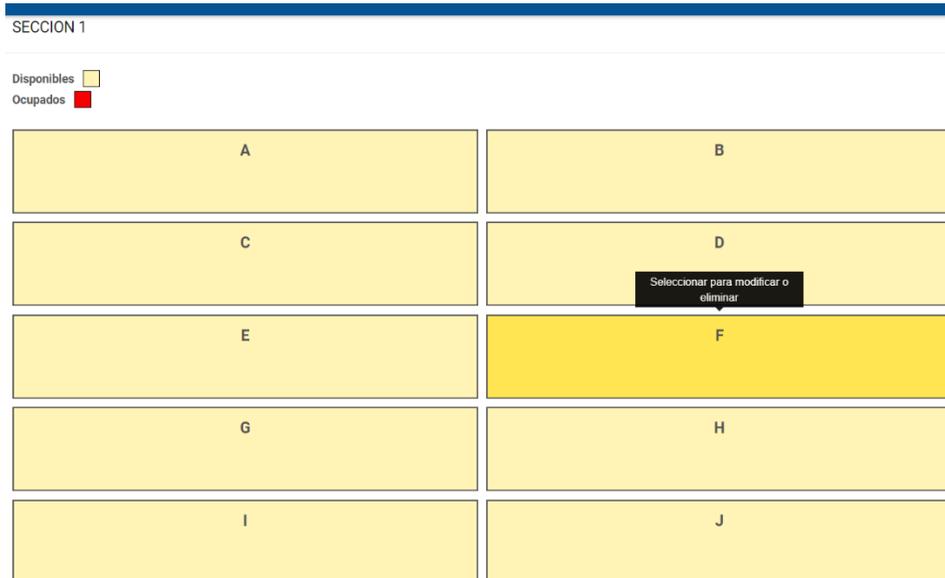
Nombre	Capacidad	Descripción	Mapeo	Acción
SECCION 1	29		<a href="#">Ver mapeo</a>	 
SECCION 2	30		<a href="#">Ver mapeo</a>	 
Nombre	Capacidad	Descripción	Mapeo	Acción

Mostrando 1 a 2 de 2 registros [Anterior](#) [1](#) [Siguiente](#)

**Fuente:** (Elaboración propia)

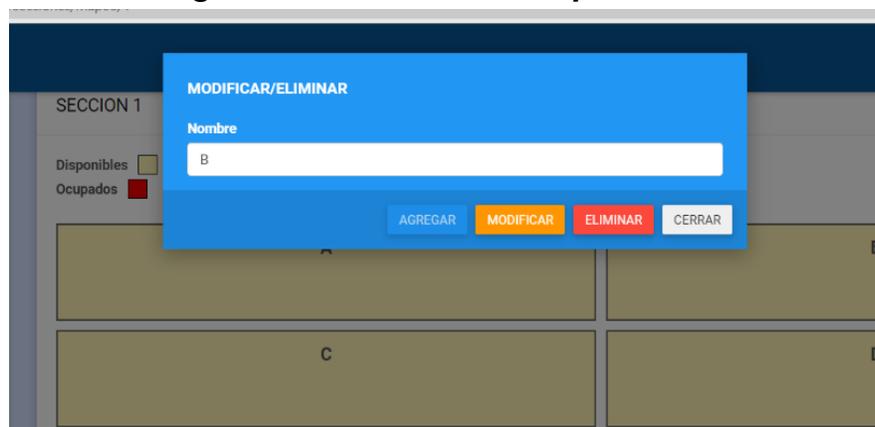
En la *figura 34*. Se realizará un nuevo registro en el mapeo, donde se debe pasar por el lector la nueva tarjeta de **RFID** y se registrará automáticamente donde se le asigna un lugar, el mapeo y además se puede agregar, editar y eliminar en la *Figura 35.*, el nombre de los espacios.

**Figura 34. Seguimiento de Mapeo de los Vehículos**



**Fuente:** (Elaboración propia)

**Figura 35. Nombre de los espacios**



**Fuente:** (Elaboración propia)

En la *figura 36*. En el módulo de cobros, se genera cuando marca la salida con la tarjeta RFID

También se puede el listado y realizar búsquedas y se visualiza la factura (*Figura 37*).

**Figura 36. Módulo de Cobros**

LISTA DE COBROS

Mostrar 10 registros Buscar: \_\_\_\_\_

Fecha	Vehiculo	Propietario	Total Bs.	Acción
2021-04-27	NISSAN KICKS	ROLANDO VARGAS	18.00	 
2021-04-24	NISSAN KICKS	ROLANDO VARGAS	18.00	 
2021-04-20	NISSAN KICKS	ROLANDO VARGAS	18.00	 
2021-04-16	NISSAN KICKS	ROLANDO VARGAS	18.00	 
2021-04-15	MAZDA CX3	CARLOS CHOQUE	500.00	 
2021-04-15	TOYOTA YARIS	ROLANDO VARGAS	18.00	 
2021-04-01	FORD XLS	HECTOR PERES	500.00	 
Fecha	Vehiculo	Propietario	Total Bs.	Acción

Fuente: (Elaboración propia)

**Figura 37. Visualiza la factura**



Fuente: (Elaboración propia)

En la *figura 38*. En el módulo de ingresos y salidas, se genera cuando marca la entrada y salida con la tarjeta RFID y se registrara automáticamente donde se muestra datos (*Figura 39*). También se puede el listado y realizar búsquedas

**Figura 38. Módulo de Cobros**

Código	Fecha	Hora	Vehículo	Acción	Observación	Acción
1	2021-03-14	20:06:00	FORD XLS	INGRESO		
3	2021-03-14	20:09:00	FORD XLS	SALIDA		
4	2021-04-15	19:30:00	FORD XLS	INGRESO		
5	2021-04-15	19:35:00	MAZDA CX3	INGRESO		
7	2021-04-15	19:41:00	MAZDA CX3	SALIDA		
8	2021-04-15	19:43:00	MAZDA CX3	INGRESO		
9	2021-04-15	19:44:00	TOYOTA YARIS	INGRESO		
10	2021-04-15	19:45:00	MAZDA CX3	SALIDA		

**Fuente:** (Elaboración propia)

**Figura 39. Visualiza la entrada y salida**

INGRESO Y SALIDA DE VEHICULOS											
VER INFORMACIÓN											
	<table border="1"> <tr> <td>ACCIÓN</td> <td>SALIDA</td> </tr> <tr> <td>VEHICULO:</td> <td>NISSAN KICKS</td> </tr> <tr> <td>PROPIETARIO:</td> <td>ROLANDO VARGAS</td> </tr> <tr> <td>HORA:</td> <td>19:31:00</td> </tr> <tr> <td>FECHA:</td> <td>27/04/2021</td> </tr> </table>	ACCIÓN	SALIDA	VEHICULO:	NISSAN KICKS	PROPIETARIO:	ROLANDO VARGAS	HORA:	19:31:00	FECHA:	27/04/2021
ACCIÓN	SALIDA										
VEHICULO:	NISSAN KICKS										
PROPIETARIO:	ROLANDO VARGAS										
HORA:	19:31:00										
FECHA:	27/04/2021										

VER INFORMACIÓN											
	<table border="1"> <tr> <td>ACCIÓN</td> <td>INGRESO</td> </tr> <tr> <td>VEHICULO:</td> <td>NISSAN KICKS</td> </tr> <tr> <td>PROPIETARIO:</td> <td>ROLANDO VARGAS</td> </tr> <tr> <td>HORA:</td> <td>19:12:00</td> </tr> <tr> <td>FECHA:</td> <td>27/04/2021</td> </tr> </table>	ACCIÓN	INGRESO	VEHICULO:	NISSAN KICKS	PROPIETARIO:	ROLANDO VARGAS	HORA:	19:12:00	FECHA:	27/04/2021
ACCIÓN	INGRESO										
VEHICULO:	NISSAN KICKS										
PROPIETARIO:	ROLANDO VARGAS										
HORA:	19:12:00										
FECHA:	27/04/2021										

**Fuente:** (Elaboración propia)

En la *figura 40*. En el módulo de reportes, genera reportes de los datos registrados por fecha, mes, día. (*Figura 41*). También se puede el listado y realizar búsquedas

**Figura 40. Módulo de Cobros**



**Fuente:** (Elaboración propia)

En la *figura 41*. Visualiza la entrada y salida de los vehículos.

**Figura 41. Visualiza la entrada y salida**



**Fuente:** (Elaboración propia)

En la *figura 42*. Visualiza el reporte de la lista de los usuarios auxiliares.

**Figura 42. Visualiza el reporte de usuarios registrados**



**PARQUE GONZALES**

LISTA DE USUARIOS

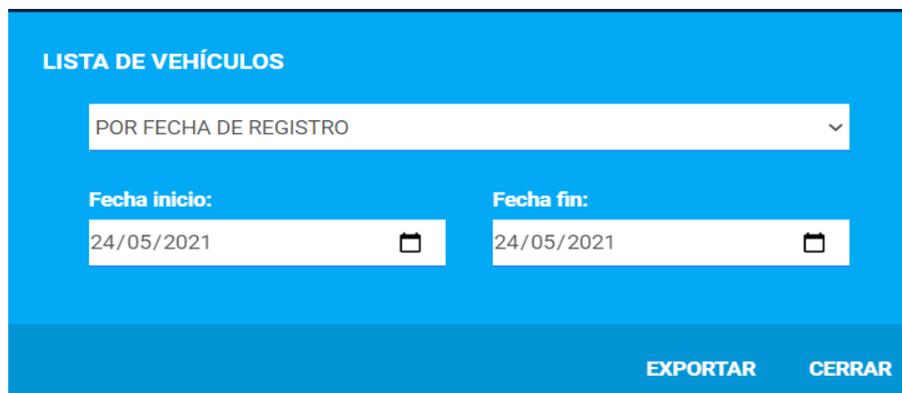
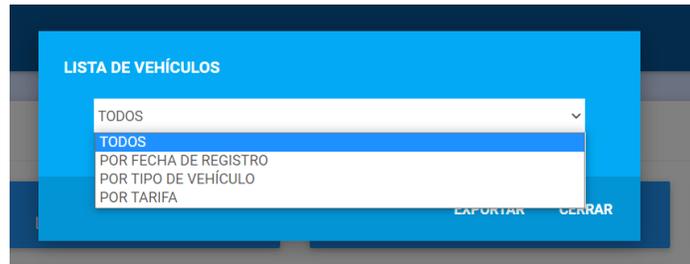
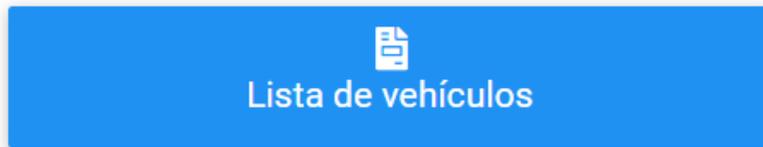
FECHA DE EMISIÓN: 25-05-2021

Nro.	Código de registro	Nombre	C.I.	Tel./Cel.	Tipo usuario
1	JVALDEZ	JOEL VALDEZ	12345678 LP	S/N / 78945635	AUXILIAR

**Fuente:** (Elaboración propia)

En la *figura 43*. Visualiza el reporte de la lista de los vehículos registrados en el sistema, los genera por todos, fecha, tipo y por la tarifa de registro.

**Figura 43. Visualiza el reporte de lista de los vehículos**



**PARQUEO GONZALES**

LISTA DE VEHICULOS

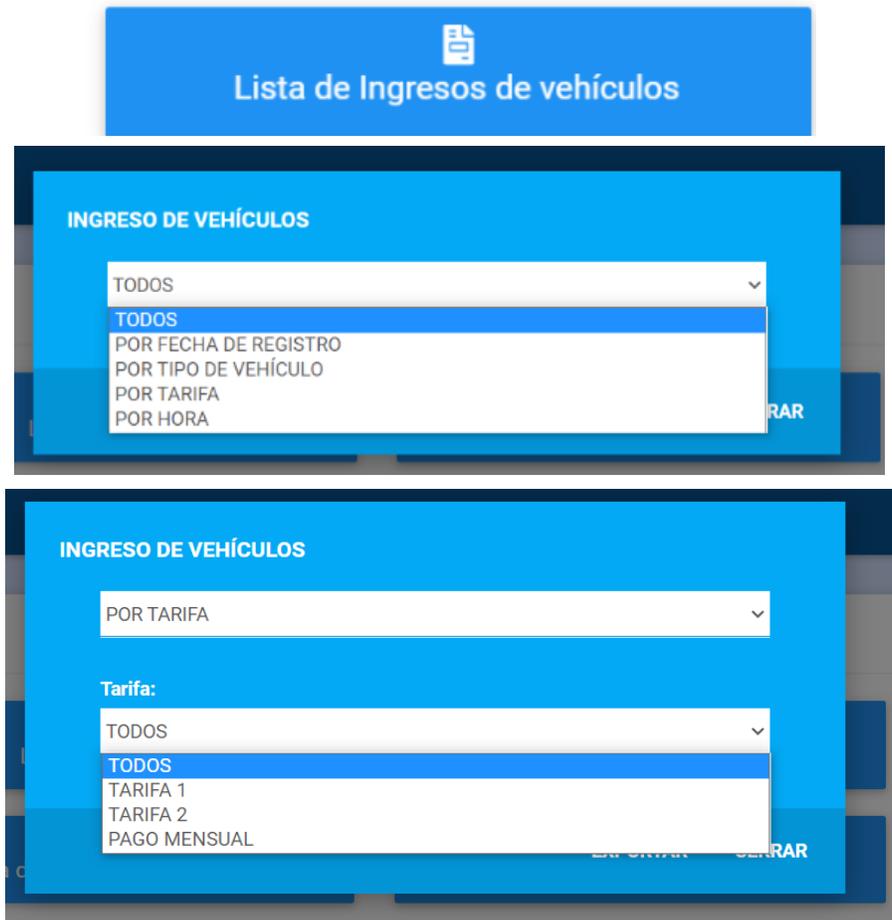
FECHA DE EMISIÓN: 25-05-2021

Nro.	Propietario	Nombre	Marca	Modelo	Tipo	Tarifa	Foto	Rfid	Fecha de registro
1	GONZALO SILVESTRE	DEZIRE	NISSAN	2018	AUTOMÓVIL	TARIFA 1		1234567890	2021-03-17
2	HECTOR PERES	FORD XLS	FORD	2020	VAGONETA	PAGO MENSUAL		123450102	2021-03-17
3	ROLANDO VARGAS	TOYOTA YARIS	TOYOTA	2017	AUTOMÓVIL	TARIFA 2		123456	2021-03-26

**Fuente:** (Elaboración propia)

En la *figura 44*. Visualiza el reporte de la lista de los vehículos registrados en el sistema, genera la lista de los que ingresos al parqueo, los genera por todos, fecha, tipo y por la tarifa de registro.

**Figura 44. Visualiza el reporte de lista de ingresos de los vehículos**



**PARQUEO GONZALES**  
LISTA DE INGRESOS DE VEHÍCULOS  
FECHA DE EMISIÓN: 25-05-2021

Nro.	Vehículo	Propietario	Fecha	Hora
1	FORD XLS	HECTOR PERES	2021-03-14	20:06:00
2	FORD XLS	HECTOR PERES	2021-04-15	19:30:00
3	MAZDA CX3	CARLOS CHOQUE	2021-04-15	19:35:00
4	MAZDA CX3	CARLOS CHOQUE	2021-04-15	19:43:00
5	MAZDA CX3	CARLOS CHOQUE	2021-04-15	19:51:00

**Fuente:** (Elaboración propia)

En la *figura 45*. Visualiza el reporte de la lista de los vehículos registrados en el sistema, genera la lista de las salidas del parqueo, los genera por todos, fecha, tipo y por la tarifa de registro.

**Figura 45. Visualiza el reporte de lista de salidas de los vehículos**

The image displays two screenshots of a web application interface for generating vehicle exit reports. The top screenshot shows a dropdown menu with the following options: TODOS, POR FECHA DE REGISTRO, POR TIPO DE VEHÍCULO, POR TARIFA, and POR HORA. The bottom screenshot shows a dropdown menu with the following options: TODOS, AUTOMÓVIL, VAGONETA, AUTOMOVIL, and SUZUQI.



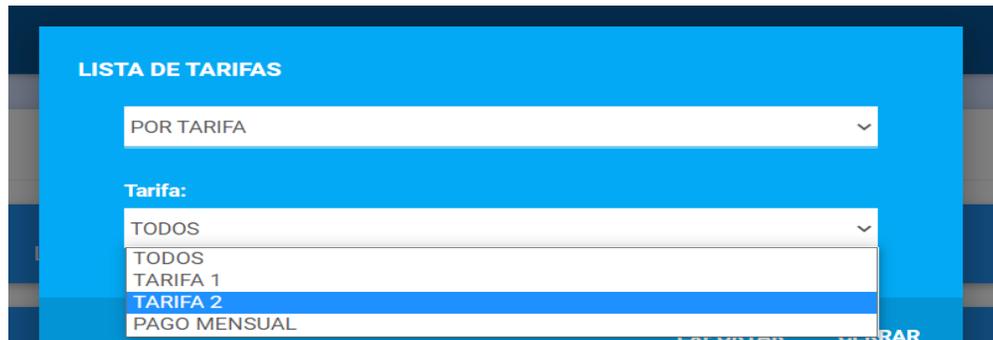
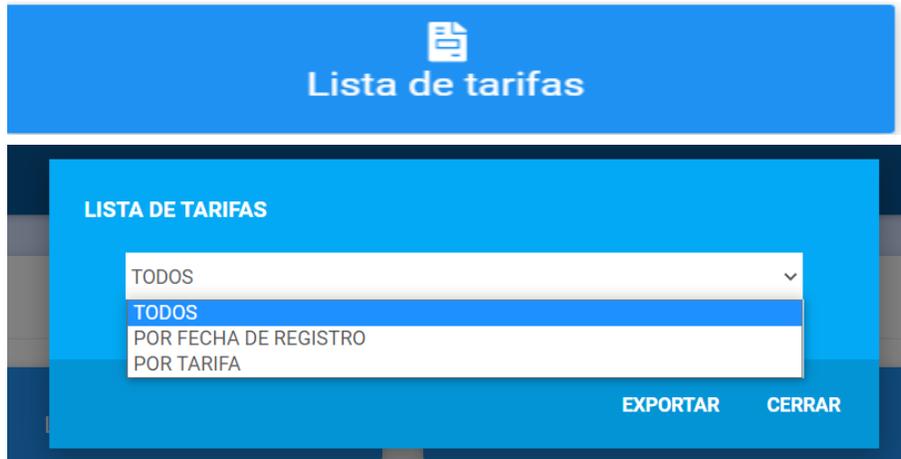
**PARQUEO GONZALES**  
 LISTA DE SALIDAS DE VEHÍCULOS  
 FECHA DE EMISIÓN: 25-05-2021

Nro.	Vehículo	Propietario	Fecha	Hora
1	FORD XLS	HECTOR PERES	2021-03-14	20:09:00
2	MAZDA CX3	CARLOS CHOQUE	2021-04-15	19:41:00
3	MAZDA CX3	CARLOS CHOQUE	2021-04-15	19:45:00
4	TOYOTA YARIS	ROLANDO VARGAS	2021-04-15	19:46:00
5	NISSAN KICKS	ROLANDO VARGAS	2021-04-16	14:50:00
6	NISSAN KICKS	ROLANDO VARGAS	2021-04-20	19:56:00
7	NISSAN KICKS	ROLANDO VARGAS	2021-04-24	17:26:00
8	NISSAN KICKS	ROLANDO VARGAS	2021-04-27	19:31:00

**Fuente:** (Elaboración propia)

En la *figura 46*. Visualiza el reporte de la lista de las tarifas registradas registrados en el sistema, los genera por todos, fecha, y por la tarifa de registro.

**Figura 46. Visualiza el reporte de lista de tarifas**



**PARQUEO GONZALES**

LISTA DE TARIFAS

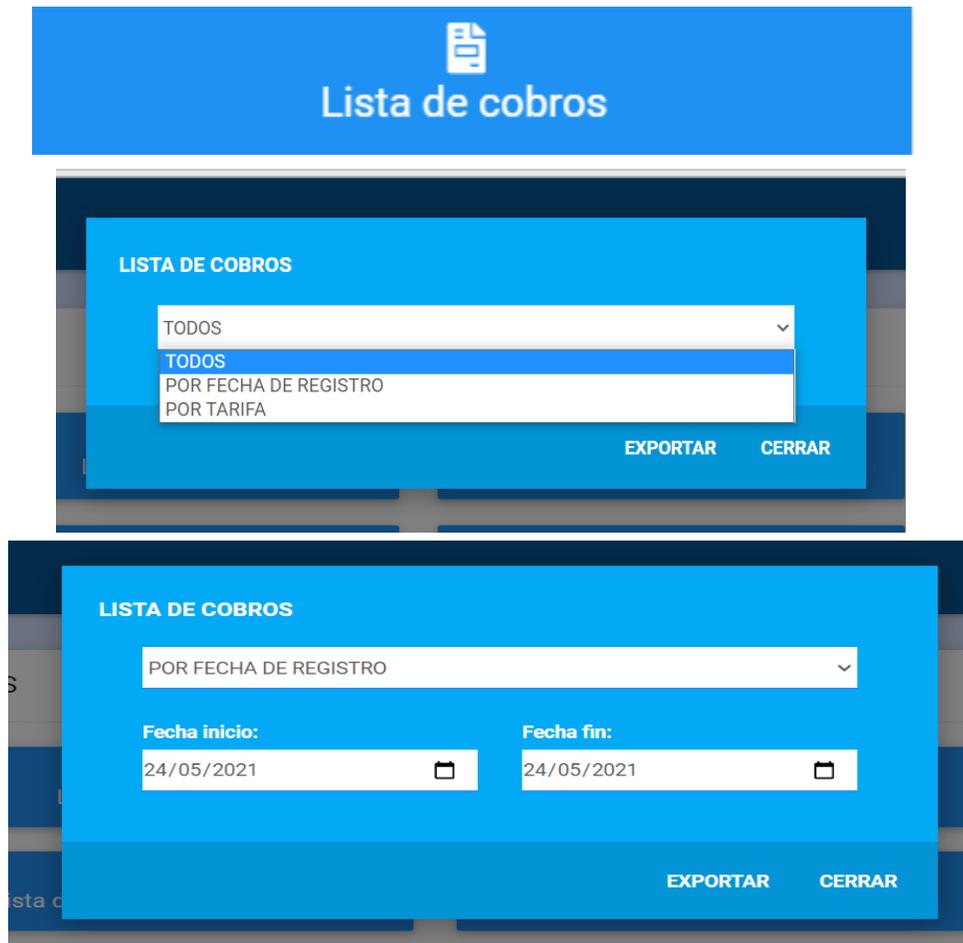
FECHA DE EMISIÓN: 25-05-2021

Nro.	Nombre	Horas	Precio	Fecha de registro	Descripción
1	TARIFA 1	1	5.00	2021-03-20	
2	TARIFA 2	2	10.00	2021-03-20	
3	PAGO MENSUAL	730	500.00	2021-03-13	

**Fuente:** (Elaboración propia)

En la *figura 47*. Visualiza el reporte de la lista de los cobros de los vehículos que usaron el servicio en el parqueo, los genera por todos, fecha, y por la tarifa de registro.

**Figura 47. Visualiza el reporte de lista de cobros**



**PARQUEO GONZALES**

LISTA DE COBROS

FECHA DE EMISIÓN: 25-05-2021

Nro.	Vehículo	Propietario	Tarifa	Tiempo cobrado (Hrs.)	Fecha y hora	Total
1	FORD XLS	HECTOR PERES	PAGO MENSUAL	672	2021-04-01 20:07:00	500.00
2	MAZDA CX3	CARLOS CHOQUE	PAGO MENSUAL	672	2021-04-15 19:35:00	500.00
3	TOYOTA YARIS	ROLANDO VARGAS	TARIFA 2	4	2021-04-15 19:46:00	18.00
4	NISSAN KICKS	ROLANDO VARGAS	TARIFA 2	4	2021-04-16 14:50:00	18.00
5	NISSAN KICKS	ROLANDO VARGAS	TARIFA 2	4	2021-04-20 19:56:00	18.00
6	NISSAN KICKS	ROLANDO VARGAS	TARIFA 2	4	2021-04-24 17:26:00	18.00
7	NISSAN KICKS	ROLANDO VARGAS	TARIFA 2	4	2021-04-27 19:31:00	18.00
<b>TOTAL</b>						1,090.00

**Fuente:** (Elaboración propia)

En la *figura 48*. Visualiza el reporte de la lista de las ubicaciones disponibles y ocupados por vehículos registrados en el sistema, los genera por sección.

**Figura 48. Visualiza el reporte de lista de ubicaciones disponibles y ocupados**



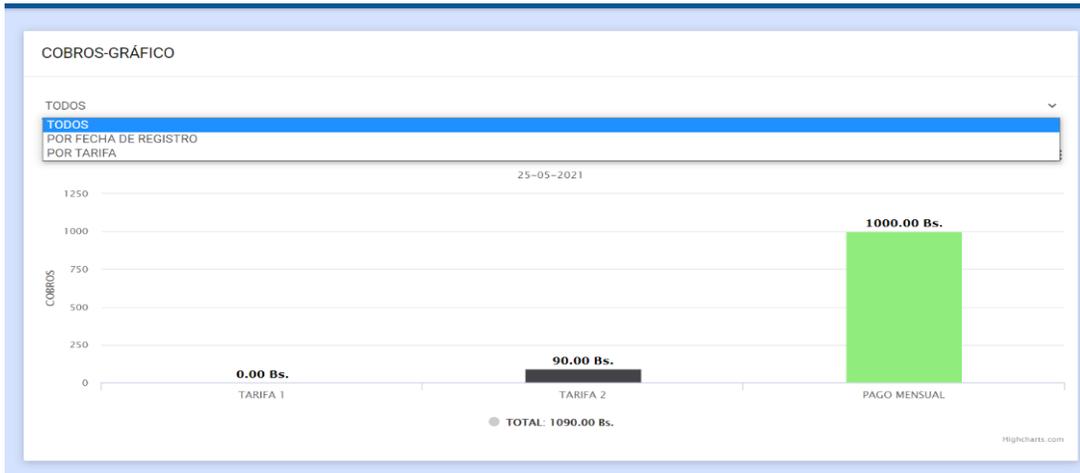
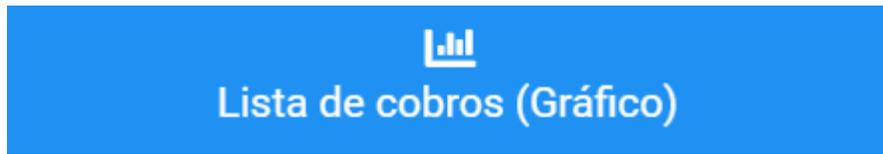
**PARQUEO GONZALES**  
 LISTA DE UBICACIONES DISPONIBLES  
 FECHA DE EMISIÓN: 25-05-2021

SECCION 1	
DISPONIBLES (28)	OCUPADOS (1)
<ul style="list-style-type: none"> <li>• B</li> <li>• C</li> <li>• D</li> <li>• E</li> <li>• F</li> <li>• G</li> <li>• H</li> <li>• I</li> <li>• J</li> <li>• K</li> <li>• L</li> <li>• M</li> <li>• N</li> <li>• O</li> <li>• P</li> <li>• Q</li> <li>• R</li> <li>• S</li> <li>• T</li> <li>• U</li> <li>• V</li> <li>• W</li> <li>• X</li> <li>• Y</li> <li>• Z</li> <li>• B-2</li> <li>• C-3</li> <li>• D-4</li> </ul>	<ul style="list-style-type: none"> <li>• A - MAZDA CX3</li> </ul>

**Fuente:** (Elaboración propia)

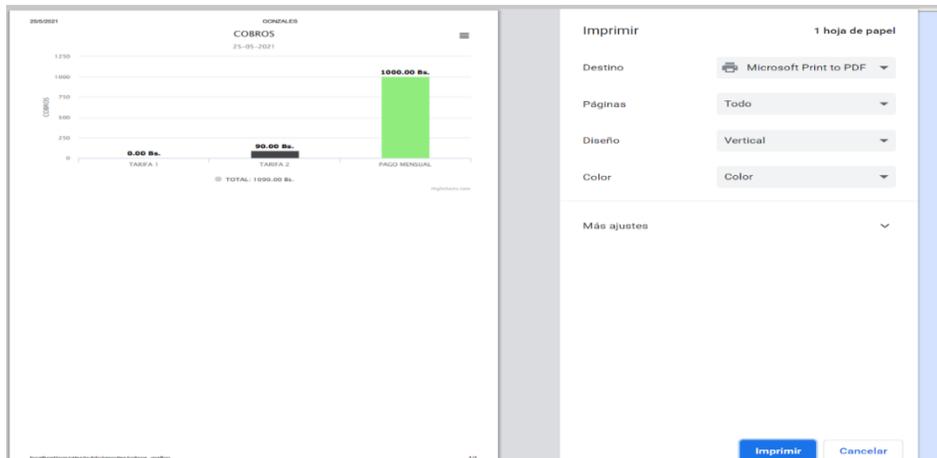
En la *figura 49*. Es donde muestra los gráficos estadísticos de los cobros y el total, lo cual genera reportes de los datos registrados por fecha, mes, día.

**Figura 49. Visualiza las estadísticas de las ganancias**



Fuente: (Elaboración propia)

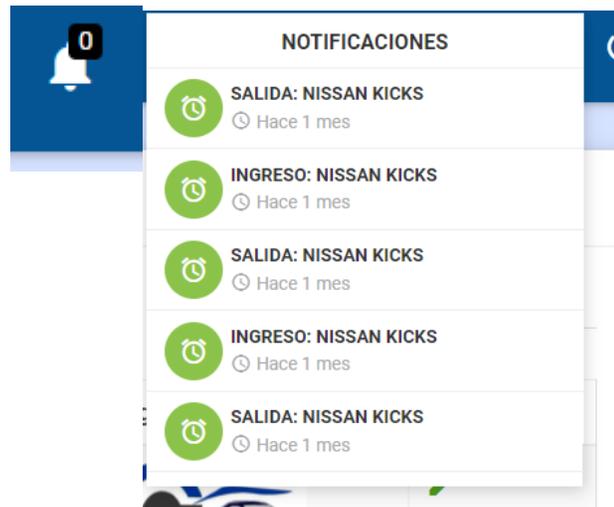
**Figura 50. Puede imprimir la estadística de las ganancias**



Fuente: (Elaboración propia)

En la *figura 51*. Es donde muestra las notificaciones de ingresos y salidas de los vehículos.

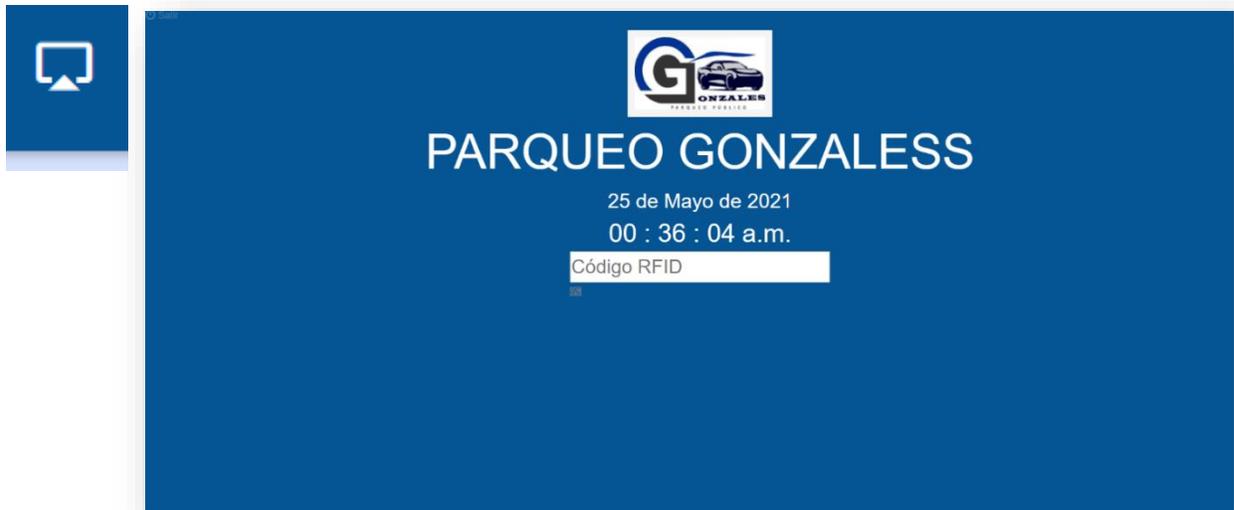
**Figura 51. Visualiza las notificaciones de ingreso y salida**



**Fuente:** (Elaboración propia)

En la *figura 52*. Es donde muestra la asignación del espacio del parqueo, al cliente por 10 segundos en pantalla una vez que marque con su tarjeta RFID.

**Figura 52. Visualiza las el control con la tarjeta en pantalla**



**Fuente:** (Elaboración propia)

En la *figura 53*. Se muestra los botones o controles para la salida del sistema.

**Figura 53. Son los controles para salir del sistema**



**Fuente:** (Elaboración propia)

En la *figura 54*. En el módulo es el registro de los datos de la empresa y editar datos de la empresa *figura 55*.

**Figura 54. Módulo de datos de la empresa**

EMPRESA

Mostrar 10 registros Buscar: \_\_\_\_\_

Código	Nombre	NIT	Nro. autorización	Teléfono-Celular	Logo	Acción
0	PARQUEO GONZALESS	0	0	2885625 74558586		
Código	Nombre	NIT	Nro. autorización	Teléfono-Celular	Logo	Acción

Mostrando 1 a 1 de 1 registros Anterior **1** Siguiente

**Fuente:** (Elaboración propia)

**Figura 55. Editar datos de la empresa**

Los campos con \* son obligatorios.

**Información:**

Código\* 0 NIT\* 0 Nro. de autorización 0

Nro. de empleador\* Nombre de la empresa\* PARQUEO GONZALESS Alias\* GZ

**Contacto:**

Departamento\* LA PAZ Ciudad\*

Calle/Avenida\* CALLE 1 Nro.\* 0

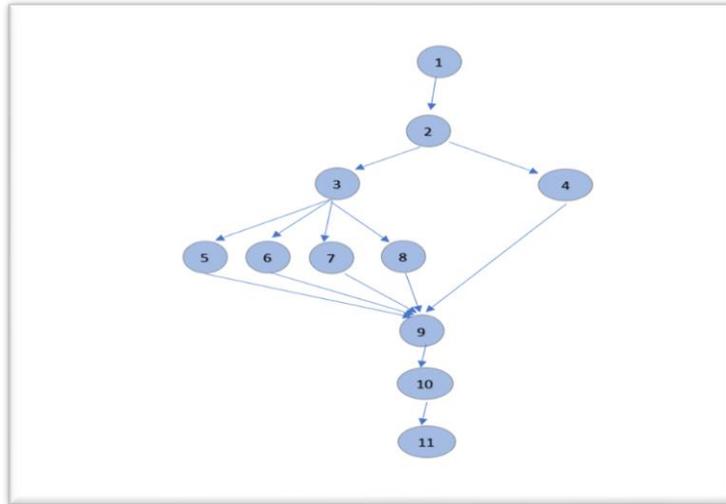
Teléfono 2885625 Celular 74558586

Web

**Fuente:** (Elaboración propia)



**Figura 57. Grafo de flujo del acceso al módulo de propietarios**



**Fuente:** (Elaboración propia)

Donde:

- Inicio del sistema (1)
- Menú principal (2)
- Módulo de propietarios (3)
- No muestra ningún modulo (4)
- Registrar propietarios (5)
- Listar propietarios (6)
- Editar propietarios (7)
- Eliminar propietarios (8)
- Fin de ciclo usuario (9)
- Fin ciclo sistema (10)
- Fin del sistema (11)

Analizando el grafo generado a partir de las características del sistema, ahora se procede a determinar la complejidad ciclomática del grafo mediante:

Cálculo de la Complejidad Ciclométrica

$$V(G) = A - N + 2$$

Donde:

**A:** es el número de aristas

**N:** es el número de nodos

En nuestro caso tenemos que A= 14 (Aristas) y N= 11(Nodos). Entonces, se tiene que:

$$V(G) = 14 - 11 + 2 \Rightarrow V(G) = 5$$

Por tanto, la complejidad Ciclométrica es ocho.

### **Determinación de los casos de prueba**

Como  $V(G) = 8$ , existe cuatro caminos básicos. Estos son:

**Camino 1:** 1-2-3-5-9-10-11

**Camino 2:** 1-2-3-6-9-10-11

**Camino 3:** 1-2-3-7-9-10-11

**Camino 4:** 1-2-3-8-9-10-11

**Camino 5:** 1-2-4-9-10-11

Preparar los casos de prueba para forzar la ejecución de cada camino. Esta última condición establece que, para la ejecución de ciertos caminos, se deben establecer las condiciones en las que al menos se ejecuta los nodos establecidos en el camino.

Entonces, los casos de prueba son los siguientes:

**Tabla 15. Prueba del módulo de propietarios**

N° CAMINOS	CASOS DE PRUEBA
<b>Camino 1</b>	Una vez que el que ingreso del administrador o usuario y tenga permiso al entrar al sistema y al módulo puede registrar datos.
<b>Camino 2</b>	Una vez que el que ingreso del administrador o usuario y tenga permiso al entrar al sistema y al módulo puede listar y realizar búsquedas de datos.
<b>Camino 3</b>	Una vez que el que ingreso del administrador o usuario y tenga permiso al entrar al sistema y al módulo puede editar datos.
<b>Camino 4</b>	Una vez que el que ingreso del administrador o usuario y tenga permiso al entrar al sistema y al módulo puede eliminar datos.
<b>Camino 5</b>	Si el administrador no le otorga el rol al usuario directamente le muestra la pantalla principal sin acceso, concluye con la administración del sistema y cierra sesión.

**Fuente:** (Elaboración propia)

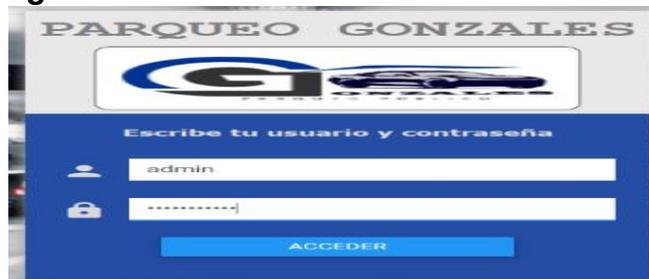
### 3.2.2. Pruebas de caja negra

Para realizar la prueba de caja negra se toman los casos de uso registrar, modificar, buscar y eliminar archivos de expedientes lo cual se demostrará en la siguiente tabla.

#### 3.2.2.1. Prueba de Inicio de sesión

Se realiza las pruebas a la interfaz de inicio de sección mostrada en la siguiente figura:

**Figura 58. Prueba de Inicio de sesión**



**Fuente:** (Elaboración propia)

**Tabla 16. Prueba de inicio de sesión**

Campo	Entrada Valida	Entrada Invalida
Usuario	Cadena de texto	Caracteres especiales, espacios en blanco
Contraseña	Cadena de texto	Caracteres especiales, espacios en blanco

Fuente: (Elaboración propia)

**Figura 59. Prueba de Acceso al Sistema**



Fuente: (Elaboración propia)

**Tabla 17. Prueba de Acceso al Sistema**

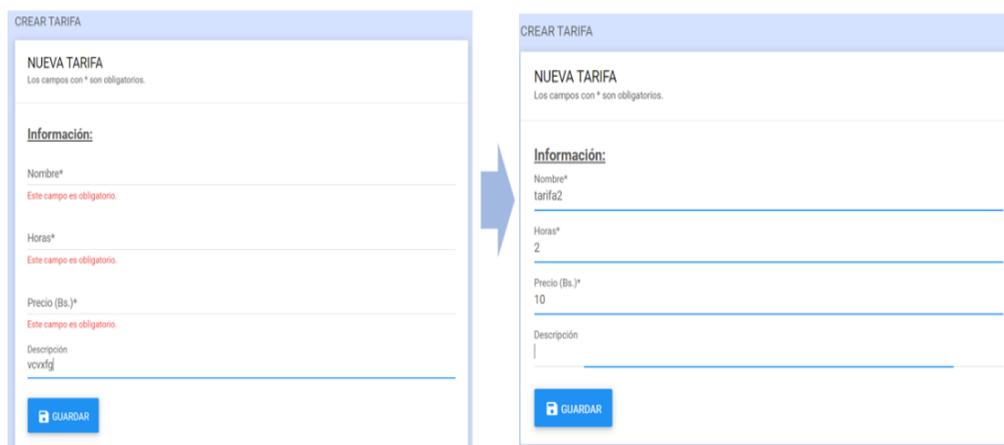
Campo		Entrada Valida	Entrada Invalida
Usuario	Contraseña		
		Ingrese el usuario y contraseña	El sistema valida que no se ingresen datos en blanco.
adminTR ador	Admin12 3	El usuario o la contraseña son incorrectas	El sistema mostrara un mensaje de error de datos
Administrador	Admin123_	Bienvenido al sistema parqueo Gonzales...!!	Al ingresar los datos correctos nos conduce al acceso al menú principal donde puede entrar al módulo de usuarios para dar roles que corresponda.

Fuente: (Elaboración propia)

Como se observó la interfaz de inicio de sesión cumple con la función programada para que el usuario se identifique al empezar el sistema.

### 3.2.2.2. Prueba de Registro de Datos de los Módulos

**Figura 60. Prueba Del Módulo De Tarifa**



Fuente: (Elaboración propia)

**Tabla 18. Prueba Del Módulo De Tarifa**

Campo	Entrada Valida	Entrada Invalida
Nombre	Cadena de texto	Caracteres especiales
Horas	Cadena Numérica	Cadena de texto
Precio	Cadena Numérica	Cadena de texto
Descripción	Cadena de texto	Caracteres especiales
<b>Salida</b>	Este campo es obligatorio	Dato registrado, espacio en blanco
<b>Resultado</b>	El sistema valida que no se deje en blanco los campos de entrada requerida.	Cuando el usuario introduce datos validos en el sistema: <ul style="list-style-type: none"> <li>- Registra la información en la base de datos.</li> </ul>

Fuente: (Elaboración propia)

**Figura 61. Prueba Del Módulo De Propietario**

**Fuente:** (Elaboración propia)

**Tabla 19. Prueba Del Módulo De Propietario**

Campo	Entrada Valida	Entrada Invalida
Nombre	Cadena de texto	Caracteres especiales
Apellido paterno	Cadena de texto	Caracteres especiales
Apellido materno	Cadena de texto	Caracteres especiales
Cedula de identidad	Cadena Numérica	Cadena de texto
Expedido	Cadena de texto	Caracteres especiales
foto	Imagen	Caracteres especiales
Dirección	Cadena de texto	Caracteres especiales
Email	Cadena de texto	Caracteres especiales
Teléfono	Cadena Numérica	Cadena de texto
Celular	Cadena Numérica	Cadena de texto
<b>Salida</b>	Este campo es obligatorio	Dato registrado, espacio en blanco
<b>Resultado</b>	El sistema valida que no se deje en blanco los campos de entrada requerida.	Cuando el usuario introduce datos validos en el sistema: - Registra la información en la base de datos.

**Fuente:** (Elaboración propia)

**Figura 62. Prueba Del Módulo De Vehículo**

Los campos con \* son obligatorios.

**Información:**

Color del vehículo\*  
Este campo es obligatorio.

Placa\*  
Este campo es obligatorio.

Marca

Modelo

Tipo\*  
Este campo es obligatorio.

Propietario\*  
Este campo es obligatorio.

RFID\*  
Seleccione esta caja de texto y pase la tarjeta RFID.  
Este campo es obligatorio.

Tarifa\*  
Este campo es obligatorio.



Elegir foto

GUARDAR

Fuente: (Elaboración propia)

**Tabla 20. Prueba Del Módulo De Vehículo**

Campo	Entrada Valida	Entrada Invalida
Color de vehículo	Cadena de texto	Cadena Numérica
Placa	Cadena de texto	Caracteres especiales
Marca	Cadena de texto	Caracteres especiales
Modelo	Cadena de texto	Caracteres especiales
Tipo	Selección	Caracteres especiales
Propietario	Selección	Caracteres especiales
RFID	Cadena Numérica	Caracteres especiales
tarifa	Selección	Caracteres especiales
<b>Salida</b>	Este campo es obligatorio	Dato registrado, espacio en blanco
<b>Resultado</b>	El sistema valida que no se deje en blanco los campos de entrada requerida.	Cuando el usuario introduce datos validos en el sistema: <ul style="list-style-type: none"> <li>- Registra la información en la base de datos.</li> </ul>

Fuente: (Elaboración propia)

**Figura 63. Prueba Del Módulo De Tipo de Vehículo**

REGISTRAR TIPO

**NUEVO TIPO DE VEHICULO**  
Los campos con \* son obligatorios.

**Información:**

Nombre\*  Descripción

Este campo es obligatorio.

**GUARDAR**

**Fuente:** (Elaboración propia)

**Tabla 21. Prueba Del Módulo De Tipo de Vehículo**

Campo	Entrada Valida	Entrada Invalida
Nombre	Cadena de texto	Caracteres especiales
Descripción	Cadena de texto	Caracteres especiales
<b>Salida</b>	Este campo es obligatorio	Dato registrado, espacio en blanco
<b>Resultado</b>	El sistema valida que no se deje en blanco los campos de entrada requerida.	Cuando el usuario introduce datos validos en el sistema: - Registra la información en la base de datos.

**Fuente:** (Elaboración propia)

**Figura 64. Prueba Del Módulo De Secciones**



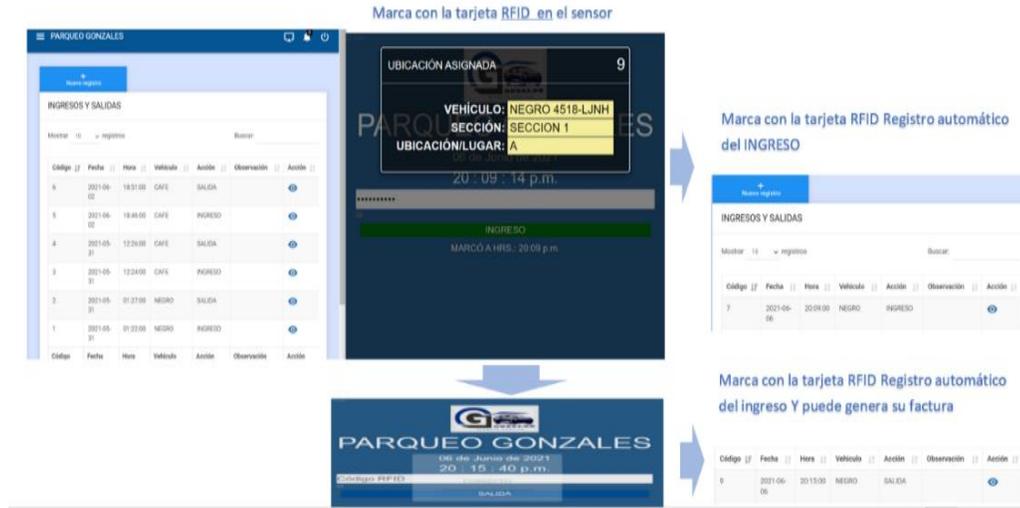
**Fuente:** (Elaboración propia)

**Tabla 22. Prueba Del Módulo De Secciones**

Campo	Entrada Valida	Entrada Invalida
Nombre	Cadena de texto	Caracteres especiales
Capacidad	Cadena Numérica	Cadena de texto
Descripción	Cadena de texto	Caracteres especiales
<b>Salida</b>	Este campo es obligatorio	Dato registrado, espacio en blanco
<b>Resultado</b>	El sistema valida que no se deje en blanco los campos de entrada requerida.	Cuando el usuario introduce datos validos en el sistema: - Registra la información en la base de datos.

**Fuente:** (Elaboración propia)

**Figura 65. Prueba De Ingresos y Salidas**



Fuente: (Elaboración propia)

**Tabla 23. Prueba Del Módulo De Ingresos y Salidas**

Campo	Entrada Valida	Entrada Invalida
código	Cadena Numérica	Cadena de texto
Fecha	Cadena Numérica	Caracteres especiales
hora	Cadena Numérica	Cadena de texto
vehiculó	Cadena de texto	Caracteres especiales
Acción	Cadena de texto	Caracteres especiales
Observación	Cadena de texto	Caracteres especiales
<b>Salida</b>	Este campo es obligatorio	RFID no registrado, espacio en blanco
<b>Resultado</b>	El sistema valida que no se deje en blanco los campos de entrada requerida.	Cuando el usuario debe ver si el cliente porta con una tarjeta RFID si la tiene marca en el lector y automáticamente se registra en la base de datos el ingreso y salida del vehículo.

Fuente: (Elaboración propia)

**Figura 66. Prueba Del módulo de cobros**



**Fuente:** (Elaboración propia)

**Tabla 24. Prueba Del Módulo De Ingresos y Reportes**

Campo	Entrada Valida	Entrada Invalida
código	Cadena Numérica	Cadena de texto
Fecha	Cadena Numérica	Caracteres especiales
vehiculó	Cadena de texto	Caracteres especiales
Total	Cadena Numérica	Cadena de texto
<b>Salida</b>	Este campo es obligatorio y se genera automático, Con la tarjeta RFID si marca su salida.	Si el cliente no marca la salida. No este registrado el cliente y su vehículo.
<b>Resultado</b>	Pase por su factura	Cuando el usuario debe ver si el cliente marca la salida con la tarjeta RFID, en pantalla y mostrara un mensaje.

**Fuente:** (Elaboración propia)

### **3.3. Métricas de Calidad**

#### **3.3.1. Factores de calidad ISO 25000**

Se hará la medición de calidad de software mediante la mediante del ISO/25010 que está basada en un modelo jerárquico de requerimientos de calidad que surge para sustituir a las antiguas ISO/9126 e ISO/IEC 14598, partiendo de las características de más alto nivel prescriptas en la norma ISO/9126 unificado el contenido de estas y definiendo a lo largo de sus distintos partes que establece cualquier componente de calidad de software puede ser descrito en termine de una de seis características básicas.

En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado. La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor.

Implementación de la ISO/IEC 25010 en la aplicación Web

En la implementación de la norma, se considera las características que mencionan esta y base a eso se fueron desarrollando los módulos de la aplicación Web tomando como referencia las métricas que dicta el auto Juan Francisco Jesús M.

##### **3.3.1.1. Adecuación funcional**

Es capaz de realizar tareas administrativas que el usuario específicamente necesita por ejemplo conocer las entradas y salidas de productos de estados financieros administración del personal proveedores y clientes entre otras funcionalidades mas.

Es la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas, este atributo valora las características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global.

Punto de función Para el cálculo de punto de función se toma en cuenta cinco características, el dominio de información, como son números de entradas, salida, condiciones, archivos e interfaz externa. Luego se realiza el cálculo de punto de función hallando la suma de estas características, parámetros de medición y el factor de ponderación también llamado punto medio de ponderación

- ✓ **Numero de entrada de usuarios:** Se cuenta cada entrada de usuario que proporciona datos al sistema.
- ✓ **Numero de salidas de usuario:** Se refiere cada salida que proporciona el sistema al usuario. Entre estos pueden ser informes, reportes y mensajes advertencia, notificaciones y errores.
- ✓ **Número de peticiones de usuario:** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta de software en forma de salidas interactivas.
- ✓ **Numero de archivos:** se cuentan archivo maestro lógico, esto pueden ser: grupo lógico de datos que sean parte de una base de datos, o un archivo independiente.
- ✓ **Numero de interfaces externas:** se cuenta las interfaces legibles por la máquina.
- ✓ Para calcular el punto de función se utilizará la siguiente formula:

$$PF = Cuenta\ total * (X + Min(Y) * \sum Fi)$$

Donde:

**PF:** Medida de la funcionalidad.

**Cuenta Total:** es el resultado del conteo de parámetros.

**X:** confiabilidad del proyecto, varía entre el 1% a 100%.

**Min(Y):** Error mínimo aceptable al de la complejidad, el margen de error es igual a 0.01.

$\sum Fi$ : son los valores de ajuste de la complejidad, donde  $i=1$  a  $i=14$

**Tabla 25. Parámetros de Medida y su Cantidad**

PARÁMETROS DE MEDIDA	CUENTA
Numero de entrada de usuario	20
Número de salidas de usuario	18
Número de peticiones de usuario	20
Numero de archivos	12
Numero de interfaces externas	3

**Fuente:** (Elaboración propia)

En la siguiente *Tabla 26*. Se calculará el punto de función se tiene que realizar el cálculo de la cuenta total con los factores de ponderación especificados.

**Tabla 26. Parámetros del punto de función**

PARAMETROS DE MEDIDA	CUENTA	FACTOR DE PONDERACION	TOTAL
Nº de entradas de usuario	20	*5	100
Nº de salidas de usuario	18	*5	90
Nº de peticiones de usuario	20	*4	90
Nº de archivos	12	*9	108
Nº de interfaces externas	3	*7	21
<b>CUENTA TOTAL</b>			<b>= 409</b>

**Fuente:** (Elaboración propia)

En la tabla anterior se muestra la cuenta total que se obtiene de la sumatoria de los factores de ponderación a los parámetros de medición.

**Tabla 27. Valores de ajustes de complejidad**

<b>IMPORTANCIA</b>	0%	20%	40%	60%	80%	100%	Fi
<b>Escala</b>	No Influencia	Incidencia	Moderado	Medio	Significativo	Esencial	
<b>Factor</b>	0	1	2	3	4	5	
1. ¿requiere el sistema copias de seguridad y de recuperación fiables?						X	5
2. ¿se requiere comunicación de datos?					X		4
3. ¿existen funciones de procesamiento distribuido?				X			3
4. ¿Es criterio el rendimiento?				X			3
5. ¿Se ejecuta el sistema en un entorno operativo existente y fuertemente utilizado?					X		4
6. ¿Requiere el sistema entrada de datos interactiva?					X		4
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?			X				2
8. ¿Se actualizan los archivos maestros de forma interactiva?				X			3
9. ¿Son complejos las entradas, las salidas, los archivos o peticiones?				X			3
10. ¿Es complejo el procesamiento interno?				X			3
11. ¿Se ha utilizado el código para ser reutilizable?					X		5
12. ¿Están incluidas en el diseño la conversión y la instalación?					X		4
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?					X		4
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?						X	5
<b>TOTAL</b>							<b>55</b>

**Fuente:** (Elaboración propia)

Una vez teniendo los valores correspondientes a las variables de la fórmula, se calculará el punto función mediante la siguiente ecuación:

$$PF = Cuenta\ total * (X + Min(Y) * \sum Fi)$$

$$PF = Cuenta\ total * (0,65 + 0.1 * \sum Fi)$$

Donde:

$$PF = 409 * (0,65 + 0,01 * 55)$$

$$PF = 409 * 1,2$$

$$PF = 490,8$$

Consideramos el máximo valor de complejidad  $\sum Fi = 70$  se calcula al 100% el nivel de confianza de la siguiente manera:

$$PF_{max} = Cuenta\ total * (0,65 + 0.01 * \sum Fi)$$

$$PF_{max} = 409 * (0,65 + 0.01 * 70)$$

$$PF_{max} = 409 * (0,65 + 0.01 * 70)$$

$$PF_{max} = 409 * 1,35$$

$$PF_{max} = 552,15$$

Después de haber calculado ambos valores se tiene que la funcionalidad real es:

$$Adecuacion\ funcional = \frac{490,8}{552,15} * 100\%$$

$$Adecuacion\ funcional = 0,88 * 100\%$$

$$Adecuacion\ funcional = 88\%$$

El resultado quiere decir que la funcionalidad del sistema es del 88%, lo que nos dice que el sistema funciona sin riesgos a fallar con operatividad constante y un 12% no satisface dichas necesidades.

### 3.3.1.2. Fiabilidad

La fiabilidad es la capacidad de un sistema o componente para desempeñar las funciones especificadas. Para este punto se realizó el análisis de nivel de fiabilidad del sistema, donde se considera la fiabilidad de cada módulo o subsistema de forma independiente.

Para calcular la fiabilidad de cada módulo se usó la fórmula:

$$R(t) = e^{-\lambda t}$$

Donde:

**$R(t)$** : Fiabilidad de un componente o subsistema  $t$ .

**$\lambda$** : Tasa de constantes de fallo ( $\lambda = N^\circ$  de fallas de acceso /  $N^\circ$  total de accesos al sistema).

**$t$** : Periodo de operación de tiempo.

**$e^{-\lambda t}$** : Probabilidad de falta de un componente o subsistema en el tiempo  $t$ .

Luego de realizar pruebas de cada módulo en un tiempo de 4 Hrs. continuas se logró llenar la siguiente tabla:

**Tabla 28. Valores de fiabilidad de cada modulo**

N°	MODULO	$\lambda$	t	R(t)
1	Modulo Seguridad	0.001	4Hrs	0.95
2	Modulo Registro	0.022	4Hrs	0.95
3	Modulo Registro de datos RFID	0.025	4Hrs	0.99
4	Modulo Reportes	0.018	4Hrs	0.97
5	Modulo Herramientas de apoyo	0.005	4Hrs	0.99

**Fuente:** (Elaboración propia)

Para calcular la fiabilidad del sistema completo, se vio que falla la autenticación (módulo de seguridad), no se podrá acceder a los demás módulos, por tanto, la conexión es en serie con los demás. Y el resto de los módulos están conectados en paralelo ya que funcionan independientemente de los demás, es por eso que la fiabilidad del sistema estaría dada por la formula.

$$\mathbf{Fiabilidad = R_s * R_p}$$

Donde:  $R_s = R_p = 0.95$  y  $R_p = \frac{\sum_{i=2}^5 (R_i * P_i)}{\sum_{i=2}^5 P_i}$

En la fórmula de  $R_s$ , la variable  $P_i$  es la participación en el equipo de desarrollo del modulo y como la participación fue al 100% entonces  $P_i = 1$ , así se tiene el siguiente resultado:

$$R_p = \frac{\sum_{i=2}^5 (R_i)}{\sum_{i=2}^5 P_i}$$

$$R = \frac{0,95 + 0,99 + 0,97 + 0,99}{4}$$

$$R = \frac{3,9}{4} = 0,98$$

Por lo tanto, la fiabilidad del sistema está dada por:

$$\mathbf{Fiabilidad = R_s * R_p = 0,95 * 0,98 = 0,93}$$

$$\mathbf{Fiabilidad = 93\%}$$

De lo cual se puede decir que existe un 7 % de probabilidad de que el sistema presente algún fallo cuando se exceda un tiempo de uso continuo, debido a que puedan existir fallas con la conexión del sistema a la base de datos, conexión del cliente al sistema, uso incorrecto del sistema por parte del usuario, errores en la entrada de datos.

### 3.3.1.3. Usabilidad

La usabilidad es la capacidad del software de ser entendido, aprendido y usado de forma fácil y atractiva. Para determinar el porcentaje de la usabilidad del sistema se optó por realizar una encuesta, donde los cuales califican en una ponderación al 100%.

Para determinar la usabilidad del sistema se utilizará la siguiente ecuación:

$$FU = \left[ \left( \sum \frac{X_i}{n} \right) * 100 \right]$$

Donde:

*X<sub>i</sub>*: Es la sumatoria de valores

*n*: Es el número de preguntas

Para responder a las preguntas, se debe considerar la siguiente tabla, esta los parámetros de medición:

**Tabla 29. Parámetros de medición**

ESCALA	VALOR
Muy Bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

**Fuente:** (Elaboración propia)

**Tabla 30. Parámetros de medición sobre la usabilidad del sistema**

N°	PREGUNTAS	RESPUESTA		EVALUACION
		SI	NO	%
1	¿Puedo utilizar fácilmente el sistema?	5	0	1
2	¿Los controles de operaciones que vio fueron de su agrado?	4	1	0,8
3	¿Los controles de operaciones que vio fueron fáciles de comprender?	5	0	1
4	¿El sistema permitió la retroalimentación de la información?	4	1	0,8
5	¿La respuesta del sistema es satisfactoria?	5	0	1
6	¿El sistema le facilita el trabajo?	5	0	1
7	¿El sistema reduce su tiempo de trabajo?	5	0	1
8	¿Le parecen complicadas las funciones del sistema?	5	0	1
9	¿El sistema facilita las respuestas requeridas?	5	0	1
10	¿El sistema no presento errores?	4	1	0,8
<b>TOTAL</b>				<b>9,4</b>

**Fuente:** (Elaboración propia)

Calculamos la usabilidad con la ecuación anterior:

$$FU = \left[ \left( \frac{9,4}{10} \right) * 100 \right]$$

$$FU = [0.94 * 100]$$

$$\mathbf{FU = 94\%}$$

Por lo tanto, existe un 94% de comprensión o entendimiento de los usuarios con respecto a la capacidad del sistema.

### 3.3.1.4. Mantenibilidad

La aplicación esta realiza por módulos por lo cual es independiente a los demás

La mantenibilidad es la cualidad que tiene el software para ser modificado, incluyendo correcciones o mejoras del software, a cambios en el entorno y especificaciones de requerimientos funcionales. Para poder calcular un índice de madures del software (IMS), se indica la estabilidad de un producto de software.

El índice de madurez se calcula con la siguiente formula:

$$IMS = \frac{M_t - (F_a + F_b + F_c)}{M_t}$$

Donde:

$M_t$ : Numero de módulos en la versión actual.

$F_a$ : Numero de módulos en la versión actual que se ha cambiado.

$F_b$ : Números de módulos en la versión actual que se han añadido.

$F_c$ : Numero de módulos en la versión anterior que se han borrado en la versión actual.

Recopilando la información requerida por la formula se obtuvo la información que se muestra en la siguiente tabla:

**Tabla 31. Información requerida por el IMS**

INFORMACION	VALOR
$M_t$	8
$F_a$	1
$F_b$	0
$F_c$	0

**Fuente:** (Elaboración propia)

Ahora calculemos el IMS, usando los valores obtenidos en la formula anterior:

$$IMS = \frac{8 - (1 + 0 + 0)}{8}$$

$$IMS = \frac{7}{8}$$

$$IMS = 0,88 * 100\%$$

$$IMS = 88\%$$

Con ese resultado se concluyó que el sistema, tiene un índice de madurez de software del 88% que es la facilidad de mantenimiento, pero el 12% restante sería el margen de error por los cambios y modificaciones.

#### **3.3.1.5. Portabilidad**

La portabilidad es la capacidad que tiene el software para ser trasladado de un entorno a otro sin ningún problema.

El sistema de control y registro de parqueo aplicando la tecnología RFID, esta diseñado en un entorno de acceso vía web y mide la portabilidad en lado del servidor y lado del cliente, la portabilidad del software se enfoca en tres aspectos:

- Hardware del servidor
- Sistema operativo del servidor
- Software del servidor

El sistema de control y registro de parqueo aplicando la tecnología RFID, es portable en sus diferentes entornos tanto en hardware y software.

**Tabla 32. Fator de portabilidad**

<b>FACTOR DE PORTABILIDAD</b>	<b>VALOR EN %</b>
Puede ser transferido de un entorno a otro	85
Se puede adaptar a otros ambientes con factibilidad	95
Es fácil de instalar	95
<b>TOTAL</b>	<b>91%</b>

**Fuente:** (Elaboración propia)

### **3.3.1.6. Calidad Global**

Una vez calculando los porcentajes de los diferentes atributos que el sistema tiene según lo propuesto por el estándar de calidad ISO/IEC 25010, se procedió a calcular la calidad global del sistema, la cual se visualiza en la siguiente tabla:

**Tabla 33. Calidad Global del Sistema**

<b>ATRIBUTO</b>	<b>VALOR DE %</b>
Adecuación Funcional	88
Fiabilidad	93
Usabilidad	94
Mantenibilidad	88
Portabilidad	91
<b>Calidad Global</b>	<b>91%</b>

**Fuente:** (Elaboración propia)

Con el resultado obtenido en la calidad global del sistema de control y seguimiento del parqueo Gonzales es del 91%.

### 3.4. Análisis de costos

#### 3.4.1. Método de estimación COCOMO II

COCOMO II define tres modos de desarrollo: basándose en la tabla 4. del marco teórico se trabajó con el modo orgánico.

**Tabla 34. Cálculo de atributos  $M(X_i)$**

Factores (cost-drivers)	Valor de los factores					
	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra
<b>Atributo del producto</b>						
Fiabilidad requerida	0,75	0,88	1,00	1,15	1,40	
Tamaño de la base de datos		0,94	1,00	1,08	1,16	
Complejidad del software	0,70	0,85	1,00	1,15	1,30	1,65
<b>Atributos del ordenador</b>						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria			1,00	1,06	1,30	1,66
Volatilidad del hardware		0,87	1,00	1,15	1,30	
Restricciones de tiempo de respuesta		0,87	1,00	1,07	1,15	
<b>Atributos del personal</b>						
Calidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia con el tipo de aplicación	1,29	1,13	1,00	0,91	0,82	
Experiencia con el hardware	1,21	1,10	1,00	0,90		
Experiencia con el lenguaje de programación	1,14	1,17	1,00	0,95		
Calidad de programadores	1,42	1,17	1,00	0,86	0,70	
<b>Atributos del proyecto</b>						
Técnicas modernas de programación	1,24	1,10	1,00	0,91	0,82	
Empleo de herramientas	1,24	1,10	1,00	0,91	0,83	
Restricciones a la duración del proyecto	1,23	1,08	1,00	1,04	1,10	
<b>TOTAL, <math>M(X_i)</math></b>	<b>0,53</b>					

Fuente: (Elaboración propia)

## I. Estimación de esfuerzo

$$E = a * kLDC^b * \prod_{i=1}^{n=15} M(Xi)$$

$$E = 2.4 * 13,128^{1.05} * 0,53$$

$$E = 18,99 \frac{\text{Personas}}{\text{Mes}} \Rightarrow \text{Equivale a } 19 \frac{\text{Personas}}{\text{Mes}}$$

## II. Estimación de tiempo

$$t = c * E^d \text{Mes}$$

$$t = 2.5 * 18,99^{0.38}$$

$$t = 7,65 \text{ Meses} = \text{Equivale a } 8 \text{ Meses}$$

## III. Otras operaciones

### Número de personas requerido para el desarrollo

$$NP = \frac{E}{T} \text{ Personas}$$

$$NP = \frac{18,99}{7,65} \text{ Personas}$$

$$NP = 2,48 \text{ Personas} = \text{Equivale a } 3 \text{ Personas}$$

### Costo total del proyecto

$$CT = NP * T * \text{Sueldo mes}$$

$$CT = 2,48 * 7,65 * 500\$$$

$$CT = 9486\$$$

Entonces se requiere estimando 3 personas un trabajo de 8 meses para el desarrollo del sistema con un costo total de 9486 \$ dólares.

## Otros costos

**Tabla 35. Tabla general de costos**

<b>Costos de software</b>	
- Licencia del software usado	<b>0</b>
- Software desarrollado	9486\$
<b>Costos de hardware</b>	
<b>Adquisición de equipos</b>	
- Computadora personal HP	715\$
- Impresora Térmica Epson TM-T20II	350\$
- Lector de RFID y tarjetas	150\$
- Barrera Vehicular	850\$
- Material de escritorio	450\$
<b>COSTO TOTAL DEL PROYECTO</b>	<b>12.0001 \$</b>

**Fuente:** (Elaboración propia)

## 3.5. SEGURIDAD

### 3.5.1. Seguridad a nivel Base de Datos

La base de datos al igual que el sistema cuenta con restricciones de acceso de usuario el cual es configurado para comunicación con el código fuente del sistema. El acceso de la base de datos solo está autorizado para el administrador que tiene los privilegios de acceso a registrar modificar eliminar y búsquedas de información.

### 3.5.2. Seguridad a nivel de Aplicación

Las principales herramientas para la seguridad del sistema son: la Autenticación, control de acceso cifrado de contraseñas

#### ✓ Autenticación

Para la autenticación de usuario el sistema realiza la búsqueda en base de datos el cual si encuentra coincidencias pues le dará acceso al sistema caso contrario se restringirá el acceso al sistema.

✓ **Control de acceso**

Luego de la autenticación viene el control de acceso, que se asegura de que solo las personas adecuadas o autorizadas puedan ver, modificar, agregar y/o eliminar la información.

✓ **Cifrado**

Para el control de acceso de sistema en cuanto a las contraseñas de cada usuario. Las mismas son encriptadas para luego ser guardadas en la base de datos

### 3.6. Sistema de gestión de seguridad de la información ISO/27001

#### 3.6.1. Seguridad Lógica

- ✓ **Copias de Seguridad:** Los respaldos o Back-up de la base de datos del sistema se debe realizar de acuerdo a la siguiente tabla:

**Tabla 36. Copias de seguridad**

DESCRIPCION	DURACION
Registro de datos	1 vez al día, Quincenal, mensual o anual
Registros orden de Entradas y salidas	1 vez al día, Quincenal, mensual o anual

**Fuente:** (Elaboración propia)

- El administrador deberá cambiar su password de sistema periódicamente 1 vez al mes para mayor seguridad.
  - En caso de ser usuario auxiliar se recomienda cambiar el password periódicamente.
- ✓ **Identificación y autenticación:** Los usuarios auxiliares cuentan con usuarios y contraseñas controlada para acceder al sistema, de esta manera se previene el ingreso al sistema de personas que no son usuarios.

- ✓ **Encriptación:** Para la seguridad de password, estará encriptado con el algoritmo Hash que es una encriptación de alta seguridad para el ingreso al sistema.

### 3.6.2. Seguridad física

- ✓ **Seguridad física y del entorno:** Los back -up de la base de datos deberán ser protegidas en áreas seguras, donde solo el administrador tenga acceso.

Manejar el back-up de acuerdo a la fecha que se realizó, para mayor organización.

### 3.6.3. Seguridad Organizativa

La información generada por el sistema debe ser procesada y recibir un nivel de protección apropiada:

- ✓ **Gestión de Archivos:** para este punto se recomienda etiquetar las diferentes copias de seguridad (backup) con la fecha de respaldo como identificador único.
- ✓ **Recursos humanos:** se recomienda la administración permanente de los registros de los diferentes usuarios, de esta forma se darán de baja a registros que no formen parte del personal administrativo presente.

**CAPITULO IV**

**CONCLUSIONES Y**

**RECOMENDACIONES**



#### 4.1. Conclusiones

Se concluye con los objetivos planteados, habiendo realizado un estudio del sistema actual en el proceso de registro de información con algunas irregularidades, que existía tomando en cuenta que, con el presente proyecto, se logró centralizar la información y efectuar un control y seguimientos de los registros datos, inventarios y cotizaciones del parqueo, así coadyuvando en una mejor administración de información del parqueo.

- ✓ Se obtuvo la información necesaria gracias al administrador quien nos facilitó la información para realizar el sistema.
- ✓ Se realizó módulos de registros de datos, para tener una mejor administración y tener búsquedas eficientes.
- ✓ Se realizó un módulo de control de ingreso y salidas de los vehículos, aplicando la tecnología RFID.
- ✓ Se centralizo la información en una base de datos, de todos los datos registrados dentro del sistema, evitando una pérdida de información.
- ✓ Se mejoró la atención de los ingresos y salidas de los vehículos, mejorando el tiempo con eficiencia.
- ✓ Para la realización de estimación de costos del proyecto se utilizó el modelo de COCOMO II tomando en cuenta el Diseño anticipado, por medio de puntos de función lo cual permitió determinar el esfuerzo, costo y tiempo del desarrollo.

Logrando todos los objetivos específicos, se concluye con el desarrollo e implementación de el “SISTEMA DE CONTROL Y SEGUIMIENTO DE PARQUEO APLICANDO LA TECNOLOGIA RFID”, por lo que es un aporte tecnológico ya que redujo tiempo de registros, consultas, búsquedas de la información y el manejo se realiza de forma segura y confiable.

## 4.2. Recomendaciones

Se recomienda los siguientes puntos:

- ✓ Capacitar a los nuevos usuarios, para poder operar el sistema de forma correcta.
- ✓ Para resguardar la información, el administrador del sistema debe realizar copias de seguridad de la base de datos cada 15 días.
- ✓ Se recomienda mucha discreción en el manejo de sus usuarios y contraseñas ya que el sistema contiene información de mucha importancia.
- ✓ Mantener un control acerca del equipo que hace de servidor físico.
- ✓ Se recomienda cambiar continuamente las contraseñas para la seguridad del sistema y proteger el acceso a personas no autorizadas.
- ✓ Se recomienda leer el manual del sistema antes de operarlo

## BIBLIOGRAFÍA

Aycart Perez, Golbert G., Hernandez M. & Mas Hernandez. (2007), *Ingeniería del software en entornos de SL*.

Adriana Gómez, María López, (2017). *Un modelo de estimación de proyectos de software Cocomo II*.

Barnola Augé, P.(2010). *Introducción a la creación de páginas web*. Barcelona: Euroca Medua, SL.

Berni Millet, P.,& Gil de Iglesia, D. (2010). *puesta en marcha de un SGBD y un servidor web local Barcelona: Euroca Medua, SL*.

Departamento de Ciencias de la Computación. (2002). *DESCALI*. ISBN 84—782782—9051.

Díaz Pulido, José Arturo. (2017). *Diseño de un e-portafolio basado en la metodología OOHDM para gestionar la información de proyectos en informática*.

Echeverría Broncano M. (2009). *Diseño de un sitio web para la venta de suministros de oficina y material publicitario*. Recuperado de:

<https://bibdigital.epn.edu.ec/bitstream/15000/1534/1/CD-2195.pdf>

El Safi Ahmed Adam, R. (2011). *RFID Access control system (Thesis)*. University of Khartoum. Recuperado a partir de <http://khartoumspace.uofk.edu/handle/123456789/18548>

Fidias G. Arias (2006) *El Proyecto de Investigación, Introducción a la Metodología científica*. 6ta Ediccion, Pág.

Javier Eguiluz Perez, (2009). *Introducción a JavaScript. La mejor guía*.

Ilyas, M., & Ahson, S. (2008). *RFID handbook: applications, technology, security, and privacy*. Boca Raton: CRC Press

J. LANDT, (2005). "The history of rfid," *IEEE Potentials*.

J. LANDT, (, 2001). *“Shrouds of time, history of rfid,” AIM Publication.*

Jimmy R. Molina, Mariuxi P. Zea, María J. Contento y Fabricio G. García, (2018), *Computacion de Metodologías en aplicaciones web.* Recuperado de: <https://www.3ciencias.com/wp-content/uploads/2018/03/art1.pdf>

Herrán Gascón, A., Cortina Selva, M. y González Sánchez, I. (2006). *La muerte y su didáctica: manual para educación infantil, primaria y secundaria.* Madrid: Universitas.

Observatorio Regional de la Sociedad de la Información, ORSI. (2007). *Tecnología de identificación por Radiofrecuencia y sus principales aplicaciones.pdf Castilla y León, España.* Recuperado a partir de:

<http://www.jcyl.es/web/jcyl/binarios/211/716/RFID.pdf?blobheader=aplicati>

Oscar Bertiglio. (2015). *Introducción a la teoría general de sistemas.* Mexico: Limusa.

Landt, J. (2005). *The history of RFID. IEEE Potentials, 24(4), 8-11.* Recuperado de: <http://doi.org/10.1109/MP.2005.1549751>

PORTILLO, J. BERMEJO, (2007). *“Tecnología de identificación por radiofrecuencia (rfid): aplicaciones en el ámbito de la salud,”.*

Rodríguez, V. (2009). *Metodologías de diseño usadas en ingeniería web, su vinculación con las NTICS.* Mar del Plata, Argentina,

Roger S. Pressman R. (2005). *Ingeniería Del Software. Un Enfoque Práctico Séptima edición*

Roger S. Pressman, R. (2010). *Ingeniería del Software un enfoque práctico.* Madrid España: McGraw Hill.

Portal ISO/25000. (2020). *ISO 25000. Obtenido de ISO 25000.* Recuperado de: <https://iso25000.com/>

Vargas V. (2013), *Sistema de Control de Acceso y Monitoreo con LA tecnología RFID*. Guayaquil: Universidad Politécnica Salesiana Ecuador. Recuperado de: <https://dspace.ups.edu.ec/bitstream/123456789/5380/1/UPS-GT000473.pdf>

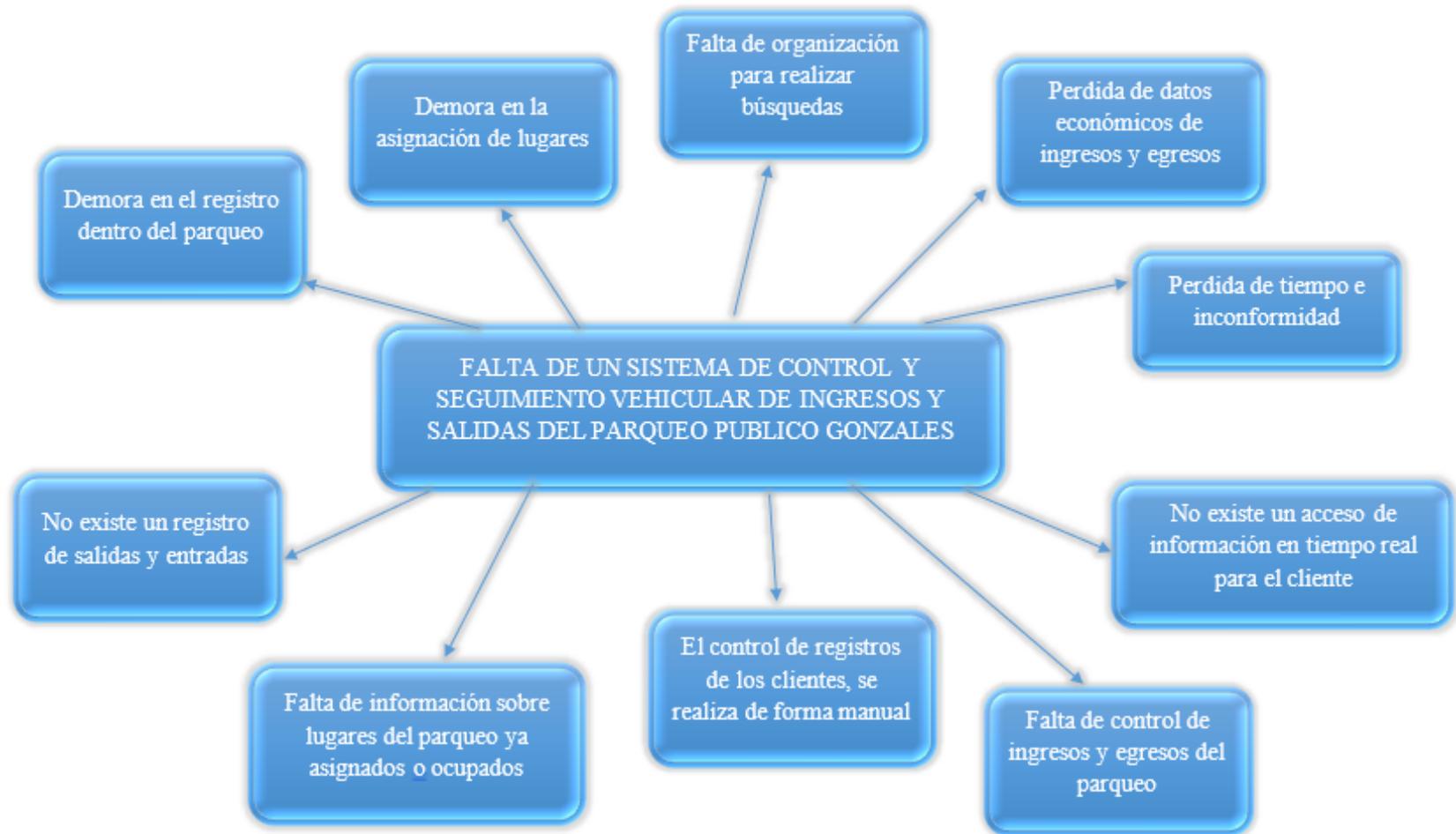
Nava Jorge D., (2013). *Diseño de un sistema de control y monitoreo de alumnos de un preescolar por medio de RFID*. Recuperado de: <http://biblioteca2.ucab.edu.ve/anexos/biblioteca/marc/texto/AAS7504.pdf>

# ANEXOS



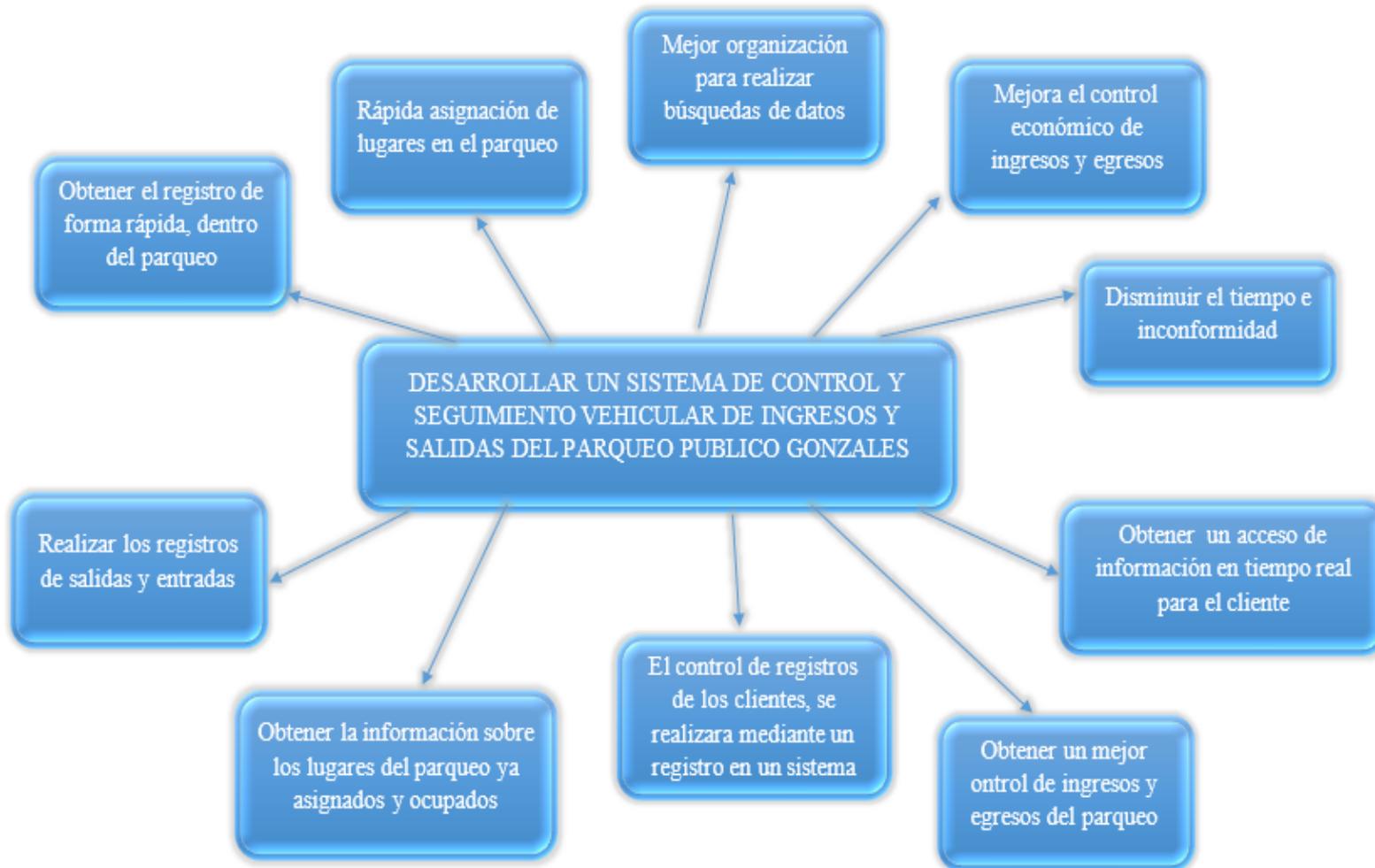
ANEXO A. ARBOL DE PROBLEMAS

**ARBOL DE PROBLEMAS**



ANEXO B. ARBOL DE OBJETIVOS

**ARBOL DE OBJETIVOS**



ANEXO C. MANUAL DE USUARIO

# MANUAL DE USUARIO

**SISTEMA DE CONTROL Y SEGUIMIENTO  
DE PARQUEO APLICANDO LA  
TECNOLOGIA RFID**



## ÍNDICE DEL MANUAL DE USUARIO

1. INTRODUCCION.....	1
2. INGRESO AL SISTEMA .....	1
3. PANTALLA PRINCIPAL.....	3
4. PANTALLA DE MODULO DE USUARIO.....	8
5. PANTALLA DE MODULO DE TARIFAS.....	9
6. PANTALLA DE MODULO DE PROPIETARIOS .....	11
7. PANTALLA DE MODULO DE VEHICULOS .....	12
8. PANTALLA DE TIPO DE VEHÍCULOS.....	14
9. PANTALLA DE MODULO DE SECCIONES.....	15
10. PANTALLA DE MÓDULO DE COBROS .....	17
11. PANTALLA DE MODULO DE INGRESOS Y SALIDAS.....	18
12. PANTALLA DE REPORTES.....	19
13. PANTALLA DE LA VISUALIZACION PARA EL CLIENTE.....	22
14. PANTALLA DE DATOS DE LA EMPRESA .....	22

## 1. INTRODUCCION

Actualmente los parqueos en su mayoría no cuentan con un sistema de control de escritorio, pero con el acceso de tecnologías de innovación que brinde una información oportuna, confiable tendrán una mejor administración de la información del lugar.

*El SISTEMA DE CONTROL Y SEGUIMIENTO DE PARQUEO APLICANDO LA TECNOLOGIA RFID*, está pensada y diseñada para facilitar a los usuarios en la mejora de administración de la información.

Para acceder al sistema el usuario entra a la página principal donde podrá ingresar al sistema, donde el administrador debe hacer uso de su nombre de usuario y contraseña.

## 2. INGRESO AL SISTEMA

### INGRESO AL SISTEMA



**Descripción:** Ingreso o Inicio de Sesión...

1

“Nombre del Parqueo”

PARQUEO GONZALES

“Logo del Parqueo”

2



3

“Mensaje de Bienvenida”

Escribe tu usuario y contraseña

4

“Campo para ingresar el nombre de usuario asignado”

5

“Campo para ingresar contraseña”

6

“Botón para ingresar al sistema”

ACCEDER

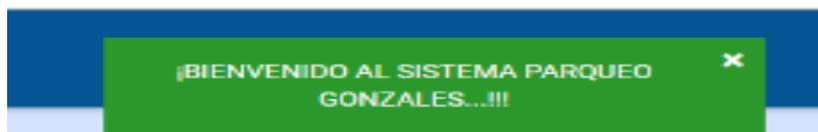
### 3. PANTALLA PRINCIPAL

## PANTALLA PRINCIPAL



**Descripción:** Pantalla principal del sistema.

1 “Mensaje de Bienvenida al sistema”



**Descripción:** Al ingresar al sistema se muestra un mensaje de bienvenida.

2 “Nombre del Parque”



**Descripción:** Muestra el nombre del parque en la pantalla principal.

3

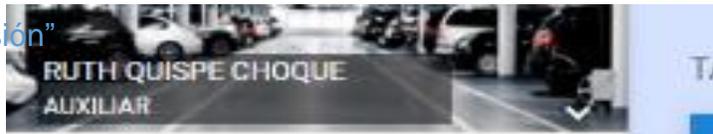
“Imagen de Administrador o Usuario Auxiliar que inicio la sesión”



**Descripción:** Se muestra la foto del que inicio la sesión.

4

“Nombre del Administrador o Usuario Auxiliar y su Rango que inicio la sesión”



**Descripción:** Muestra el nombre del Usuario del que inicio la sesión y muestra su tipo de rango.

5

“Botón de inicio”



**Descripción:** El botón de inicio, es donde se lleva a la pantalla principal, privilegio administrador.

6

“ Botón de usuario”



**Descripción:** Es el botón de usuarios, donde lleva al módulo de usuarios, privilegio administrador.

7

“Botón de Tarifas”



**Descripción:** Es el botón de las tarifas, donde lleva al módulo de tarifas, privilegio administrador.

8 “Botón de Propietarios”



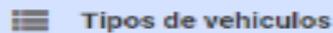
**Descripción:** Es el botón de los propietarios, donde lleva al módulo de propietarios, privilegio administrador.

9 “Botón de Vehículos”



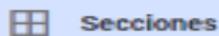
**Descripción:** Es el botón de los vehículos, donde lleva al módulo de vehículos, privilegio administrador.

10 “Botón de Tipo de Vehículos”



**Descripción:** Es el botón de tipo de Vehículos, privilegio administrador.

11 “Botón de Secciones”



**Descripción:** Es el botón de las Secciones, donde lleva al módulo de secciones, privilegio administrador.

12 “Botón de cobros”



**Descripción:** Es el botón de los cobros, donde lleva a los cobros, privilegio administrador.

13

### “Botón de ingresos y salidas”



**Descripción:** Es el botón de ingresos y salidas, donde lleva al módulo de ingresos y salidas, privilegio administrador.

14

### “Botón de reportes”



**Descripción:** Es el botón de los reportes, donde lleva la lista de los reportes, privilegio administrador.

15

### “Botón de empresa”



**Descripción:** Es el botón de la empresa, privilegio administrador.

16

### “Notificaciones”



**Descripción:** Icono de notificación, cuando los vehículos realizan el ingreso y la salida del parqueo.

17

### “Salir”



**Descripción:** Icono de Salir de sistema, cerrar la sesión y te lanza a Login.

18

### “Icono de conteo de vehículos al día”



**Descripción:** Es el icono donde muestra el conteo total de ingresos al día de los vehículos que usaron el servicio del parqueo.

19

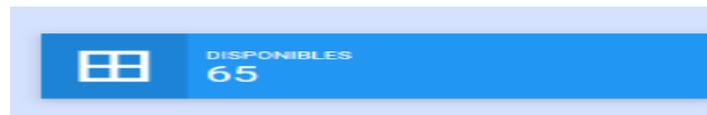
### “Icono de conteo de secciones creadas”



**Descripción:** Es el icono donde muestra el conteo total de secciones disponibles o creadas.

20

### “Icono de conteo de espacios disponibles”



**Descripción:** Es el icono donde muestra el conteo total de los lugares disponibles de los espacios del parqueo.

21

### “Icono de conteo de los vehículos”



**Descripción:** Es el icono donde muestra el conteo total de los vehículos registrados en el sistema.

22

### “Estadísticas de lugares disponibles”

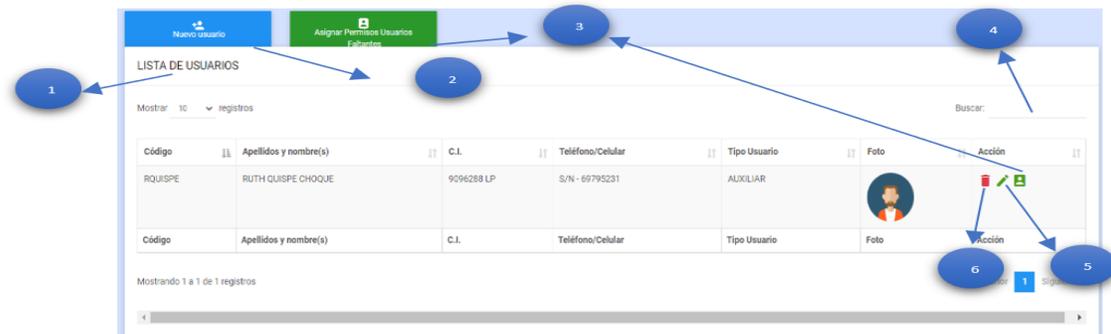


**Descripción:** Es una estadística de lugares disponibles por secciones, donde muestra los lugares el conteo de lugares disponibles y ocupados.

#### 4. PANTALLA DE MODULO DE USUARIO

### MÓDULO DE USUARIO

**SOLO EL ADMINISTRADOR TIENE ACCESO AL MODULO USUARIOS.**



**Descripción:** Es el formulario de usuarios.

#### 1 “Lista de Usuarios”

Mostrar 10 registros

Código	Apellidos y nombre(s)	C.I.	Teléfono/Celular	Tipo Usuario	Foto	Acción
RQUISPE	RUTH QUISPE CHOQUE	9096288 LP	S/N - 69795231	AUXILIAR		

Mostrando 1 a 1 de 1 registros

**Descripción:** Es el formulario donde se muestra la lista de los usuarios, registrados en el sistema, con su nombre de usuario, nombre completo, numero de cedula de identidad, numero de celular, tipo de usuario, la foto y la acción.

#### 2 “Botón para adicionar un nuevo usuario”

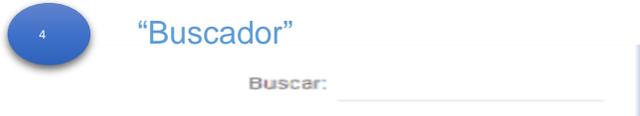


**Descripción:** Es el botón para adicionar o crear un nuevo usuario, donde debe llenar sus datos y el rango que se le asigne, también debe subir una foto y guardarlo y se registra en el sistema.

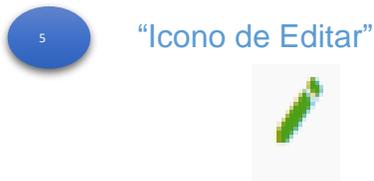
#### 3 “Botón para asignar roles”



**Descripción:** Es el botón para asignar roles a los usuarios, el administrador tiene el control del todo el sistema, pero el puede dar acceso a módulos o quitar privilegios.



**Descripción:** Es para realizar búsquedas de los datos registrados.



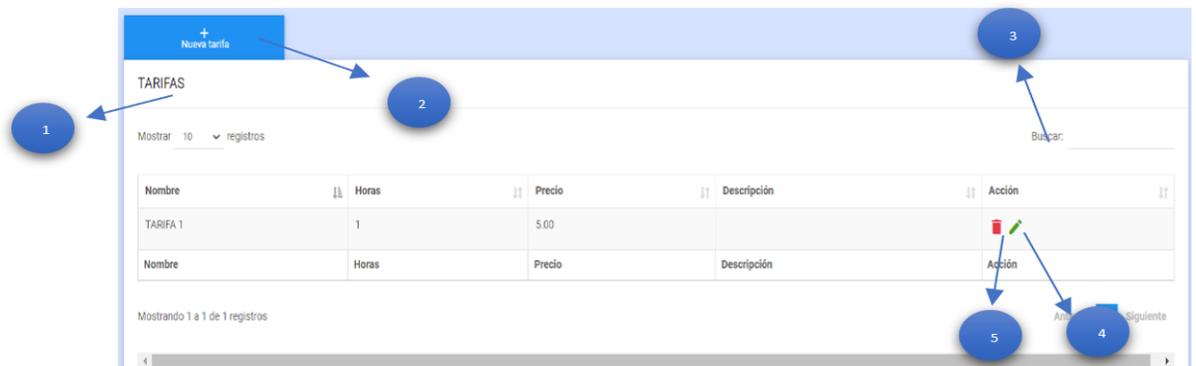
**Descripción:** Es el botón o icono de editar, donde lleva a un formulario para editar datos ya registrados.



**Descripción:** Es el botón o icono eliminar, donde se eliminar un registro ya registrado.

## 5. PANTALLA DE MODULO DE TARIFAS

### MÓDULO DE TARIFAS



**Descripción:** Es el formulario de tarifas.

1

## “Lista de Tarifas”

TARIFAS

Mostrar 10 registros Buscar: \_\_\_\_\_

Nombre	Horas	Precio	Descripción	Acción
TARIFA 1	1	5.00		 
Nombre	Horas	Precio	Descripción	Acción

**Descripción:** Es el formulario donde se muestra la lista de las tarifas, registrados en el sistema, con su nombre, hora, precio, descripción, y la acción.

2

## “Botón para adicionar una nueva tarifa”



**Descripción:** Es el botón para adicionar o crear una nueva tarifa, donde debe llenar los datos requeridos y guardarlo.

3

## “Buscador”



**Descripción:** Es para realizar búsquedas de los datos registrados.

4

## “Icono de Editar”



**Descripción:** Es el botón o icono de editar, donde lleva a un formulario para editar datos ya registrados.

5

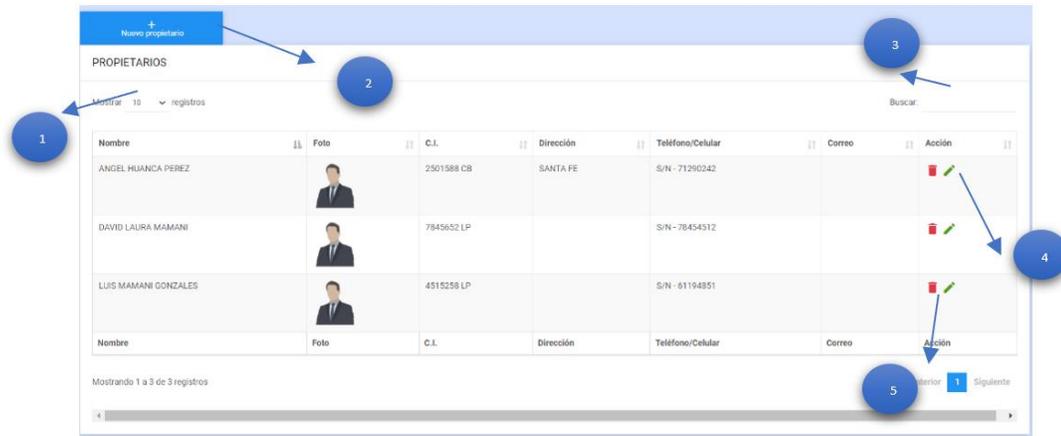
## “Icono de Eliminar”



**Descripción:** Es el botón o icono eliminar, donde se eliminar un registro ya registrado.

## 6. PANTALLA DE MODULO DE PROPIETARIOS

### MÓDULO DE PROPIETARIOS



**Descripción:** Es el formulario de propietarios.

#### 1 “Lista de Propietarios”

The screenshot shows the 'Lista de Propietarios' table. It has the same structure as the main screenshot, with columns for 'Nombre', 'Foto', 'C.I.', 'Dirección', 'Teléfono/Celular', 'Correo', and 'Acción'. The table contains three rows of data for 'ANGEL HUANCA PEREZ', 'DAVID LAURA MAMANI', and 'LUIS MAMANI GONZALES'. The 'Acción' column contains icons for edit and delete for each row.

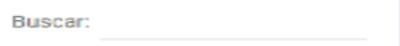
**Descripción:** Es el formulario donde se muestra la lista de los propietarios, registrados en el sistema, con su nombre, foto, numero de cedula de identidad, dirección, teléfono/celular, correo y la acción.

#### 2 “Botón para adicionar una nueva propietario”



**Descripción:** Es el botón para adicionar o crear unos nuevos propietarios (Clientes), donde debe llenar los datos requeridos y guardarlo.

#### 3 “Buscador”



**Descripción:** Es para realizar búsquedas de los datos registrado.

4 “Icono de Editar”



**Descripción:** Es el botón o Icono de editar, donde lleva a un formulario para editar datos ya registrados.

5 “Icono de Eliminar”



Descripción: Es el botón o icono eliminar, donde se eliminar un registro ya registrado.

## 7. PANTALLA DE MODULO DE VEHICULOS

### MÓDULO DE VEHÍCULOS

Color	Foto	Placa	Marca	Modelo	Propietario	Tarifa	RFID	Acción
BLANCO		545-HGHNN	TOYOTA	2000	LUIS MAMANI GONZALEZ	TARIFA 1	464646	[Iconos de acción]
CAFI		4519-HGHG	TOYOTA	2000	DAVID LAIBRA MAMANI	TARIFA 1	0733062676	[Iconos de acción]
NEGRO		4518-LJNH	HYUNDAI SANTA FE	2019	LUIS MAMANI GONZALEZ	TARIFA 1	0544978404	[Iconos de acción]
NEGRO		4757-006	NISSAN	1999	DAVID LAIBRA MAMANI	TARIFA 1	564654	[Iconos de acción]
ROJO		1243-HH00H	TOYOTA	1955	ANGEL HIRACIA PEREZ	TARIFA 1	64666	[Iconos de acción]

**Descripción:** Es el formulario de Vehículos.

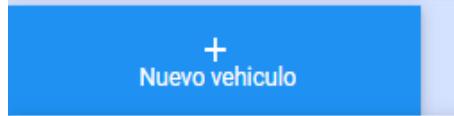
1 “Lista de Vehículos”

Color	Foto	Placa	Marca	Modelo	Propietario	Tarifa	RFID	Acción
BLANCO		545-HGHNN	TOYOTA	2000	LUIS MAMANI GONZALES	TARIFA 1	464646	[Iconos de acción]

**Descripción:** Es el formulario donde se muestra la lista de los vehículos, registrados en el sistema, el color, la foto, placa, marca, modelo, propietario, tarifa, RFID y la acción.

2

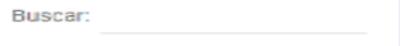
“Botón para adicionar un nuevo vehículo”



**Descripción:** Es el botón para adicionar o crear una nuevos Vehículos donde debe llenar los datos requeridos y guardarlo.

3

“Buscador”



**Descripción:** Es para realizar búsquedas de los datos registrados.

4

“Icono de Editar”



**Descripción:** Es el botón o Icono de editar, donde lleva a un formulario para editar datos ya registrados

5

“Icono de Eliminar”



**Descripción:** Es el botón o icono eliminar, donde se eliminar un registro ya registrado.

5

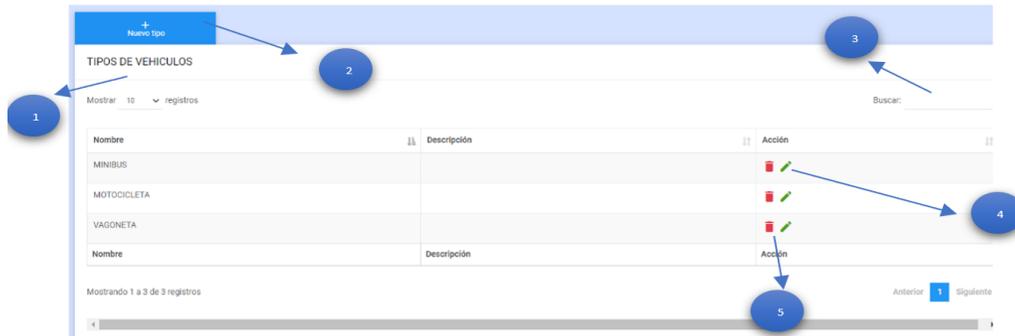
“Icono de Kardex”



**Descripción:** Es el botón o icono una vista que lleva reporte, donde muestra los datos de los vehículos.

## 8. PANTALLA DE TIPO DE VEHÍCULOS

### MÓDULO DE TIPO DE VEHÍCULOS



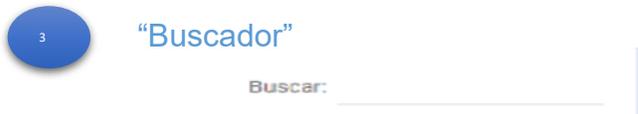
**Descripción:** Es el formulario de tipo de Vehículos.



**Descripción:** Es el formulario donde se muestra la lista de los propietarios, registrados en el sistema, con su nombre, foto, numero de cedula de identidad, dirección, teléfono/celular, correo y la acción.



**Descripción:** Es el botón para adicionar o crear unos nuevos propietarios (Clientes), donde debe llenar los datos requeridos y guardarlo.



**Descripción:** Es para realizar búsquedas de los datos registrados.

4

### “Icono de Editar”



**Descripción:** Es el botón o Icono de editar, donde lleva a un formulario para editar datos ya registrados.

5

### “Icono de Eliminar”



**Descripción:** Es el botón o icono eliminar, donde se eliminar un registro ya registrado.

## 9. PANTALLA DE MODULO DE SECCIONES

**MÓDULO DE SECCIONES**

Nombre	Capacidad	Descripción	Mapeo	Acción
SECCION 1	30		Ver mapeo	[Icono de eliminar] [Icono de editar]
SECCION 2	30		Ver mapeo	[Icono de eliminar] [Icono de editar]
SECCION PRUEBA	5		Ver mapeo	[Icono de eliminar] [Icono de editar]

**Descripción:** Es el formulario de Secciones.

1

### “Lista de Secciones”

LISTA DE SECCIONES - UBICACIONES

Mostrar 10 registros

Nombre	Capacidad	Descripción	Mapeo	Acción
SECCION 1	30		Ver mapeo	[Icono de eliminar] [Icono de editar]

**Descripción:** Es el formulario donde se muestra la lista de los Secciones, registrados en el sistema, nombre, capacidad, descripción mapeo y la acción.

2

“Botón para adicionar una nueva sección”



**Descripción:** Es el botón para adicionar o crear unos nuevos propietarios (Clientes), donde debe llenar los datos requeridos y guardarlo.

3

“Buscador”



**Descripción:** Es para realizar búsquedas de los datos registrados.

4

“Icono de Editar”



**Descripción:** Es el botón o Icono de editar, donde lleva a un formulario para editar datos ya registrados.

5

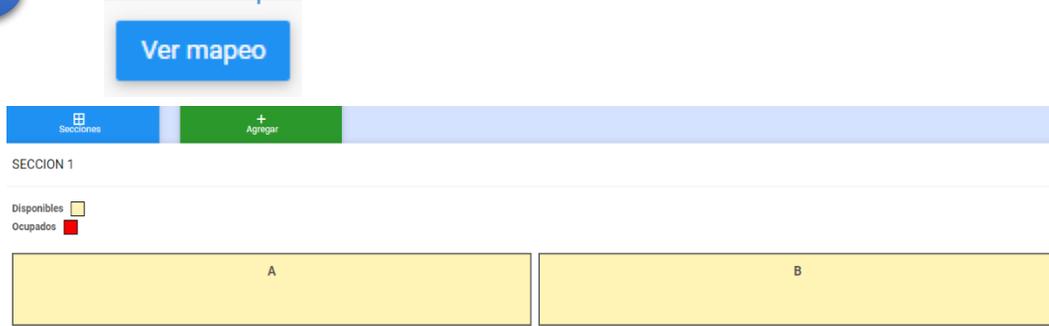
“Icono de Eliminar”



**Descripción:** Es el botón o icono eliminar, donde se eliminar un registro ya registrado.

6

“Icono de Mapeo”



**Descripción:** Es el botón o icono de mapeo, dirige a un formulario donde se muestra los espacios disponibles, de igual manera se pueden crear, modificar los nombres y eliminar espacios.

## 10. PANTALLA DE MÓDULO DE COBROS

### FORMULARIO DE COBROS



**Descripción:** Es el formulario de cobros.

1 “Lista de cobros”



**Descripción:** Es el formulario donde se muestra la lista de los cobros de salidas de vehículos del parqueo, registrados en el sistema, la acción.

2 “Buscador

Buscar:

**Descripción:** Es para realizar búsquedas de los cobros registrados.

4 “Icono de Vista”



**Descripción:** Es el botón o icono de vista, genera la factura de prueba para realizar el cobro.

5

“Icono de Eliminar”



**Descripción:** Es el botón o Icono de eliminar, puede eliminar la factura de prueba.

## 11. PANTALLA DE MODULO DE INGRESOS Y SALIDAS

### FORMULARIO DE INGRESOS Y SALIDAS

Mostrar 10 registros

Código	Fecha	Hora	Vehículo	Acción	Observación	Acción
10	2021-06-06	20:28:00	CAFE	SALIDA		
9	2021-06-06	20:15:00	NEGRO	SALIDA		

**Descripción:** Es el formulario de ingresos y salidas.

1

“Lista de ingreso y salidas”

Mostrar 10 registros

Código	Fecha	Hora	Vehículo	Acción	Observación	Acción
10	2021-06-06	20:28:00	CAFE	SALIDA		

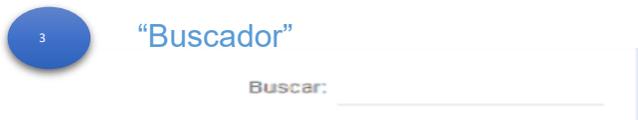
**Descripción:** Es el formulario donde se muestra la lista de los ingresos y salidas de vehículos del parqueo, se registra de forma automática usando la tarjeta rfid, la acción.

2

“Botón para adicionar un nuevo ingreso y salida”



**Descripción:** Es el botón para adicionar o crear un nuevo registro de ingresos y salidas donde debe llenar se llenan automáticamente usando la tecnología rfid.



**Descripción:** Es para realizar búsquedas de los ingresos y salidas registrado.



**Descripción:** Es el botón o Icono de vista, generala salida y el ingreso de vehículo con la hora y fecha.



**Descripción:** Es el botón o Icono de eliminar registro.

## 12. PANTALLA DE REPORTE

### FORMULARIO DE REPORTE



**Descripción:** es el formulario donde están todos los reportes y puede generar el que desee.

1

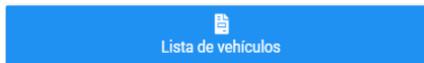
“botón de lista de usuarios”



**Descripción:** genera el reporte de lista de usuarios.

2

“botón de lista vehículos”



**Descripción:** genera el reporte de lista de los vehículos registrados en el sistema.

3

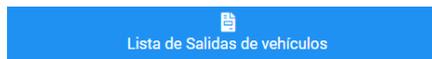
“botón de lista de ingresos de vehículos”



**Descripción:** genera el reporte de lista de los vehículos registrados que ingresaron al parqueo, donde se genera por categoría, día, fecha y tarifa.

4

“botón de lista de salidas de vehículos”



**Descripción:** genera el reporte de lista de los vehículos registrados de salidas al parqueo, donde se genera por categoría, día, fecha y tarifa.

5

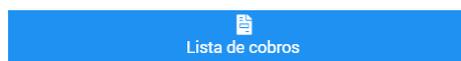
“botón de lista de tarifas”



**Descripción:** genera el reporte de lista de tarifas.

6

“botón de lista de cobros”



**Descripción:** genera el reporte del total que genero de los cobros.

7

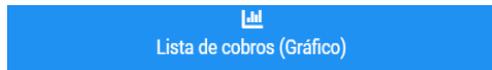
“botón de lista de ubicaciones disponibles”



**Descripción:** genera el reporte de lista de espacios disponibles en el mapeo.

8

“estadística del total de los cobros”



**Descripción:** genera la estadística de los gráficos de total de cobros por día, fecha y genera un reporte.

9

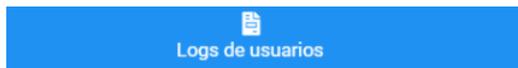
“gráficos de espacios disponibles en el mapeo”



**Descripción:** genera el grafico de total de espacios disponibles en el mapeo por secciones.

10

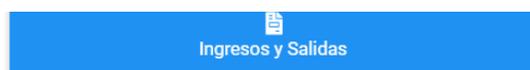
“logs usuarios”



**Descripción:** genera el reporte de seguimiento del usuario, de que realizo en el sistema.

11

“botón de lista de ingresos y salidas de vehículos”



**Descripción:** genera el reporte de lista de ingreso y entrada de vehículos.

### 13. PANTALLA DE LA VISUALIZACION PARA EL CLIENTE

#### CONTROL EN PANTALLA



**Descripción:** Es donde muestra al cliente los datos que desea.

### 14. PANTALLA DE DATOS DE LA EMPRESA

#### FORMULARIO DE DATOS DE LA EMPRESA

EMPRESA

Mostrar 10 registros Buscar: \_\_\_\_\_

Código	Nombre	Nit	Nro. autorización	Teléfono-Celular	Logo	Acción
0	PARQUEO GONZALES	0	0	S/N S/N		
Código	Nombre	Nit	Nro. autorización	Teléfono-Celular	Logo	Acción

Mostrando 1 a 1 de 1 registros Anterior **1** Siguiente

**Descripción:** Es donde puede cambiar los datos de su parqueo, para que no afecte los cambios en el sistema.

ANEXO C. MANUAL TECNICO

# MANUAL TECNICO

## SISTEMA DE CONTROL Y SEGUIMIENTO DE PARQUEO APLICANDO LA TECNOLOGIA RFID



## INDICE DE MANUAL TECNICO

1. OBJETIVOS.....	1
2. REQUERIMIENTOS MÍNIMOS DE HARDWARE:.....	1
3. REQUERIMIENTOS MÍNIMOS DE SOFTWARE .....	1
4. INSTACION DE MANERA LOCAL .....	1
4.1. INSTALAR XAMPP .....	1
5. SERVIDOR .....	9
5.1. INSTALACION DEL SISTEMA .....	9

## 1. OBJETIVOS

- ✓ Dar a conocer toda la información necesaria a los administradores que llevaran a cabo la instalación y control del sistema.
- ✓ Representar la estructura técnica y diseño de la plataforma.

## 2. REQUERIMIENTOS MÍNIMOS DE HARDWARE:

- ✓ Procesador: Core
- ✓ Memoria RAM: mínimo: 1 Gigabytes (GB)
- ✓ Disco Duro: 500Gb.

## 3. REQUERIMIENTOS MÍNIMOS DE SOFTWARE

- ✓ Privilegios de administrador
- ✓ Sistema Operativo: Windows 7/ 8 /8.1 /10

## 4. INSTACION DE MANERA LOCAL

### 4.1. INSTALAR XAMPP

Antes de instalar un servidor de páginas web es conveniente comprobar sino existe uno ya instalado, o al menos si no está en funcionamiento.

#### Paso 1: Descargar

Las versiones con PHP 5.5, 5.6 O 7 se puede descargar gratuitamente desde la página del proyecto [Apache Friends](#).

#### Paso 2: Ejecutar el archivo.exe

Una vez descargado el paquete, puedes ejecutar el archivo.exe haciendo clic en él.



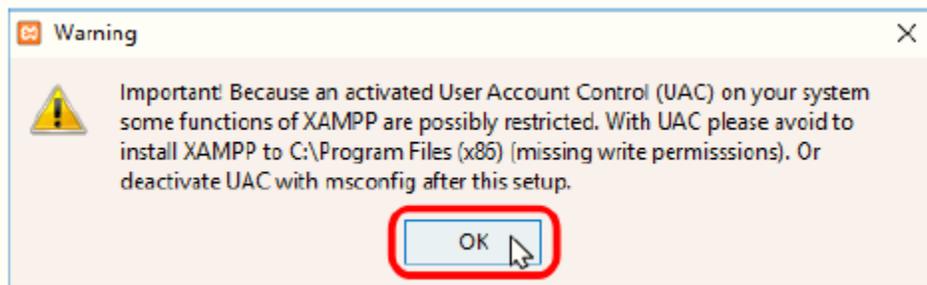
xampp-windows-x64-7.2.24-0-VC15-installer

### Paso 3: Desactivar el programa antivirus

Se recomienda desactivar el programa antivirus hasta que todos los componentes estén instalados, ya que puede obstaculizar el proceso de instalación.

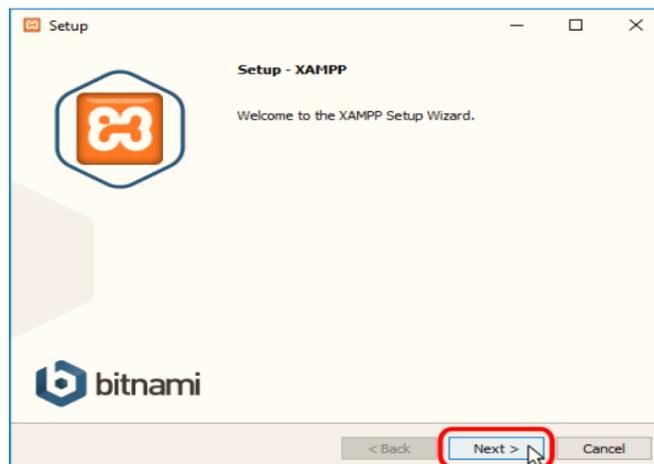
### Paso 4: Desactivar el programa antivirus

También el control de cuentas de usuario (User Account Control, UAC) pueden interferir en la instalación, ya que limita los derechos de escritura en la unidad de disco C:\. Una vez obtenido el archivo de instalación de XAMPP, hay que hacer doble clic sobre él para ponerlo en marcha. Al poner en marcha el instalador XAMPP nos muestra un aviso que aparece si está activado el Control de Cuentas de Usuario y recuerda que algunos directorios tienen permisos restringidos:



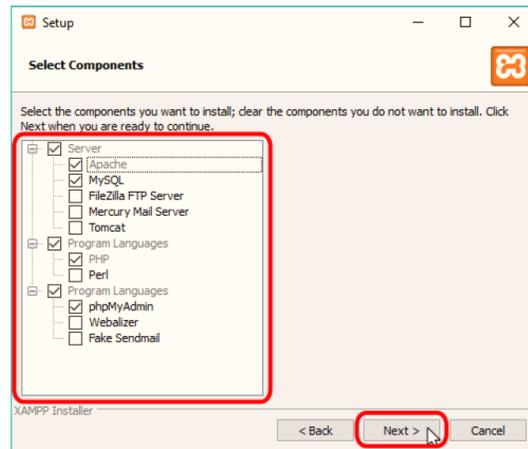
### Paso 5: Iniciar el asistente de instalación

Una vez superados estos pasos, aparece la pantalla de inicio del asistente para instalar XAMPP. Para ajustar las configuraciones de la instalación se hace clic en "Next".



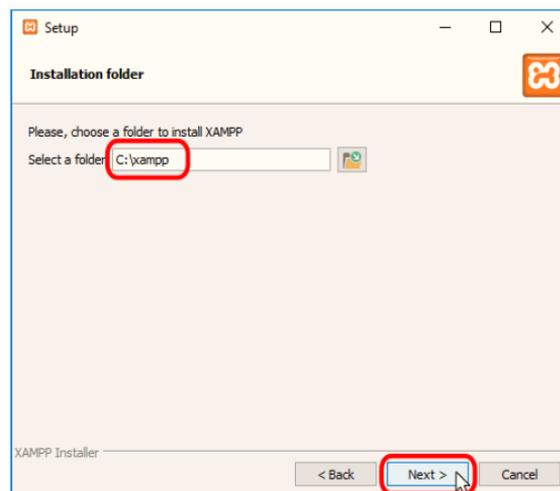
## Paso 6: Selección de los componentes del software

Los componentes mínimos que instala XAMPP son el servidor Apache y el lenguaje PHP, pero XAMPP también instala otros elementos. En la pantalla de selección de componentes puede elegir la instalación o no de estos componentes. Para seguir estos apuntes se necesita al menos instalar MySQL y phpMyAdmin. Confirma la selección haciendo clic en "Next".



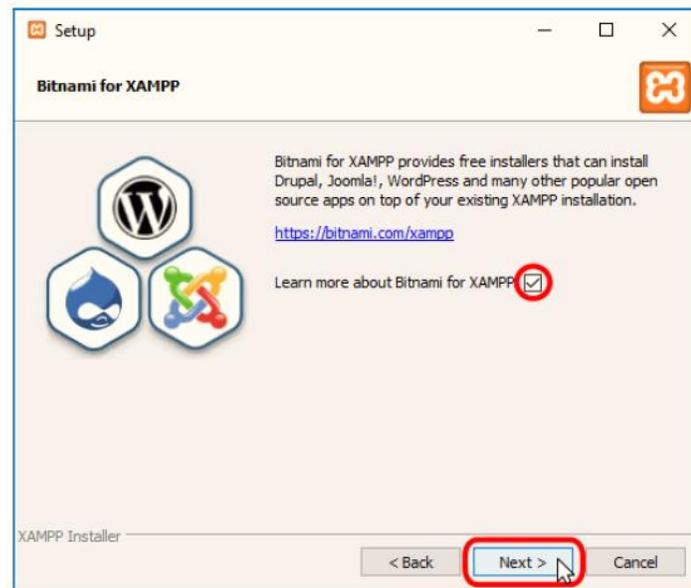
## Paso 7: Selección del directorio para la instalación

En la siguiente pantalla puede elegir la carpeta de instalación de XAMPP. La carpeta de instalación predeterminada es C:\xampp. Si quiere cambiarla, haga clic en el icono de carpeta y seleccione la carpeta donde quiere instalar XAMPP. Para continuar la configuración de la instalación, haga clic en el botón "Next".

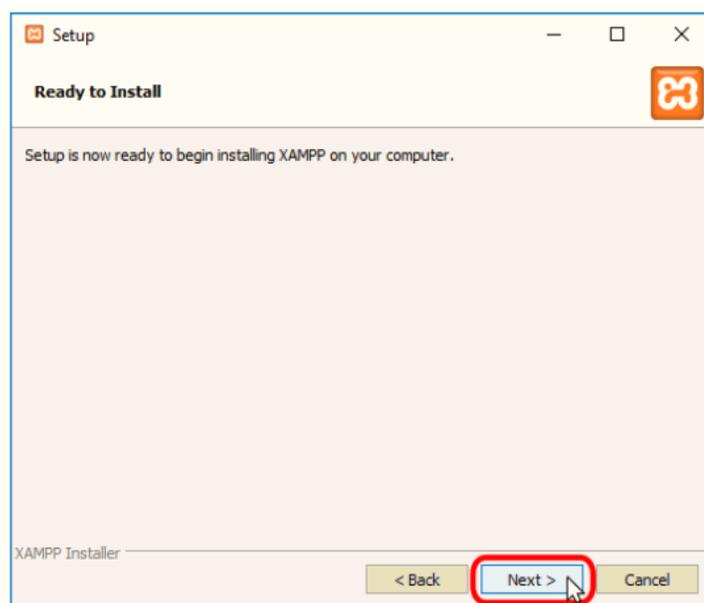


## Paso 7: Iniciar el proceso de instalación

La siguiente pantalla ofrece información sobre los instaladores de aplicaciones para XAMPP creados por Bitnami. Haga clic en el botón "Next" para continuar. Si deja marcada la casilla, se abrirá una página web de Bitnami en el navegador.



Una vez elegidas las opciones de instalación en las pantallas anteriores, esta pantalla es la pantalla de confirmación de la instalación. Haga clic en el botón "Next" para comenzar la instalación en el disco duro.

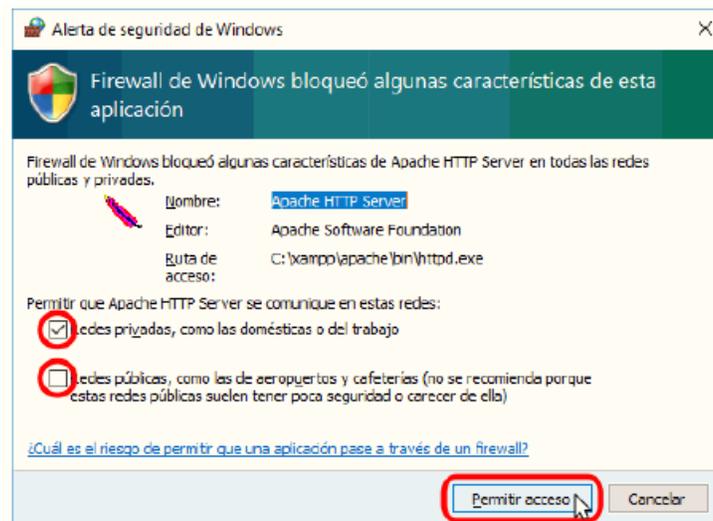


El proceso de copia de archivos puede durar unos minutos.



## Paso 9: configurar Firewall

Durante la instalación, si en el ordenador no se había instalado Apache anteriormente, en algún momento se mostrará un aviso del cortafuegos de Windows para autorizar a Apache a comunicarse en las redes privadas o públicas. Una vez elegidas las opciones deseadas (en estos apuntes se recomienda permitir las redes privadas y denegar las redes públicas), haga clic en el botón "Permitir acceso".



## Paso 10: Cerrar la instalación

Una vez extraídos e instalados todos los componentes puedes cerrar el asistente con la tecla “Finish”. Para acceder inmediatamente al panel de control solo es necesario marcar la casilla que pregunta si deseamos hacerlo.



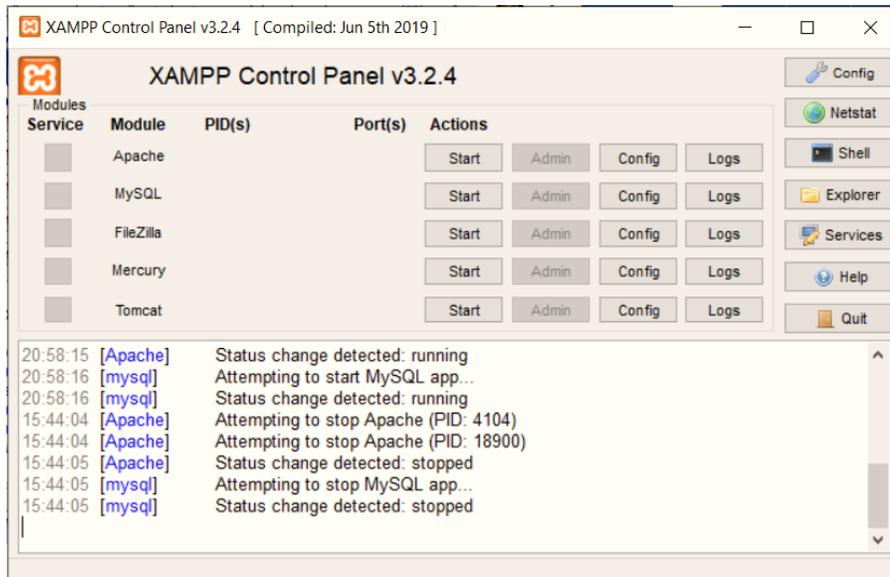
Parece ser que el instalador de XAMPP tiene un problema en Windows 10 y no crea la carpeta del menú inicio. (Referencias: foro de xampp).

## Panel de control de XAMPP

En la diáfana **interfaz de usuario** del panel de control se protocolan todas las acciones y es posible activar o desactivar los módulos por separado con un simple clic. Además, se dispone de diversas utilidades como:

- ✓ **Config:** para configurar XAMPP, así como otros componentes aislados.
- ✓ **Netstat:** muestra todos los procesos en funcionamiento en el ordenador local
- ✓ **Shell:** lanza una ventana de comandos UNIX
- ✓ **Explorer:** abre la carpeta XAMPP en el explorador de Windows
- ✓ **Services:** muestra todos los servicios en funcionamiento

- ✓ **Help:** incluye enlaces a foros de usuarios
- ✓ **Quit:** se usar para salir del panel de control



## Iniciar Modulo

En la parte superior se pueden iniciar o interrumpir los módulos de XAMPP por separado mediante los comandos “Start” y “Stop” bajo “Actions”. Los módulos que se activaron aparecen marcados en verde.



## SUBLIME TEXT

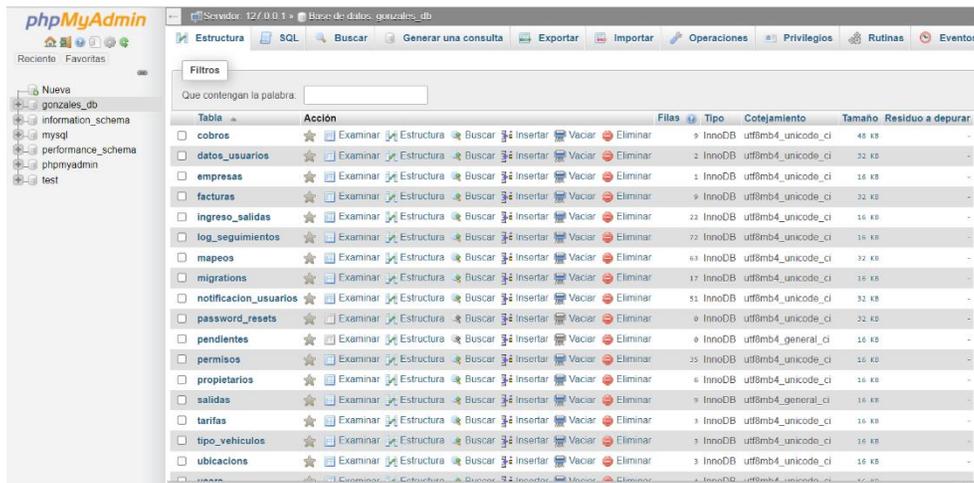


Es una herramienta concebida para programar sin distracciones. El sistema de resaltado de sintaxis de Sublime Text soporta un gran número de lenguajes y es un editor de código multiplataforma.

## CONFIGURACION DE BASE DE DATOS

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=gonzales_db
DB_USERNAME=root
DB_PASSWORD=
```

## CARGAR LA BASE DE DATOS A MySQL



## 5. SERVIDOR

Para el funcionamiento más eficiente del sistema de control y seguimiento de parqueo aplicando la tecnología RFID, se requiere instalar el software sobre un hosting, preferentemente una de paga, con las siguientes características.

- Disco duro de 40Gb (almacenamiento).
- Memoria ram 4Gb
- Tener instalado el sistema gestor de bases de datos María DB o MySQL

### 5.1. INSTALACION DEL SISTEMA

#### ✓ Configuración MySQL para producción

Para iniciar la configuración de MySQL es:

```
mysql_secure_installation
```

#### ✓ Instalación de phpMyAdmin

*Instalar phpMyAdmin es un paso opcional*

Entonces, para contar con phpMyAdmin, ejecuta el siguiente comando para iniciar con el proceso de instalación:

```
sudo apt-get install phpmyadmin
```

Una vez instalado editamos el archivo de configuración de Apache:

```
sudo nano /etc/apache2/apache2.conf
```

Y añadimos esta línea al final de todo:

```
Include /etc/phpmyadmin/apache.conf
```

Con ello finalizamos la instalación, y debemos reiniciar Apache antes de continuar:

✓ **Configuración específica para laravel**

Lo primero que haremos será instalar Composer:

```
curl -sS https://getcomposer.org/installer | php
sudo mv composer.phar /usr/local/bin/composer
```

✓ **Y luego Git, para que luego podamos hacer deploy a través de un repositorio.**

```
sudo apt-get install git
```

Para clonar el proyecto laravel nos ubicamos en /var/www:

```
root@lamp-512mb-incidencias:/# cd ..
root@lamp-512mb-incidencias:/# cd ..
root@lamp-512mb-incidencias:/# ls
bin      etc          lib          media  proc  sbin  sys  var
boot    home        lib64       mnt    root  snap  tmp  vmlinuz
dev     initrd.img  lost+found  opt    run   srv   usr
root@lamp-512mb-incidencias:/# cd var/www
```

Y clonamos el repositorio:

<https://github.com/jaeruthq/gonzales.git>

Como mi proyecto se llama *gonzales*, voy a verificar que esta carpeta se haya creado correctamente ingresando con `cd` y luego consultando su contenido con `ls`.

Así como lo muestra la imagen:

```
root@lamp-512mb-incidencias:/var/www# git clone https://github.com/JCarlosR/gestion-incidencias.git
Cloning into 'gestion-incidencias'...
remote: Counting objects: 323, done.
remote: Compressing objects: 100% (179/179), done.
remote: Total 323 (delta 126), reused 323 (delta 126), pack-reused 0
Receiving objects: 100% (323/323), 117.10 KiB | 0 bytes/s, done.
Resolving deltas: 100% (126/126), done.
Checking connectivity... done.
root@lamp-512mb-incidencias:/var/www# ls
gestion-incidencias  html
root@lamp-512mb-incidencias:/var/www# cd gestion-incidencias
root@lamp-512mb-incidencias:/var/www/gestion-incidencias# ls
app                composer.lock     levels-how-it-works.php  readme.md  storage
artisan            config            package.json             resources  tests
bootstrap          database          phpunit.xml              routes
composer.json     gulpfile.js      public                   server.php
```

Nuestra carpeta se ha creado. Pero recuerda:

Un proyecto Laravel tiene dependencias sobre otros paquetes. Estas dependencias se encuentran declaradas en el archivo **composer.json**.

Entonces para instalar lo necesario ejecutamos el siguiente comando:

```
composer install
```

La descarga de las dependencias probablemente tarde un par de minutos, dependiendo del proyecto.

```
sudo apt-get install php7.4-curl php7.4-gd php7.4-json php7.4-mbstring php7.4-intl php7.4-mysql
```

El comando **composer install** creará una carpeta **vendor** dentro de la carpeta de tu proyecto Laravel, con las dependencias que han sido descargadas.

Una vez que haya finalizado esta descarga de dependencias, debemos dar permisos a la carpeta **storage**.

Para ello ejecutamos:

```
sudo chown -R www-data: storage
sudo chmod -R 755 storage
```

### ✓ Apuntar a la carpeta/public

Lo que nos falta es que nuestro host reconozca a la carpeta public como la ruta base de nuestro proyecto.

Para ello tenemos que editar un archivo de configuración de Apache con el siguiente comando:

```
nano /etc/apache2/sites-enabled/000-default.conf
```

Aquí tenemos que reemplazar estas líneas: Lo reemplazamos con el gonzales

```
DocumentRoot /var/www/html
```

```
<Directory /var/www/html/>
```

Por estas 2, respectivamente:

```
<Directory /var/www/gonzales/public/>
```

```
<Directory /var/www/gonzales/public/>
```

Y agregar estas 2 líneas dentro de la etiqueta Directory: gonzales

```
RewriteEngine On
```

```
RewriteBase /var/www/gonzales/public
```

Guardamos los cambios en el archivo, y reiniciamos Apache una última vez:

```
sudo service apache2 restart
```

### ✓ **Creando nuestro archivo de configuración. env**

Nuestro servidor ya está configurado, pero recordemos que todo proyecto Laravel se basa en un archivo. *env* para guardar allí las credenciales que usa nuestro sistema.

Este archivo. *env* forma parte de nuestro. *gitignore* por lo que (como se espera) no fue clonado y tenemos que crearlo.

Para ello tenemos 2 alternativas:

- Copiar el archivo de ejemplo que viene con Laravel y reemplazar allí las credenciales.

- Crear un archivo nuevo vacío y pegar allí el contenido de nuestro archivo. `env local`.

A fin de cuentas, el resultado es el mismo.

Si optas por la alternativa (1) puedes usar los siguientes 2 comandos. Si optas por la alternativa (2) sólo usa el 2do y el archivo estará vacío.

```
cp .env.example .env
```

```
nano .env
```

### ✓ **Base de datos, migraciones y seeders**

En mi archivo. `env` la variable `DB_DATABASE` tiene como valor `incidencias`. Eso significa que ese es el nombre de la base de datos que usa mi proyecto.

Lo que haré entonces será crear esta base de datos.

Tú debes hacer lo mismo:

Crear la base de datos que tu proyecto requiere.

- Lo puedes hacer desde phpMyAdmin,
- o desde la consola misma (`CREATE DATABASE incidencias;`).

Esta base de datos estará totalmente vacía, pero las migraciones harán magia y crearán las tablas:

```
php artisan migrate
```

Si tus seeders solo contienen datos de prueba que usas de forma local, no será necesario ejecutarlos aquí.

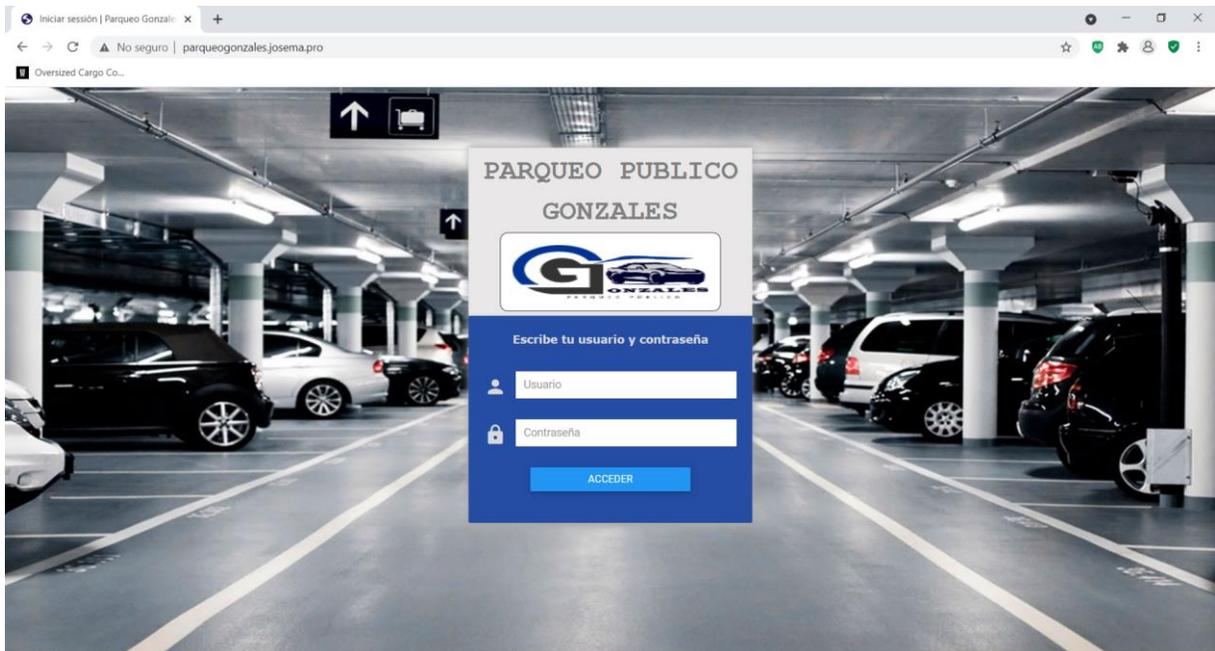
Pero si tus seeders agregan datos iniciales importantes para el funcionamiento de tu aplicación, entonces puedes ejecutarlos con:

```
php artisan db:seed
```

O en su defecto, usar el siguiente comando (equivalente a los 2 anteriores).

```
php artisan migrate --seed
```

una vez realizado los pasos de la compilación el proyecto, dirigirse al navegador de la siguiente dirección <http://parqueogonzales.josema.pro/> donde el sistema estará cargando.



# **ANEXO D.**

# **DOCUMENTACION**

