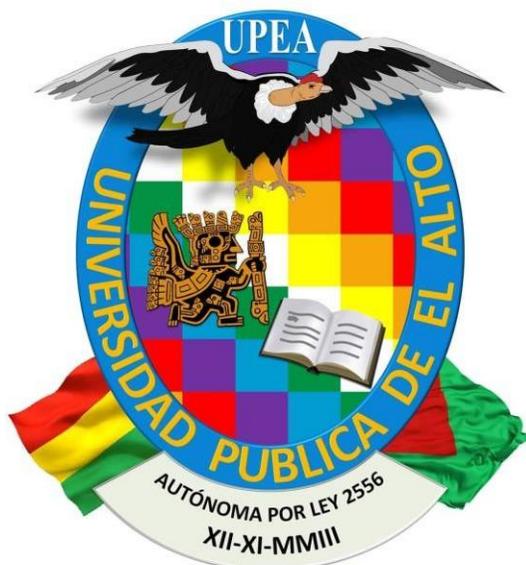


# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



## PROYECTO DE GRADO

**“SISTEMA WEB DE INFORMACIÓN GEOGRÁFICA ORIENTADA AL  
MARKETING PARA LAS EMPRESAS EN CONVENIO”**

**CASO: FACEBOL S.R.L.**

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

**Postulante: Univ. Janneth Verónica Ambrocio Mamani.**

**Tutor Metodológico: Ing. Enrique Flores Baltazar.**

**Tutor Especialista: Ing. Ramiro Kantuta Limachi.**

**Tutor Revisor: Ing. Rolman Cordero Mamani.**

**EL ALTO - BOLIVIA**

**2022**

**DECLARACIÓN JURADA DE  
AUTENTICIDAD Y RESPONSABILIDAD**

Yo, **Janneth Verónica Ambrocio Mamani** estudiante con **C.I. 7322279 OR.** mediante la presente **declaro** de manera pública que la propuesta del **TRABAJO DE GRADO** titulada “**SISTEMA WEB DE INFORMACIÓN GEOGRÁFICA ORIENTADA AL MARKETING PARA LAS EMPRESAS EN CONVENIO**” es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados,

Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el **TRABAJO DE GRADO** haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, mayo del 2022.

**Janneth Verónica Ambrocio Mamani**  
**C.I. 7322279 OR.**  
**e-mail: veroambrociomamani@gmail.com**

## **DEDICATORIA**

Dedico el presente proyecto a mi querida y prestigiosa Universidad Pública de El Alto, por permitirme formar parte de una gran familia donde conocí grandes personas y sobre todo me formé profesionalmente para servir a mi país.

A mis queridos padres por ser el pilar fundamental en todo lo que soy, en toda mi educación, como de la vida, por su incondicional apoyo en cada momento.

A mis queridas hermanas, que me alentaron a seguir adelante y no rendirme a pesar de los problemas.

A mis estimados tutores por guiarme e inculcarme sus conocimientos para el desarrollo y culminación del mismo, muchas gracias.

## **AGRADECIMIENTOS**

Agradezco a Dios por darme fuerzas y cuidarme en todo el proceso de formación profesional, a mi familia por todo su apoyo incondicional y por ser mi más grande fortaleza, a mis tutores por todo su apoyo y palabras de aliento que me dieron ánimos para seguir adelante y a mis amigos por todo su apoyo y consejos.

# ÍNDICE GENERAL

<b>MARCO PRELIMINAR</b> .....	1
<b>1.1. INTRODUCCIÓN</b> .....	1
<b>1.2. ANTECEDENTES</b> .....	2
1.2.1. Antecedente Institucional .....	2
1.2.2. Antecedentes a fines del proyecto de grado.....	2
<b>1.3. PLANTEAMIENTO DEL PROBLEMA</b> .....	4
1.3.1. Problema Principal .....	4
1.3.2. Problemas Secundarios .....	5
1.3.3. Formulación del Problema .....	5
<b>1.4. OBJETIVOS</b> .....	5
1.4.1. Objetivo General.....	5
1.4.2. Objetivos Específicos .....	5
<b>1.5. JUSTIFICACIÓN</b> .....	6
1.5.1. Justificación Técnica .....	6
1.5.2. Justificación Económica.....	6
1.5.3. Justificación Social .....	6
<b>1.6. METODOLOGÍA</b> .....	7
1.6.1. Metodología XP .....	7
1.6.2. Metodología Kanban .....	8
1.6.3. Métrica de Calidad del Software .....	8
1.6.4. Estimación de Costos.....	9
1.6.5. Seguridad .....	9
1.6.6. Pruebas al Software .....	10
<b>1.7. HERRAMIENTAS</b> .....	11
1.7.1. Hardware .....	11
1.7.2. Software.....	11
<b>1.8. LÍMITES Y ALCANCES</b> .....	15
1.8.1. Limites.....	15
1.8.2. Alcances .....	15
<b>1.9. APORTES</b> .....	16
<b>MARCO TEORICO</b> .....	17

<b>2.1. INTRODUCCIÓN</b> .....	17
<b>2.2. SISTEMA</b> .....	17
<b>2.3. INFORMACIÓN</b> .....	17
<b>2.4. SISTEMA WEB</b> .....	18
<b>2.5. SISTEMA DE INFORMACIÓN GEOGRÁFICA</b> .....	18
<b>2.5.1. Definición</b> .....	18
<b>2.5.2. Componentes de los Sistemas de Información geográfica</b> .....	19
<b>2.5.3. Funciones de un SIG</b> .....	22
<b>2.5.4. Funciones de los Sistemas de Información Geográfica</b> .....	23
<b>2.6. MARKETING</b> .....	24
<b>2.6.1. ¿Qué es Marketing Digital?</b> .....	25
<b>2.6.2. Características del Marketing Digital</b> .....	25
<b>2.7. INGENIERÍA DE SOFTWARE</b> .....	27
<b>2.7.1. ¿Qué es la ingeniería de Software?</b> .....	27
<b>2.7.2. ¿Por qué es importante la ingeniería de Software?</b> .....	27
<b>2.7.3. Procesos del Software</b> .....	28
<b>2.7.4. Modelo de Proceso de Software</b> .....	29
<b>2.8. METODOLOGÍA</b> .....	32
<b>2.8.1. Metodología de Programación Extrema XP</b> .....	32
<b>2.8.2. Metodología Kanban</b> .....	41
<b>2.9. MÉTRICAS DE CALIDAD DEL SOFTWARE</b> .....	44
<b>2.9.1. Factores de calidad ISO 25000</b> .....	44
<b>2.9.2. Implementación de la ISO/IEC 25010</b> .....	45
<b>2.10. ESTIMACIÓN DE COSTOS</b> .....	49
<b>2.10.1. COCOMO II</b> .....	49
<b>2.10.2. Método de Estimación de Costos Nivel Intermedio</b> .....	50
<b>2.10.3. Características de COCOMO</b> .....	51
<b>2.10.4. Fórmulas de Estimación</b> .....	53
<b>2.11. SEGURIDAD</b> .....	54
<b>2.11.1. Técnicas de seguridad</b> .....	54
<b>2.12. PRUEBAS AL SOFTWARE</b> .....	55
<b>2.12.1. Proceso de Pruebas</b> .....	56
<b>2.12.2. Tipos de prueba a usar</b> .....	57

2.13. HERRAMIENTAS .....	59
2.13.1. Hardware .....	59
2.13.2. Software.....	60
2.13.3. Framework.....	62
2.13.4. Base de datos.....	63
2.13.5. Api de Google Maps .....	63
2.13.6. Servicios Api Rest (GET POST PUT DELETE).....	64
<b>MARCO APLICATIVO .....</b>	<b>69</b>
<b>3.1. INTRODUCCIÓN .....</b>	<b>69</b>
<b>3.2. ESTRUCTURA O ESQUEMA DEL SISTEMA.....</b>	<b>69</b>
3.2.1. Funcionalidad del sistema.....	70
<b>3.3. APLICACIÓN DE LA METODOLOGÍA KANBAN .....</b>	<b>70</b>
3.3.1. Implementación del Tablero Kanban.....	70
3.3.2. Roles Kanban .....	71
<b>3.4. APLICACIÓN DE LA METODOLOGÍA XP .....</b>	<b>76</b>
3.4.1. Fase de Iteraciones.....	76
3.4.2. Fase de Diseño.....	81
3.4.3. Fase de Implementación .....	86
3.4.4. Fase de Pruebas.....	106
<b>METRICAS DE CALIDAD, COSTO Y SEGURIDAD .....</b>	<b>112</b>
<b>4.1. INTRODUCCIÓN .....</b>	<b>112</b>
<b>4.2. MÉTRICA DE CALIDAD .....</b>	<b>112</b>
4.2.1. Factores de calidad ISO/IEC 25000 .....	112
<b>4.3. ANALISIS DE COSTOS.....</b>	<b>120</b>
4.3.1. Aplicación del COCOMO II Nivel Intermedio .....	120
4.3.2. Costos del Hardware .....	122
<b>4.4. SEGURIDAD .....</b>	<b>123</b>
4.4.1. Seguridad Web .....	123
4.4.2. Seguridad de password con Bcrypt.....	124
4.4.3. Seguridad con Cifrado de datos con Laravel .....	124
<b>4.5. PRUEBAS AL SOFTWARE.....</b>	<b>125</b>
4.5.1. Prueba de Caja Blanca .....	125
4.5.2. Prueba de Caja Negra .....	131

4.5.3. Prueba de Estrés .....	132
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>135</b>
5.1. CONCLUSIONES Y RECOMENDACIONES.....	135
5.1.1. CONCLUSIONES.....	135
5.1.2. RECOMENDACIONES .....	135
<b>REFERENCIAS BIBLIOGRAFICAS .....</b>	<b>136</b>
<b>ANEXOS .....</b>	<b>138</b>
ANEXO A. ARBOL DE PROBLEMAS .....	138
ANEXO C.....	140
ANEXO D. ....	141

## ÍNDICE DE TABLAS

Tabla 2.1. Valores constantes COCOMO Intermedio.....	51
Tabla 2.2. Factores de costos .....	52
Tabla 2.3. Estimación de Costos .....	54
Tabla 3.1. Fases de trabajo, limite WIP y responsables .....	71
Tabla 3.2. Historias de usuario .....	74
Tabla 3.3. Estimación de tiempos de la Iteración 1.....	76
Tabla 3.4. Estimación de tiempos de la Iteración 2.....	79
Tabla 3.5. Prueba de Aceptación Iteración 1.....	106
Tabla 3.6. Prueba de Aceptación Iteración 2.....	109
Tabla 4.1. Parámetros de Medida y su Cantidad.....	113
Tabla 4.2. Parámetros del punto de función.....	114
Tabla 4.3. Valores de fiabilidad de cada modulo .....	116
Tabla 4.4. Parámetros de medición .....	117
Tabla 4.5. Información requerida por el IMS.....	118
Tabla 4.6. Factor de portabilidad .....	119
Tabla 4.7. Resultados .....	120
Tabla 4.8. Factores de Costos.....	120
Tabla 4.9. Tabla general de costos .....	123
Tabla 4.10. Costo total del sistema .....	123
Tabla 4.11. Nivel de acceso al sistema de parte de los usuarios .....	125
Tabla 4.12. Casos de prueba.....	127
Tabla 4.13. Casos de prueba.....	129
Tabla 4.14. Casos de prueba.....	131
Tabla 4.15. Caso de Prueba al Módulo Usuarios .....	131
Tabla 4.16. Caso de Prueba al Módulo Empresas .....	132
Tabla 4.17. Caso de Prueba al Módulo Reportes .....	132
Tabla 4.18. Resultados .....	133
Tabla 4.19. Resultados .....	134
Tabla 4.20. Resultados .....	135

## ÍNDICE DE FIGURAS

Figura 2.1. Componentes que forman un SIG .....	19
Figura 2.2. Características del Marketing Digital. ....	26
Figura 2.3. Las 4F del Marketing Digital.....	26
Figura 2.4. Modelo en Cascada.....	30
Figura 2.5. Desarrollo incremental.....	31
Figura 2.6. Ingeniería de Software orientada a la reutilización.....	31
Figura 2.7. Fases de la Metodología XP .....	41
Figura 2.8. Panel de Kanban .....	43
Figura 2.9. Características de la calidad del producto de software .....	45
Figura 2.10. El Proceso de Pruebas.....	57
Figura 2.11. Ejemplo de caja negra.....	58
Figura 2.12. Ejemplo de caja negra.....	59
Figura 3.1. Arquitectura del sistema .....	69
Figura 3.2. Diagrama de procesos o diagrama de actividades .....	70
Figura 3.3. Muro Kanban .....	71
Figura 3.4. Diagrama de caso de uso general del sistema .....	81
Figura 3.5. Caso de uso gestión de Usuario .....	81
Figura 3.6. Caso de uso de gestión de configuración.....	82
Figura 3.7. Caso de uso de gestión de empresas .....	82
Figura 3.8. Caso de uso de gestión de suscripción.....	83
Figura 3.9. Caso de uso de gestión de afiliación.....	83
Figura 3.10. Caso de uso de gestión de pagos.....	84
Figura 3.11. Caso de uso de gestión de reportes .....	84
Figura 3.12. Diagrama físico de la Base de Datos del sistema .....	85
Figura 3.13. Diagrama de Base de Datos .....	86
Figura 3.14. Registrar usuarios al sistema.....	87
Figura 3.15. Código referente a la vista registro de usuarios .....	87
Figura 3.16. Asignar roles a los usuarios .....	88
Figura 3.17. Código referente a la vista asignación de roles.....	88
Figura 3.18. Inicio y cierre de sesiones .....	89
Figura 3.19. Código referente a la vsta inicio de sesión.....	89
Figura 3.20. Administrar rubros.....	90
Figura 3.21. Código referente a la vista rubros .....	90

Figura 3.22. Administrar planes.....	91
Figura 3.23. Código referente a la vista planes .....	91
Figura 3.24. Registrar empresas.....	92
Figura 3.25. Código referente a la vista empresas .....	92
Figura 3.26. Registrar ubicación de empresas.....	93
Figura 3.27. Código referente a la vista ubicación de empresas .....	93
Figura 3.28. Registrar servicios de empresa.....	94
Figura 3.29. Código referente a la vista servicios .....	94
Figura 3.30. Registrar suscripción de la empresa.....	95
Figura 3.31. Código referente a la vista suscripción .....	95
Figura 3.32. Designar plan .....	96
Figura 3.33. Código referente a la vista de planes .....	96
Figura 3.34. Generar boleta de suscripción.....	97
Figura 3.35. Código referente a la vista de suscripciones .....	97
Figura 3.36. Registrar datos de los clientes.....	98
Figura 3.37. Código referente a la vista de clientes .....	98
Figura 3.38. Registrar afiliación.....	99
Figura 3.39. Código referente a la vista afiliaciones .....	99
Figura 3.40. Asignar tarjeta de afiliación.....	100
Figura 3.41. Código referente a la vista tarjetas.....	100
Figura 3.42. Registrar pago de afiliación .....	101
Figura 3.43. Código referente a la vista pagos .....	101
Figura 3.44. Generar comprobante de pago .....	102
Figura 3.45. Código referente a la vista comprobante de pago .....	102
Figura 3.46. Generar reporte de empresas.....	103
Figura 3.47. Código referente a la vista reporte de empresas .....	103
Figura 3.48. Generar reporte de empresas suscritas.....	104
Figura 3.49. Código referente a la vista reporte de suscripciones.....	104
Figura 3.50. Generar reporte de afiliaciones .....	105
Figura 3.51. Código referente a la vista reporte de afiliaciones .....	105
Figura 4.1. Grafo de Flujo del Módulo Usuario.....	126
Figura 4.2. Grafo de flujo del Módulo Empresas.....	128
Figura 4.3. Grafo de flujo del Módulo Reportes .....	130
Figura 4.4. Prueba al Módulo Empresa.....	133
Figura 4.5. Prueba al Módulo de afiliación .....	134

Figura 4.6. Prueba al Módulo de Suscripciones ..... 135

## RESUMEN

El presente proyecto de grado hace referencia a los Sistemas de Información Geográfica "SIG", la cual está transformando muchos campos de la actividad humana, acompañadas de las tecnologías de la información. El mismo coadyuvará en gran magnitud para la gestión de la información, así también brinda el acceso directo e interactivo entre las tareas y funciones que se realizaban manualmente, ahora son realizadas por medio de un sistema web de información.

En el primer capítulo del Marco Preliminar, se realiza el análisis de la problemática junto a los antecedentes de la empresa, y a partir de ello se define los objetivos a desarrollar para dar solución a la problemática planteada. Así mismo mediante toda la información obtenida se describe la justificación social, técnica y económica. Y por último se define las herramientas que se utilizará, además de los límites y alcances del presente proyecto.

En el segundo capítulo del Marco Teórico, se define a toda la parte teórica la cual se establecen en marco anterior, sobre la cual se basa el presente proyecto.

En el tercer capítulo que es el Marco Aplicativo, se muestra el análisis y desarrollo del sistema con el apoyo de la metodología XP y Kanban para la planificación en el proceso de desarrollo del sistema.

El cuarto capítulo, que es Métricas de Calidad, Costo y Seguridad, se desarrolla las características para la calidad del Software con la norma ISO/IEC 25010, para la estimación de costos se utilizó COCOMO II y se describe la seguridad en diferentes partes del sistema.

Finalmente se describe las conclusiones donde hacemos referencia a todos los objetivos cumplidos satisfactoriamente y las recomendaciones describen sugerencias a un futuro para mejorar el presente proyecto de grado.

**Palabras Clave: Información, Geográfica, SIG, XP, Kanban, COCOMO II.**

## **ABSTRACT**

This degree project refers to Geographic Information Systems "GIS", which is transforming many fields of human activity, accompanied by information technologies. It will contribute greatly to the management of information, and also provides direct and interactive access between tasks and functions that were performed manually, are now performed through a web information system.

In the first chapter of the Preliminary Framework, the analysis of the problem is carried out together with the company's background, and from this, the objectives to be developed to solve the problem are defined. Likewise, through all the information obtained, the social, technical and economic justification is described. And finally, the tools to be used are defined, as well as the limits and scope of this project.

In the second chapter of the Theoretical Framework, all the theoretical part is defined, which is established in the previous framework, on which the present project is based.

The third chapter, which is the Application Framework, shows the analysis and development of the system with the support of the XP and Kanban methodology for the planning of the system development process.

The fourth chapter, which is Quality, Cost and Security Metrics, develops the characteristics for software quality with the ISO/IEC 25010 standard, COCOMO II was used for cost estimation and security is described in different parts of the system.

Finally, we describe the conclusions where we make reference to all the objectives satisfactorily fulfilled and the recommendations describe suggestions for the future to improve the present degree project.

**Keywords: Information, Geographic, GIS, XP, Kanban, COCOMO II.**

## LISTADO DE ABREVIATURAS

- UNDSS: Unidad de Análisis del departamento de seguridad de naciones unidas.
- HTTP: Hypertext Transfer Protocol – Protocolo de Transferencia de Hipertexto.
- SIG: Sistemas de Información Geográfica.
- ISO: International Organization for Standardization - Organización Internacional de Normalización.
- XP: Programación Extrema.
- COCOMO: COConstructive COst Model - Modelo Constructivo de Costos.
- HTML: Hypertext Markup Languaje.
- CSS: Cascading Style Sheets - Hojas de Estilo.
- PHP: Personal Home Page.
- H: Historia.
- RF: Requerimiento funcional.
- RNF: Requerimiento no Funcional.

# **CAPÍTULO I**

## **MARCO PRELIMINAR**

## 1.1. INTRODUCCIÓN

Actualmente la tecnología avanza a pasos agigantados para el manejo de la información de acuerdo a las necesidades que se presentan en la sociedad. Es por ello que los Sistemas de Información Geográfica se han constituido durante los últimos años como una de las herramientas más importantes de la informática capaz de gestionar, analizar y modelizar información geográfica.

En este sentido la utilidad de los Sistemas de Información Geográfica facilitará a la empresa a gestionar información y obtener ubicación geográfica como técnica de marketing digital, esto permitirá ofrecer contenido personalizado en base a la localización de las empresas en convenio.

Actualmente FaceBol S.R.L., es una empresa dedicada al marketing y publicidad digital, con una comunidad de más de 450 empresas en convenio y más de 3500 afiliados entre emprendedores y clientes, la misma no cuenta con la geolocalización exacta y detallada de las empresas, además de la gestión de información de las empresas en convenio y clientes afiliados.

El presente proyecto tiene como objetivo aportar en la construcción de conocimiento de los Sistemas de Información Geográfica de la misma manera contribuir e incentivar el interés por los temas de geolocalización y demostrar que existen diferentes formas de ubicar remotamente un determinado lugar, haciendo uso de las tecnologías que van evolucionando.

Las herramientas utilizadas para el desarrollo del presente proyecto son, el lenguaje PHP y JavaScript con Framework de Laravel y para la administración de la base de datos bajo el entorno MySQL, utilizando el servidor apache XAMMP.

Así también para la calidad de Software se hará uso la norma ISO/IEC 25010 que hace referencia a la seguridad de Software.

Finalmente se hará uso del modelo de costos COCOMO II, para determinar el costo y beneficio del presente proyecto.

## **1.2. ANTECEDENTES**

### **1.2.1. Antecedente Institucional**

FaceBol SRL, es una empresa legalmente establecida que se dedica a la publicidad y marketing digital, con una comunidad de cientos de emprendedores, con sede central en la ciudad de El Alto, ubicada a dos cuadras al norte de la plaza Ballivián en plena avenida No 50. Así mismo, brinda capacitación en educación financiera, apoyo a emprendimientos, convenios de publicidad gratuita para empresas y negocios, descuentos de hasta el 50% y promociones de hasta el 2x1, en más de 400 empresas, y ofrece la oportunidad de trabajar, crecer y ser dueño de su propio negocio (FaceBol SRL, 2018).

#### **1.2.1.1. Misión**

Brindar a las personas la oportunidad, de aumentar sus ingresos, reducir sus gastos, e impulsar su educación e inteligencia financiera (FaceBol SRL, 2018).

#### **1.2.1.2. Visión**

Ser una de las empresas más reconocidas a nivel nacional e internacional, y poder dar un aporte en bien de la sociedad (FaceBol SRL, 2018).

### **1.2.2. Antecedentes a fines del proyecto de grado**

#### **1.2.2.1. Antecedentes Internacionales**

- La “Implementación de un Sistema de Información Geográfica en la unidad de análisis del departamento de seguridad de naciones unidas para Colombia”, del postulante William Eduardo torres Gutiérrez, gestión 2015, universidad distrital Francisco José de Caldas, Facultad de Ingeniería de la carrera Ingeniería Catastral y Geodesia, el propósito de este proyecto es la implementación de un sistemas de información geográficos (SIG), llevada a cabo en la unidad de análisis del departamento de seguridad de naciones unidas (UNDSS), esta tiene como objetivo recopilar los hechos y acciones que

afectan la seguridad en las diferentes regiones del país para su posterior análisis, concebido principalmente para funcionarios que operan en terreno realizando funciones humanitarias, de desarrollo, derechos humanos y demás misiones del departamento.

- El “Sistema de Información Geográfico para Visualización y Consulta de Datos de servicio de inserción social del ayuntamiento de Barcelona”, del postulante Karla abad Sacoto, gestión 2012, Universidad Autónoma de Barcelona, el sistema permite a los clientes de la empresa puedan, a través de un browser y conexión a internet, visualizar los resultados de los informes presentados por los educadores de calle.

#### **1.2.2.2. Antecedentes Nacionales**

- El “Sistema de Información para el Control y Seguimiento de Proyectos Distritales Georreferenciados Vía Web”, postulante Wilmer Mendoza Trujillo, Universidad Mayor de San Andrés, carrera de Informática, gestión 2016, el Sistema de Información Vía Web para realizar el control y seguimiento de proyectos de modo que permita disponer información oportuna, rápida y actualizada, para una adecuada toma de decisiones en cuanto a la ejecución física y financiera de los proyectos, además el proyecto se realizó utilizando la metodología de desarrollo RUP con el lenguaje UML.
- El “Sistema de Información Geográfica para el Control de Pedidos y entrega de productos aplicando la ubicación geoespacial” postulante Marcelo Alejo Calle, Universidad Mayor de San Andrés, Carrera de Informática, gestión 2016, el sistema de información geográfico tiene como objeto el control de pedidos y entrega de productos, aplicando la ubicación geoespacial, partiendo de la creciente población y áreas donde se asientan, las empresas o instituciones deben realizar el control de las constantes solicitudes de pedidos de productos y/o servicios que se ofrecen, para esto se piden realizar un croquis del lugar donde se debe entregar o realizar el pedido.

### **1.2.2.3. Antecedentes Locales**

- El “Sistema de Geolocalización de Vehículos Recolectores de basura aplicando internet de las cosas” postulante Israel Guillermo Quispe Mamani, Universidad Pública de El Alto, carrera Ingeniería de Sistemas, gestión 2020, el sistema de geolocalización pretende hacer uso de estas tecnologías aplicado a la geolocalización en tiempo real de los vehículos encargados en la recolección de basura teniendo conocido características principales de la misma. Así también pretende monitorear las rutas por cumplir, los tiempos de llegada y de esta manera coadyuvar en la toma de decisiones para el mejoramiento del servicio.
- “Sistemas de Información Geográficas aplicado al monitoreo de riesgos de quema con imágenes de satélite”, postulante Katteryn Flavia Huanca Villanueva, Universidad Pública de El Alto, carrera Ingeniería de Sistemas, gestión 2020, La propuesta de aplicar sistemas de información geográficos para el monitoreo de riesgos de quema con imágenes de satélite, se fundamenta en la importancia a el monitoreo que describe a un proceso mediante el cual se reúne, observa, estudia y emplea información para luego realizar un seguimiento de un programa o hecho particular, las imágenes de satélite se definen como la representación visual de la información capturada por un sensor montado en un satélite artificial. Estos sensores recogen la información reflejada por la superficie de la Tierra que luego es enviada de regreso a ésta y es procesada convenientemente.

## **1.3. PLANTEAMIENTO DEL PROBLEMA**

### **1.3.1. Problema Principal**

Actualmente la empresa FaceBol S.R.L., no cuenta con un sistema para la gestión de información espacial de las entidades en convenio, lo que genera ineficiencia en la administración comercial.

### **1.3.2. Problemas Secundarios**

- Inexistencia de un visor de ubicaciones en el mapa geográfico de las empresas en convenio, lo cual genera pérdida de enfoque a otras cercanas.
- Ausencia de reportes de las entidades registradas, suscripciones y de afiliaciones, la cual genera pérdida de información relevante para la empresa FaceBol S.R.L.
- No se proporciona comprobante de suscripción y/o afiliación, lo cual genera pérdida de información para la empresa y el cliente.
- Carece de asignación sistematizada de servicios y planes para las empresas en convenio, lo cual genera un trabajo manual y tedioso.

### **1.3.3. Formulación del Problema**

¿De qué manera un Sistema Web de Información Geográfica Orientada al Marketing para las Empresas en Convenio, mejorará la gestión de información espacial de las entidades en convenio para FaceBol S.R.L.?

## **1.4. OBJETIVOS**

### **1.4.1. Objetivo General**

Desarrollar e implementar un Sistema Web de Información Geográfica Orientada al Marketing para las Empresas en Convenio con “FaceBol S.R.L.”, que permita gestionar información espacial de las entidades, para brindar un mejor servicio.

### **1.4.2. Objetivos Específicos**

- Realizar el análisis de la situación actual de la institución.
- Esquematizar la estructura de funcionamiento del sistema.
- Centralizar la información recolectada en una base de datos.
- Desarrollar los módulos del sistema para una mejor organización.
- Hacer uso de APIs de Google Maps, para georreferenciar a las empresas en convenio.

- Implementar el sistema y sus respectivas pruebas de Software.
- Optimizar la generación de reportes, con información que pueda utilizar la empresa.

## **1.5. JUSTIFICACIÓN**

### **1.5.1. Justificación Técnica**

El presente sistema de información geográfica, busca aportar en la construcción de conocimiento de los sistemas de información geográfica de la misma manera contribuir e incentivar el interés por los temas de geo localización y demostrar que existen diferentes formas de ubicar remotamente un determinado lugar, haciendo uso de las tecnologías que van evolucionando.

Además, la empresa cuenta con ordenadores de amplio acceso a internet que permitirá manipular el Sistema Web.

### **1.5.2. Justificación Económica**

La implementación del presente proyecto permitirá reducir los costos que se producen al mover recursos humanos, materiales, etc., para ello los sistemas de información espacial automatizará la misma. Además de que cuenta con un Software Libre, lo cual no tiene costo, y por otro lado el uso de APIs de Google Maps, que tienen licencias de desarrollo, para mostrar la información espacial de las empresas en convenio de forma actualizada.

### **1.5.3. Justificación Social**

La implementación de presente proyecto, aportara de gran manera a la empresa, ya que optimiza tareas en la gestión de información geográfica con un enfoque de marketing digital, a todos aquellos clientes que necesiten de información espacial actualizada, organizada y confiable.

## **1.6. METODOLOGÍA**

### **1.6.1. Metodología XP**

#### **1.6.1.1. Metodología de Programación Extrema XP**

La Programación Extrema XP, es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y proporcionando un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, además donde existe un alto riesgo técnico (Asociación de Investigación, 2006).

#### **1.6.1.2. Características de la Metodología XP**

- Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.
- Se aplica de manera dinámica durante el ciclo de vida del Software.
- Es capaz de adaptarse a los cambios de requisitos.
- Los individuos e interacciones son más importantes que los procesos y herramientas.
- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.

La gente es el principal factor de éxito de un proyecto Software. Es más importante construir un buen equipo que construir el entorno (Bustamante & Rodriguez, 2014).

## **1.6.2. Metodología Kanban**

### **1.6.2.1. Definición de Kanban**

El Kanban es un sistema de gestión donde se produce exactamente aquella cantidad de trabajo que el sistema es capaz de asumir. El Kanban es un sistema de trabajo *just in time*, lo que significa que evita sobrantes innecesarios de stock, que en la gestión de proyectos multimedia equivale a la inversión innecesaria de tiempo y esfuerzo en lo que no necesitaremos (o simplemente es menos prioritario) y evita sobrecargar al equipo.

En la gestión del trabajo en curso con Kanban, se busca un concepto clave como es limitar el trabajo en curso. Está demostrado que, cuanto más trabajo en curso se gestione a la vez, los índices de calidad disminuyen drásticamente. En la producción de proyectos multimedia, aumentar el trabajo en curso implica aumentar la cantidad de errores que este proyecto multimedia tendrá como consecuencia de la poca capacidad de concentración que los desarrolladores podrán dedicarles a las tareas (Bermejo, El Kanban, 2010).

## **1.6.3. Métrica de Calidad del Software**

### **1.6.3.1. ISO/IEC 25010**

Según el estándar (ISO 25000, 2022), menciona que el modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. 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. 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.

## 1.6.4. Estimación de Costos

### 1.6.4.1. Modelo COCOMO II

El modelo COCOMO (CONstructive COSt MOdel: modelo constructivo de costos), el modelo COCOMO original se convirtió en uno de los modelos de estimación de costo más ampliamente utilizados y estudiados en la industria. Evoluciono hacia un modelo de estimación más exhaustivo, llamado COCOMO II. Como su predecesor, COCOMO II en realidad es una jerarquía de modelos de estimación que aborda las áreas siguientes:

- *Modelo de composición de aplicación.* Se usa durante las primeras etapas de la ingeniería de Software, cuando son primordiales la elaboración de prototipos de las interfaces de usuario, la consideración de la interacción del Software y el sistema, la valoración del rendimiento y la evaluación de la madurez de la tecnología.
- *Modelo de etapa temprana de diseño.* Se usa una vez estabilizados los requisitos y establecida la arquitectura básica del Software.
- *Modelo de etapa postarquitectónica.* Se usa durante la construcción del Software.

Como todos los modelos de estimación para Software, los modelos COCOMO II requieren información sobre dimensionamiento. Como parte de la jerarquía del modelo, están disponibles tres diferentes opciones de dimensionamiento: puntos objetos, puntos de función y líneas de código fuente (Boehm, 1981).

El modelo COCOMO II considera diferentes enfoques para el desarrollo de Software, como el de la construcción de prototipos, el desarrollo basado en componentes y el uso de programación con bases de datos (Sommerville, 2005).

## 1.6.5. Seguridad

- **Seguridad Web**

La seguridad en Web tiene 3 etapas primarias las cuales son:

- Seguridad de la computadora de usuario.
- Seguridad del servidor Web y de los datos almacenados ahí.
- Seguridad de la información que viaja entre el servidor Web y el usuario.
- **Seguridad de password con Bcrypt**

Una contraseña o clave (en inglés password) es una forma de autenticar al acceso de la información secreta para el control del mismo. Por tanto, la contraseña debe ser único y directo a aquellos que tengan el permiso para acceder a la información (Taylor, 2022).

- **Hash**

La fachada de Laravel hash proporciona hash Bcrypt y Argon2 seguro para almacenar contraseñas de usuario (Taylor, 2022).

## **1.6.6. Pruebas al Software**

### **1.6.6.1. Caja Blanca**

Según los autores (Maldonado, Alva, & Santa Cruz, 2021), las pruebas de caja blanca conocida también como caja de cristal o caja transparente. Es una técnica de diseño de caso de prueba que usa la estructura de control para obtener los casos de prueba. Dentro de esta estructura de control podemos encontrar la estructura de un componente de Software como pueden ser sentencias de decisiones, caminos distintos del código, la estructura de una página web, etc.

Los métodos de pruebas de caja blanca aportan los siguientes puntos:

- Garantizan que todas las rutas del código se revisan al menos una vez.
- Revisan las condiciones lógicas.
- Revisan las estructuras de datos.

### **1.6.6.2. Caja Negra**

Según los autores (Maldonado, Alva, & Santa Cruz, 2021), las pruebas de caja negra, llamada pruebas de comportamiento. Son las que utilizan el análisis de especificación, tanto funcional como no funcional, sin tener en cuenta la estructura

interna del programa para diseñar los casos de pruebas. A diferencia de las pruebas de caja blanca, estas suelen realizarse durante las últimas etapas de las pruebas.

Con los métodos de caja negra se intentan encontrar los errores:

- Funciones incorrectas y faltantes.
- Errores de inicialización y terminación.
- Errores de interfaz.
- Errores en la estructura.

### **1.6.6.3. Pruebas de Estrés**

Con las pruebas de estrés se somete el sistema a cargas pesadas de datos o de actividad en un tiempo corto. Enlistamos también ejemplos de otros tipos de prueba: de seguridad, de usabilidad, de almacenamiento, de configuración, de compatibilidad, de instalación, de confiabilidad, y de recuperación a fallas (Gómez, Cervantes, & Gonzales, 2019).

## **1.7. HERRAMIENTAS**

### **1.7.1. Hardware**

Un equipo de computación ya sea de escritorio o portátil, tiene la importante tarea de unidad de procesamiento central de información y almacenamiento, para que la información fluya a través de un sistema de computación de manera rápida y eficiente.

### **1.7.2. Software**

#### **1.7.2.1. Lenguajes de programación**

- **PHP**

PHP es un acrónimo recursivo para “PHP: Hypertext Preprocessor”, originalmente Personal Home Page, es un lenguaje interpretado libre, usado originalmente solamente para el desarrollo de aplicaciones presentes y que actuaran en el lado del servidor, capaces de generar contenido dinámico en la World Wide Web. Figura

entre los primeros lenguajes posibles para la inserción en documentos HTML, dispensando en muchos casos de archivos externos para eventuales procedimientos de datos. El código es interpretado en el lado del servidor por el módulo PHP, que también genera la página web para ser visualizada en el lado del cliente. El lenguaje evolucionó, pasó a ofrecer funcionalidades en la línea de comandos, y además ganó características adicionales, que posibilitaron usos adicionales del PHP. Es posible instalar el PHP en la mayoría de los sistemas operativos, totalmente de manera gratuita. Siendo competidor directo de la tecnología ASP perteneciente a Microsoft, PHP es utilizado en aplicaciones como Media Wiki, Facebook, Drupal, Joomla, WordPress, Magento y Oscommerce (Arias, 2017).

- **JavaScript**

JavaScript, comúnmente abreviado como JS, es un lenguaje de programación del tipo interpretado. Está basado en el estándar ECMAScript y, a pesar de ser débilmente tipado y dinámico, también se define como un lenguaje orientado a objetos.

JavaScript nació el año 1995, prácticamente de la mano de la WWW, o el Internet comercial. Fue diseñado en un principio por Netscape Communications (hoy, Mozilla Foundation). Integrado originalmente como una especie de plugin en los navegadores de la primera era web, fue muy criticado por su lentitud para procesar código, dado que esta acción generaba un retardo notable en la carga completa de una página. A los pocos años de vida fue destronado por Flash Player, pero tan solo una década después, volvió completamente recargado, y recuperó su lugar y labor dentro del desarrollo de sitios web (Luna, 2019).

- **jQuery**

jQuery es una librería de JavaScript muy popular entre los programadores web que permite simplificar significativamente el código (Recio, 2016).

El enfoque de jQuery no consiste solo en una codificación más intuitiva y concisa de los scripts, sino que su filosofía permite concentrarse en el conjunto de elementos que gestiona el DOM. El JavaScript tradicional, en su evolución histórica, se ha tenido que acoplar al Document Object Model. Con jQuery, John Resing ha

reconsiderado de alguna forma el JavaScript manteniéndolo como un verdadero lenguaje de programación, basado en el DOM (Aubry & Van, 2017).

- **HTML - 5**

El HyperText Markup Lenguaje (HTML) es un lenguaje de programación. A diferencia de otros lenguajes, no está compuesto por instrucciones, sino por un conjunto de etiquetas que organizan y declaran el propósito de cada contenido del documento.

En el sentido estricto, HTML es un texto escrito con una sintaxis particular que el navegador es capaz de leer y aplicar. Se trata de un lenguaje que fue creado para compartir a través de Internet no solo el texto incluido en los documentos sino también su formato (Gauchat, 2013).

El lenguaje HTML debe funcionar en diferentes plataformas y navegadores, ha sido desarrollado con la premisa de que cualquier tipo de dispositivo debería ser capaz de usar información de la Web (ordenadores, teléfonos móviles, PDA's, etc.), para ello ha sido necesario un gran consenso entre los interlocutores de la industria cada vez aparecía una nueva versión de HTML, para que así cada nueva versión sea el estándar que sea implementado por los diferentes navegadores Web (Casado, 2019).

- **CSS 3**

Las Hojas de Estilo (o CSS, por Cascading StyleSheets) son un mecanismo que permiten aplicar formato a los documentos escritos en HTML ó XHTML (y en otros lenguajes estructurados, como XML) separando el contenido de las páginas de su apariencia. Para el diseñador, esto significa que la información estará contenida en la página HTML, pero este archivo no debe definir cómo será visualizada esa información, sino la hoja de estilos asociada a él. Las indicaciones acerca de la composición visual del documento estarán especificadas en el archivo de la CSS (Florián, 2007).

### **1.7.3. Framework**

- **Laravel 8**

Laravel es el framework PHP más utilizado del momento. Su filosofía es crear código simple y elegante.

En un principio, Laravel no fue creado con el patrón de arquitectura MVC, y su foco estaba puesto principalmente en resolver problemas de autenticación. No obstante, la primera versión incorporaba funcionalidades que fueron bien recibidas, y de forma rápida, por la comunidad de desarrolladores.

La segunda versión tardó menos de seis meses en salir al mercado. Laravel terminó de adoptar el patrón MVC para su arquitectura e incorporó el siguiente slogan:

“liberándote del código espagueti, Laravel te ayuda a crear aplicaciones maravillosas usando una sintaxis simple y expresiva. El desarrollo debe ser una experiencia creativa que disfrutes, no algo que sea doloroso. Disfruta del aire fresco” (Cíceri, 2018).

- **Angular 10**

Angular es una plataforma que permite desarrollar aplicaciones web en la sección cliente utilizando HTML y JavaScript para que el cliente asuma la mayor parte de la lógica y descargue al servidor con la finalidad de que las aplicaciones ejecutadas a través de Internet sean más rápidas. El hecho de estar mantenido por Google, así como una serie de innumerables razones técnicas, ha favorecido su rápida adopción por parte de la comunidad de desarrolladores.

Además de mejorar el rendimiento de las aplicaciones web, su utilización en dispositivos móviles está optimizada ya que, en ellos, los ciclos de CPU y memoria son críticos para su óptimo funcionamiento (Boada & Gómez, 2018).

### **1.7.4. Base de Datos**

- **MySQL**

Dentro de las múltiples posibilidades que los sitios dinámicos ofrecen al usuario está la obtención de información almacenada en una base de datos en el servidor.

MySQL es gratuito (aunque también existen licencias comerciales). Por otra parte, los datos se almacenan en archivos que tienen un tamaño mucho menor que otras bases de datos. Además, el motor de datos de MySQL es mucho más rápido, tanto grabando datos como localizándolos y recuperándolos, que el de otras bases de datos. Eso sin contar con que MySQL ofrece una gran seguridad sobre la integridad de los datos almacenados (López, 2014).

## **1.8. LÍMITES Y ALCANCES**

### **1.8.1. Limites**

Las limitaciones del presente proyecto, son aspectos más allá de sus funciones, que son inalcanzables al sistema.

- El sistema no presentará proyección de ventas.
- El sistema no proporcionará información de inventarios.
- El sistema no efectuará proceso de facturación.
- El sistema no realizará gestión de correos.

### **1.8.2. Alcances**

La implementación de un sistema para la información geográfica de empresas en convenio orientada al marketing tendrá los siguientes alcances:

- Módulo de gestión usuarios.  
El administrador del sistema podrá crear, modificar e inhabilitar usuarios que accedan al sistema.
- Módulo de configuración.  
El administrador de sistema gestiona categorías, ciudades y planes a las empresas que se registran en el sistema.
- Módulo de gestión de empresas.  
El administrador del sistema registra a las empresas en convenio, según información requerida.
- Módulo de suscripción.

El administrador del sistema suscribe a las empresas en convenio para tener constancia de empresas activas.

- Módulo de gestión de afiliación

El administrador del sistema registra las afiliaciones de clientes para poder gozar descuentos y promociones de las empresas en convenio.

- Módulo de pagos.

El administrador del sistema gestiona los ingresos de afiliaciones y convenio registradas al sistema.

- Módulo de reportes

El administrador del sistema genera reportes de las todas las empresas, afiliaciones y suscripciones, registradas en el sistema.

## **1.9. APORTES**

El presente proyecto, aporta de gran manera a la empresa, en la gestión de información espacial de todas sus entidades en convenio, para tener una información actualizada y confiable, así mismo en la gestión de nuevas afiliaciones, suscripciones y pagos generados por los mismos.

# **CAPÍTULO II**

## **MARCO TEORICO**

## **2.1. INTRODUCCIÓN**

En el presente proyecto de grado, se inicia con la recopilación de toda la información posible, referente a los Sistemas de Información Geográfica SIG, además de su importancia y aplicación en diferentes áreas.

Así mismo se muestra información sobre el Marketing digital, como técnica de comercialización para mostrar la esencia de la marca de las empresas.

Además, se describe información sobre las herramientas, metodologías, etc. que se usa en el todo el proceso de desarrollo del presente proyecto de grado.

## **2.2. SISTEMA**

La teoría general de sistemas define, pues, un sistema como un conjunto de elementos interrelacionados según las normas de cierta estructura y que persigue un fin común. Todo sistema se puede dividir en subsistemas. Dado que la empresa se comporta como un sistema, es posible fragmentar sus partes en subsistemas (Lapiedra, Fore, & Martinez, 2021).

## **2.3. INFORMACIÓN**

La información es un conjunto de datos transformados de forma que contribuye a reducir la incertidumbre del futuro y por tanto ayuda a la toma de decisiones. La información representa los datos transformados de forma significativa para la persona que los recibe, es decir, tiene un valor real o percibido para sus decisiones y para sus acciones. Así pues, la información son datos que han sido interpretados y comprendidos por el receptor del mensaje. La relación entre los datos y la información es equivalente a la que existe entre la materia prima y el producto acabado. Una información será significativa en cuanto que sea útil como materia prima para una decisión determinada (Lapiedra, Fore, & Martinez, 2021).

## **2.4. SISTEMA WEB**

Según el autor (Garcia, 2022) define a los sistemas web de la siguiente manera:

- Están orientados a documentos que contienen páginas web estáticas o dinámicas.
- Se centran en el look & feel y enfatizan la creatividad visual y la presentación en la interfaz.
- Son conducidos por el contenido, incluyendo el desarrollo del contenido.
- Necesitan ofrecer servicios a usuarios con diversidad de características y capacidades.
- Ejemplifican los vínculos entre el arte y la ciencia que generalmente aparecen en el desarrollo del software.
- Los desarrolladores de los sistemas web:
  - Difieren en gran medida en su formación, características, conocimiento y comprensión del sistema.
  - Diferencias en su percepción de la Web y de la calidad del sistema web.

## **2.5. SISTEMA DE INFORMACIÓN GEOGRÁFICA**

### **2.5.1. Definición**

Según los autores (Morea & Huerta, 2005), definen a los sistemas de información geográfica como:

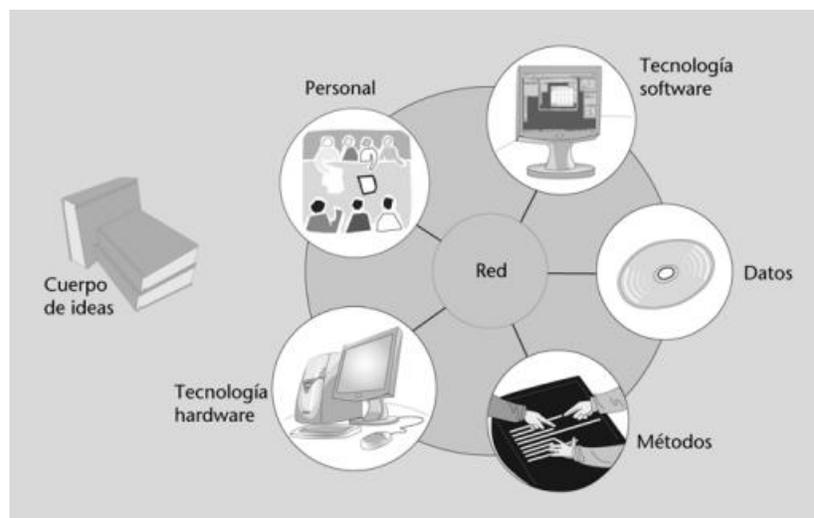
- Base de datos computarizada que contiene información espacial.
- Tecnología informática para gestionar y analizar información espacial.
- Conjunto de herramientas para reunir, introducir en el ordenador, almacenar, recuperar, transformar y cartografiar datos espaciales sobre el mundo real para un conjunto particular de objetivos.
- Tipo especializado de base de datos con capacidad de manejar datos geográficos (espacialmente referenciados) y que se pueden representar como imágenes.

- Sistema hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión.
- Conjunto de mapas de la misma porción del territorio donde un lugar concreto tiene la misma localización en todos los mapas incluidos en el sistema, sobre la que se aplica una tecnología para resolver problemas territoriales.

### 2.5.2. Componentes de los Sistemas de Información geográfica

Según los autores (Rodríguez & Olivella, 2006) identifican los componentes de los SIG con el programa informático preparado para trabajar con datos georreferenciados, pero un SIG no sólo es un software instalado en un ordenador, sino que es un conjunto de componentes: tecnológicos (programas y equipos, es decir, software y hardware), datos, métodos, personal cualificado, ideas y red sin los cuales no sería posible su existencia.

**Figura 2.1.** Componentes que forman un SIG



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

A continuación, se muestra a cada componente de manera particular:

- **Tecnología (Software y Hardware)**

La tecnología es un componente del SIG que puede definirse como el software y el hardware. Eso incluye un conjunto de procesos que son la base de un software SIG y que consisten en una serie de algoritmos que sirven para acceder, presentar, analizar y sintetizar los datos almacenados en la base de datos, en función de sus atributos espaciales y también no espaciales. Relacionados con este software, hay otros para gestionar datos, para extraer información de una base de datos, para visualizar los datos y para llevar a cabo otras funciones, como la importación y exportación de los datos. Estos programas habitualmente están integrados en el sistema operativo y se usan junto con otras aplicaciones en una misma sesión de ordenador.

El hardware se centra en la plataforma informática y tiene dispositivos periféricos de entrada y salida. Eso incluye aparatos de lectura convencionales que permiten la transferencia de archivos, así como los mecanismos de red, junto con aparatos de salida, como por ejemplo las impresoras. Además, incluye dispositivos específicos utilizados ampliamente en aplicaciones SIG que sirven para la entrada y salida de mapas tanto en formato digital como en formato analógico

- **Ideas**

El cuerpo de ideas es el conjunto de conocimientos que hacen progresar y desarrollar el uso de los SIG. Dentro del cuerpo de ideas que hay detrás del desarrollo del hardware, debemos considerar la ingeniería, las matemáticas y la física. Con respecto al software, hemos de considerar la geomática, los algoritmos para el procesamiento de datos, las bases de datos y el análisis espacial, entre otras cosas. Detrás de la cuestión de organización, están las teorías de gestión, el diseño de sistemas, el comercio, la sociología, la psicología, la psicofísica, la ergonomía... Los métodos se basan en el cuerpo teórico de distintas ramas de la ciencia como la sociología, la gestión, el comercio, etc.

- **Personal**

Un SIG sólo tiene sentido en el contexto de una organización. La organización está formada por una gran variedad de componentes, pero nosotros la definimos como el conjunto de objetivos, procesos, gestión, operadores y personal. Antes de definir el SIG de una organización, debemos prestar especial atención al proceso de gestión, a los operadores y al personal, ya que todos estos factores tienen una repercusión directa sobre el sistema que queremos diseñar, cómo lo vamos a implantar y los mecanismos de control que deberían seguirse.

- **Red**

La red se considera recientemente un componente fundamental de los SIG, ya que permite la comunicación y compartir información de manera rápida y eficaz. Los SIG aparecen con fuerza en las redes, tanto en Internet como en las intranets de las organizaciones. A medida que Internet se convierte día a día en un canal de comunicación más importante y ofrece más posibilidades para transmitir y recibir todo tipo de información, los SIG también se van adaptando a este desarrollo. Eso significa que, para los usuarios de información geográfica, que hasta hace poco tenían problemas para trabajar o bien recabar buenos datos, se abren nuevas vías de información para poder obtener y comparar la nueva información proveniente de Internet.

- **Datos**

Para que un SIG funcione, debemos implicarnos en procesos (que pueden ser complejos) de transferencia y acceso a los datos. Eso conlleva que debemos mantener y asegurar la calidad de los datos almacenados en la base de datos y considerar la relación que hay entre los datos, la tecnología y los métodos. Una vez identificados los objetos del modelo del mundo real, con los datos se identifican las propiedades que lo forman mediante sus atributos o elementos descriptivos, el tipo de geometría y el elemento espacial. Los principales componentes del sistema de información relacionados con los datos son los de entrada de datos, las bases de datos espaciales y no espaciales y los sistemas de control de calidad.

- **Métodos**

Los métodos son procedimientos independientes o normas para llevar a cabo diferentes tareas relacionadas con el diseño, creación y funcionamiento de los SIG. Hay métodos específicos para el análisis espacial, la manipulación de datos, el diseño de una base de datos, el análisis de las necesidades de los usuarios, la interpretación de mapas, etc. Cada proceso tiene un método y éste es el que determina el procedimiento lógico y las especificaciones de cada acción. El método tiene la finalidad de establecer la estructura de un SIG y, en concordancia con ésta, implementar aplicaciones que faciliten la toma de decisiones. Es el método lo que determina la calidad del resultado obtenido o de la acción que hemos llevado a cabo. Por eso, el método es la clave de todo en las operaciones SIG y, entre otras cosas, será lo que determine el éxito o fracaso del proyecto.

### **2.5.3. Funciones de un SIG**

Según los autores (Morea & Huerta, 2005), describen cinco funciones de un Sistema de Información Geográfica, las cuales son:

- **Incorporación de la información**

Se incluye aquí no solo el paso de datos geográficos a una forma analítica (digitalización), sino también la integración en el sistema de las bases de datos. Previamente, hay que realizar un análisis lo más exhaustivo posible de la información necesaria para su correcta estructuración en función de las necesidades del proyecto.

- **Gestión de la información**

Toda la información incorporada debe poder ser gestionada por el sistema realizando búsquedas temáticas, espaciales y con capacidad de selección multi condicionadas para su posterior análisis. La gestión se refiere tanto a datos geográficos como alfanuméricos.

- **Análisis de la información**

Esta función es la específica de los SIG y es su elemento característico. A partir de la preparación de los datos (apartados A y B) se utilizan procedimientos de análisis territorial para la obtención de resultados. En realidad, son modelos resultantes del tratamiento inteligente de la información, condicionados por las hipótesis contempladas. Los resultados pueden ser más o menos reales en función de la bondad de los datos y de la veracidad de las hipótesis utilizadas.

- **Interrelación con el usuario**

Los puntos anteriores deben tener su finalización en un relación práctica e intuitiva con el usuario del sistema, de tal forma que pueda elegir la salida más adecuada de los datos utilizados y de los resultados obtenidos en relación con los fines perseguidos. Un SIG debe tener capacidad para ofrecer salidas tales como mapas, gráficos, tablas resumen y en cualquier soporte (papel, pantalla, exportación de datos. etc.).

#### **2.5.4. Funciones de los Sistemas de Información Geográfica**

Según los autores (Rodríguez & Olivella, 2006), los programas SIG tienen una serie de funciones diseñadas para la gestión de información geográfica como son la entrada de datos (la creación de nuevos datos o el paso de información analógica a información digital), el almacenamiento, la recuperación, el análisis y la consulta de datos (para la creación de bases de datos, topología, consultas gráficas, alfanuméricas, combinadas, superposición de planos e información) y, finalmente, las salidas de datos (impresión de informes, traficación de planos y publicación en varios formatos electrónicos).

Según Las funciones de un SIG se deben a la entrada de datos, almacenamiento, recuperación, análisis y consulta y salida de datos. A continuación, se describe cada función:

- **Entrada de datos**

Según el modo de recogida diferenciamos dos tipos de datos:

- Los datos primarios son aquellos que genera un individuo u organización con la finalidad principal de usarlos ellos mismos. Normalmente, han sido adquiridos siguiendo unas especificaciones (escala y resolución, clasificaciones y niveles de agregación, etc.).
- Los datos secundarios son datos obtenidos mediante distribuidores externos. Una manera eficiente de implementar un SIG es limitar, desde un principio, la cantidad de tiempo y dinero necesario para desarrollar las bases de datos.

- **Almacenamiento, recuperación, análisis y consulta**

En los procesos de captura se generan gran cantidad de datos espaciales que hay que almacenar de alguna manera. Nos interesara guardar datos sobre los objetos geográficos: la geometría, los atributos y los métodos de acceso para poder recuperarlos posteriormente.

La información geográfica requiere grandes capacidades de almacenamiento de datos. Una vez almacenadas en el SIG, normalmente el usuario estará interesado en llevar a cabo diferentes operaciones de recuperación y análisis de los datos.

- **Salida de datos**

El proceso de salida consiste en trasferir los datos, imágenes o mapas que contiene un SIG a otro medio o soporte. Esta operación no implica necesariamente ningún proceso de conversión de datos para transformarlos del formato original al formato receptor.

## **2.6. MARKETING**

El Marketing es un conjunto de estrategias, principios y prácticas para analizar el comportamiento del consumidor y cumplir sus necesidades y expectativas, buscando el posicionamiento de un producto o servicio en la mente del consumidor.

Por ello, es necesario que dentro de las estrategias creadas haya que recordar siempre la necesidad de dar un buen servicio y vender con el corazón. El Marketing debe tener un espíritu libre, para sentir, imaginar, pensar, analizar, trazar, diseñar, crear y materializar las estrategias que permitan conquistar la mente, espíritu y

corazón del consumidor. Asimismo, se debe enfocar en vender experiencias y emociones, donde sea capaz de mover y retumbar el mundo (Shum, 2019).

### **2.6.1. ¿Qué es Marketing Digital?**

El Marketing digital es la aplicación de estrategias y técnicas de comercialización llevadas a cabo a través de los medios digitales. Todas las técnicas de marketing off-line son adaptadas al mundo digital.

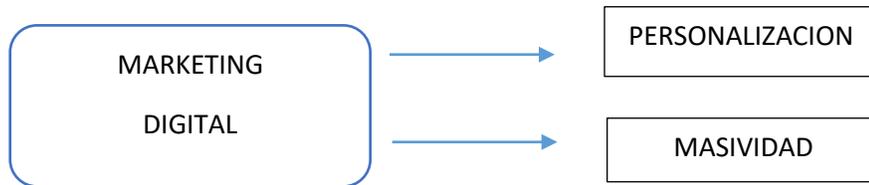
El marketing digital está caracterizado por el uso de dispositivos electrónicos tales como computadoras, teléfonos inteligentes (Smartphones), teléfonos celulares, tableta (tablet), smartTV's y consolas de videojuegos. El marketing digital utiliza la tecnología y las plataformas como sitios webs, correo electrónico, aplicaciones móviles, foros, blogs, canales de televisión, radios, revista, periódicos digitales y redes sociales para conectar e interactuar con el cliente.

Muchas agencias, organizaciones y marcas combinan los canales tradicionales y digitales del marketing, debido a que amplía más la posibilidad de penetración, y si el objetivo es captar clientes, no solo se debe tener presencia exclusiva en los medios on-line y off-line, sino all-line, esto significa que se debe estar en todos los canales de comunicación que te permiten captar, conectar y comunicar con la audiencia (Shum, 2019).

### **2.6.2. Características del Marketing Digital**

Según el autor (Selman, 2017), define que el Marketing Digital se caracteriza por dos aspectos fundamentales: la personalización y la masividad. Los sistemas digitales permiten crear perfiles detallados de los usuarios, no solo en características sociodemográficas, sino también en cuatro a gustos, preferencias, intereses, búsquedas, compras. La información que se genera en Internet puede ser totalmente detalla. De esta manera, es más fácil conseguir un mayor volumen de conversión en el mundo online que en el mundo tradicional. Esto es lo que se conoce como personalización. Además, la masividad supone que con menos presupuesto tienes mayor alcance y mayor capacidad de definir cómo tus mensajes llegan a públicos específico.

**Figura 2.2.** Características del Marketing Digital.



**Fuente:** (Elaboración propia).

Además, el Marketing Digital se basa en las 4F, que son:



**Figura 2.3.** Las 4F del Marketing Digital.

**Fuente:** (Elaboración Propia).

- **Flujo**

El flujo es la dinámica que un sitio web propone al visitante. El usuario se tiene que sentir atraído por la interactividad que genera el sitio e ir de un lugar a otro, según se ha planeado.

- **Funcionalidad**

La navegabilidad tiene que ser intuitiva y fácil para el usuario; de esta manera, previenes que abandone la página por haberse perdido. El sitio debe captar su atención y evitar que abandone la página.

- **Feedback (retroalimentación)**

Debe haber una interactividad con el internauta para construir una relación de confianza con él. Las redes sociales ofrecen una excelente oportunidad para esto.

- **Fidelización**

Una vez que has entablado una relación con tu visitante, la fidelización consiste en lograr que esa relación se extienda a largo plazo. Usualmente, la fidelización se logra con la entrega de contenidos atractivos para el usuario.

## **2.7. INGENIERÍA DE SOFTWARE**

### **2.7.1. ¿Qué es la ingeniería de Software?**

La ingeniería de Software se ha definido por varios autores. Según Ian Sommerville, considerado uno de los padres de la ingeniería de Software, la ingeniería de Software "es una disciplina de la ingeniería que comprende todos los aspectos de la producción del Software" (Sommerville, 2005).

La IEEE (The Institute of Electrical and Electronics Engineers: el Instituto de Ingenieros Electricistas y en Electrónica), es la asociación internacional más grande del mundo formada por profesionales de las nuevas tecnologías, como ingenieros electricistas, electrónicos, en sistemas y en telecomunicaciones. La IEEE define a la ingeniería de Software como "la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del Software".

Una de las definiciones más interesantes es la de Bohem: "ingeniería de Software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de Software o producción de Software". Definimos la ingeniería de Software como: Una disciplina en la que se aplican técnicas y principios de forma sistemática en el desarrollo de sistemas de Software para garantizar su calidad (Boehm, 1981).

### **2.7.2. ¿Por qué es importante la ingeniería de Software?**

Normalmente, los clientes que mandan construir un sistema de Software lo quieren lo más pronto posible (o antes). Es por esto que para "acabar más rápido" existe la tentación de comenzar a codificar y probar sin hacer antes un buen análisis de cuáles son los requerimientos del cliente ni un diseño. Si no se tienen claros los

requerimientos del cliente, entonces lo más probable es que el sistema que se desarrolle no cumpla con sus necesidades. Cuando no se cuenta con un diseño, se cometen todo tipo de errores y es muy difícil encontrarlos y resolverlos de forma adecuada. Lo más probable es que el tiempo invertido para que el sistema funcione haya sido mayor al que se hubiera invertido aplicando ingeniería de Software. Como consecuencia, los desarrolladores quedan frustrados y exhaustos, y el cliente difícilmente quedará satisfecho. Además, al final se entrega un producto de mala calidad al que difícilmente se le podrá dar un buen mantenimiento.

La experiencia dice que, si en un proyecto grande no se aplica ingeniería de Software, éste fracasará. La probabilidad de que un proyecto sea exitoso es mucho mayor cuando se aplica la ingeniería de Software (Gómez, Cervantes, & Gonzales, 2019).

### **2.7.3. Procesos del Software**

Un proceso de Software es una serie de actividades relacionadas que conduce a la elaboración de un producto de Software. Estas actividades pueden incluir el desarrollo de Software desde cero en un lenguaje de programación estándar como Java o C. Sin embargo, las aplicaciones de negocios no se desarrollan precisamente de esta forma. El nuevo Software empresarial con frecuencia ahora se desarrolla extendiendo y modificando los sistemas existentes, o configurando e integrando el Software comercial o componentes del sistema.

Existen muchos diferentes procesos de Software, pero todos deben incluir cuatro actividades que son fundamentales para la ingeniería de Software:

- a) Especificación del Software: Tienen que definirse tanto la funcionalidad del Software como las restricciones de su operación.
- b) Diseño e implementación del Software: Debe desarrollarse el Software para cumplir con las especificaciones.
- c) Validación del Software: Hay que validar el Software para asegurarse de que cumple lo que el cliente quiere.

- d) Evolución del Software: El Software tiene que evolucionar para satisfacer las necesidades cambiantes del cliente.

En cierta forma, tales actividades forman parte de todos los procesos de Software. Por supuesto, en la práctica éstas son actividades complejas en sí mismas e incluyen subactividades tales como la validación de requerimientos, el diseño arquitectónico, la prueba de unidad, etcétera. También existen actividades de soporte al proceso, como la documentación y el manejo de la configuración del Software.

Cuando los procesos se discuten y describen, por lo general se habla de actividades como especificar un modelo de datos, diseñar una interfaz de usuario, etcétera, así como del orden de dichas actividades. Sin embargo, al igual que las actividades, también las descripciones de los procesos deben incluir:

- Productos, que son los resultados de una actividad del proceso. Por ejemplo, el resultado de la actividad del diseño arquitectónico es un modelo de la arquitectura de Software.
- Roles, que reflejan las responsabilidades de la gente que interviene en el proceso. Ejemplos de roles: gerente de proyecto, gerente de configuración, programador, etcétera.
- Precondiciones y pos condiciones, que son declaraciones válidas antes y después de que se realice una actividad del proceso o se cree un producto. Por ejemplo, antes de comenzar el diseño arquitectónico, una precondición es que el cliente haya aprobado todos los requerimientos; después de terminar esta actividad, una pos condición podría ser que se revisen aquellos modelos UML que describen la arquitectura (Sommerville, 2005).

#### **2.7.4. Modelo de Proceso de Software**

Un modelo de proceso de Software es una representación simplificada de este proceso. Cada modelo del proceso representa a otro desde una particular perspectiva y, por lo tanto, ofrece sólo información parcial acerca de dicho proceso. Por ejemplo, un modelo de actividad del proceso muestra las actividades y su secuencia, pero quizá sin presentar los roles de las personas que intervienen en

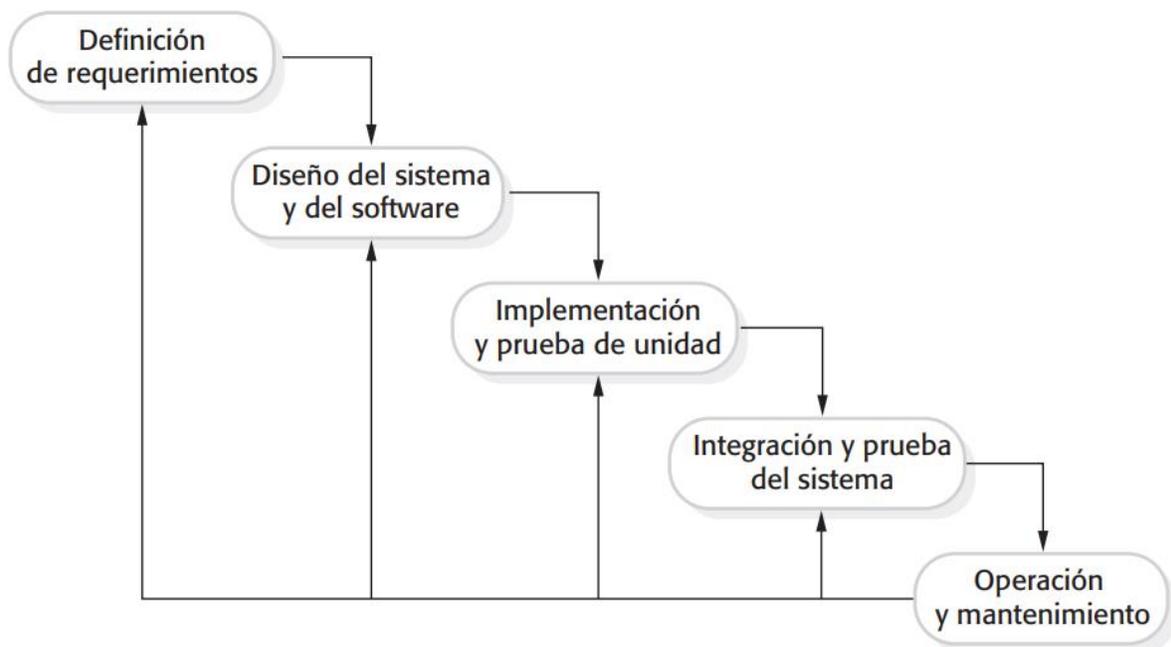
esas actividades. En esta sección se introducen algunos modelos de proceso muy generales (en ocasiones llamados “paradigmas de proceso”) y se muestran desde una perspectiva arquitectónica. En otras palabras, se ve el marco (framework) del proceso, pero no los detalles de las actividades específicas.

Tales modelos genéricos no son descripciones definitivas de los procesos de Software. Más bien, son abstracciones del proceso que se utilizan para explicar los diferentes enfoques del desarrollo de Software. Se pueden considerar marcos del proceso que se extienden y se adaptan para crear procesos más específicos de ingeniería de Software.

Los modelos del proceso que se examinan aquí son:

1. El modelo en cascada (waterfall): Éste toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución y, luego, los representa como fases separadas del proceso, tal como especificación de requerimientos, diseño de Software, implementación, pruebas, etc.

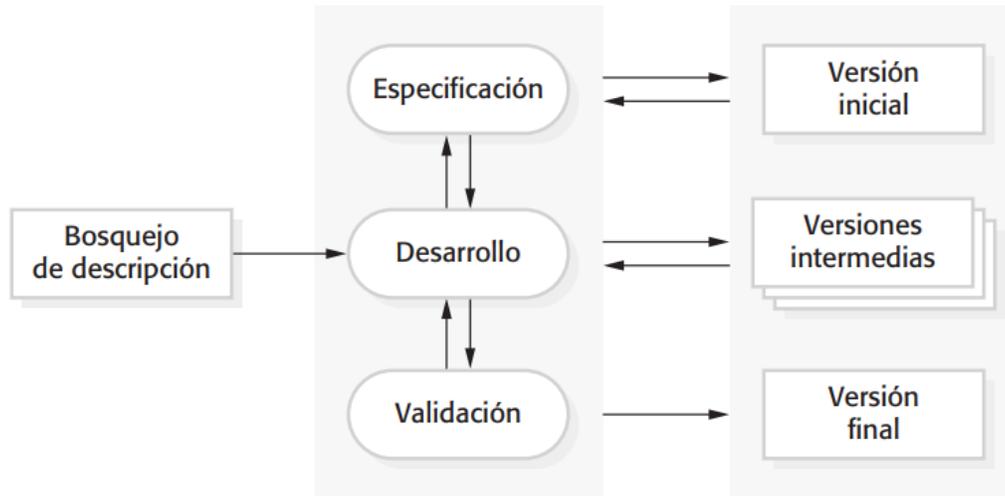
**Figura 2.4.** Modelo en Cascada



*Fuente: (Sommerville, 2011)*

2. Desarrollo incremental: Este enfoque vincula las actividades de especificación, desarrollo y validación. El sistema se desarrolla como una serie de versiones (incrementos), y cada versión añade funcionalidad a la versión anterior.

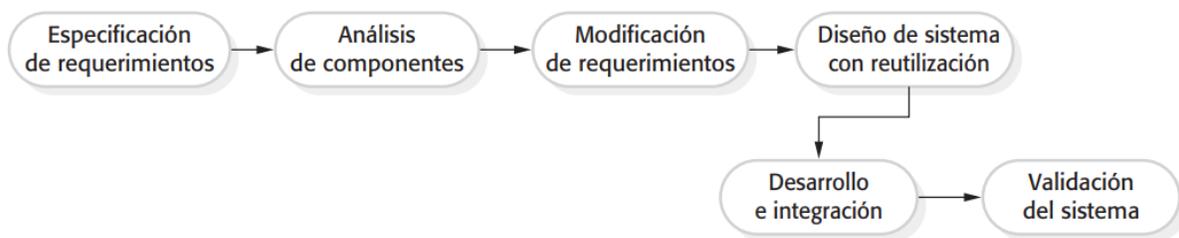
**Figura 2.5.** Desarrollo incremental



**Fuente:** (Sommerville, 2011).

3. Ingeniería de Software orientada a la reutilización: Este enfoque se basa en la existencia de un número significativo de componentes reutilizables. El proceso de desarrollo del sistema se enfoca en la integración de estos componentes en un sistema, en vez de desarrollarlo desde cero.

**Figura 2.6.** Ingeniería de Software orientada a la reutilización



**Fuente:** (Sommerville, 2011).

Dichos modelos no son mutuamente excluyentes y con frecuencia se usan en conjunto, sobre todo para el desarrollo de grandes sistemas. Para este tipo de sistemas, tiene sentido combinar algunas de las mejores características de los

modelos de desarrollo en cascada e incremental. Se necesita contar con información sobre los requerimientos esenciales del sistema para diseñar la arquitectura de Software que apoye dichos requerimientos. No puede desarrollarse de manera incremental. Los subsistemas dentro de un sistema más grande se desarrollan usando diferentes enfoques. Partes del sistema que son bien comprendidas pueden especificarse y desarrollarse al utilizar un proceso basado en cascada. Partes del sistema que por adelantado son difíciles de especificar, como la interfaz de usuario, siempre deben desarrollarse con un enfoque incremental. (Sommerville, 2005).

## **2.8. METODOLOGÍA**

### **2.8.1. Metodología de Programación Extrema XP**

La programación extrema o Extreme Programming (XP) es una metodología de desarrollo de la ingeniería de Software formulada por Kent Beck, debido a que el enfoque fue desarrollado utilizando buenas prácticas reconocidas, como el desarrollo iterativo, y con la participación del cliente en niveles extremos.

En la programación extrema, todos los requerimientos se expresan como escenarios (llamados historias de usuario), los cuales se implementan directamente como una serie de tareas.

Los programadores trabajan en parejas y desarrollan pruebas para cada tarea antes de escribir el código. Todas las pruebas se deben ejecutar satisfactoriamente cuando el código nuevo se integre al sistema. Existe un pequeño espacio de tiempo entre las entregas del sistema (Sommerville, 2005).

#### **2.8.1.1. Características de la metodología XP**

Las características principales de la metodología XP, según los autores (Bustamante & Rodriguez, 2014) son:

- Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.
- Se aplica de manera dinámica durante el ciclo de vida del Software.
- Es capaz de adaptarse a los cambios de requisitos.

- Los individuos e interacciones son más importantes que los procesos y herramientas.
- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.

La gente es el principal factor de éxito de un proyecto Software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.

- Software que funcione es más importante que documentación exhaustiva.
- Desarrollar Software que funciona más que conseguir una buena documentación.

La regla a seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental.

- La colaboración con el cliente es más importante que la negociación de contratos.
- La colaboración con el cliente más que la negociación de un contrato.

Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

- La respuesta ante el cambio es más importante que el seguimiento de un plan.

#### **2.8.1.2. Valores de la metodología XP**

Los Valores originales de la programación extrema son: simplicidad, comunicación, retroalimentación (feedback) y coraje. Un quinto valor, respeto, fue añadido en la segunda edición de Extreme Programming Explained. Los cinco valores se detallan a continuación:

- Simplicidad:

La simplicidad es la base de la programación extrema. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento.

Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hacen que la complejidad aumente exponencialmente.

Para mantener la simplicidad es necesaria la refactorización del código, ésta es la manera de mantener el código simple a medida que crece.

También se aplica la simplicidad en la documentación, de esta manera el código debe comentarse en su justa medida, intentando eso sí que el código esté autodocumentado. Para ello se deben elegir adecuadamente los nombres de las variables, métodos y clases. Los nombres largos no decrementan la eficiencia del código ni el tiempo de desarrollo gracias a las herramientas de autocompletado y refactorización que existen actualmente.

Aplicando la simplicidad junto con la autoría colectiva del código y la programación por parejas se asegura que cuanto más grande se haga el proyecto, todo el equipo conocerá más y mejor el sistema completo.

- **Comunicación:**

La comunicación se realiza de diferentes formas. Para los programadores el código comunica mejor cuanto más simple sea.

Si el código es complejo hay que esforzarse para hacerlo inteligible. El código autodocumentado es más fiable que los comentarios ya que éstos últimos pronto quedan desfasados con el código a medida que es modificado.

Debe comentarse sólo aquello que no va a variar, por ejemplo, el objetivo de una clase o la funcionalidad de un método. Las pruebas unitarias son otra forma de comunicación ya que describen el diseño de las clases y los métodos al mostrar ejemplos concretos de cómo utilizar su funcionalidad.

Los programadores se comunican constantemente gracias a la programación por parejas. La comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo. El cliente decide qué características tienen prioridad y siempre debe estar disponible para solucionar dudas.

- Retroalimentación (FEEDBACK):

Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real.

Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a el programador esa centrarse en lo que es más importante. Considérense los problemas que derivan de tener ciclos muy largos.

Meses de trabajo pueden tirarse por la borda debido a cambios en los criterios del cliente o malentendidos por parte del equipo de desarrollo.

El código también es una fuente de retroalimentación gracias a las herramientas de desarrollo. Por ejemplo, las pruebas unitarias informan sobre el estado de salud del código. Ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debidos a cambios recientes en el código.

- Coraje o valentía:

Muchas de las prácticas implican valentía. Una de ellas es siempre diseñar y programar para hoy y no para mañana. Esto es un esfuerzo para evitar empantanarse en el diseño y requerir demasiado tiempo y trabajo para implementar todo lo demás del proyecto. La valentía les permite a los desarrolladores que se sientan cómodos con reconstruir su código cuando sea necesario. Esto significa revisar el sistema existente y modificarlo si con ello los cambios futuros se implementaran más fácilmente. Otro ejemplo de valentía es saber cuándo desechar un código: valentía para quitar código fuente obsoleto, sin importar cuanto esfuerzo y tiempo se invirtió en crear ese código. Además, valentía significa persistencia: un programador puede permanecer sin avanzar en un problema complejo por un día entero, y luego lo resolverá rápidamente al día siguiente, sólo si es persistente (Bustamante & Rodriguez, 2014).

#### 2.8.1.4. Pasos de la metodología XP

Los Pasos fundamentales inmersos en las fases del método son:

- **Desarrollo iterativo e incremental:** Pequeñas mejoras, unas tras otras.
- **Pruebas unitarias continuas:** Son frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- **Programación en parejas:** Se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera el código es revisado y discutido mientras se escribe es más importante que la posible pérdida de productividad inmediata.
- **Frecuente integración del equipo de programación con el cliente o usuario:** Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Corrección de todos los errores antes de añadir nueva funcionalidad:** Hacer entregas frecuentes.
- **Refactorización del código:** Es decir, reescribir ciertas partes del código para aumentar su legibilidad y Mantenibilidad, pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- **Propiedad del código compartido:** en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- **Simplicidad del código:** es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias.

Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Cuanto más simple es el sistema, menos tendrá que comunicar sobre éste, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores (Bustamante & Rodriguez, 2014).

#### **2.8.1.5. Fases de la metodología XP**

Según los autores (Bustamante & Rodriguez, 2014) clasifican en cuatro fases a la metodología XP para el desarrollo de software.

##### **Fase I: Planificación del proyecto**

- *Historias de usuario:* El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso, pero con algunas diferencias: Constan de 3 o 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.
- *Release Planning:* Después de tener ya definidas las historias de usuario es necesario crear un plan de publicaciones, en inglés "Release plan", donde se indiquen las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicarán estas versiones. Un "Release plan" es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa. Después de un "Release plan" tienen que estar claros

estos cuatro factores: los objetivos que se deben cumplir (que son principalmente las historias que se deben desarrollar en cada versión), el tiempo que tardarán en desarrollarse y publicarse las versiones del programa, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado. (\*Release plan: Planificación de publicaciones).

- *Iteraciones:* Todo proyecto que siga la metodología X.P. se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores.
- *La Velocidad del Proyecto:* Es una medida que representa la rapidez con la que se desarrolla el proyecto; estimarla es muy sencillo, basta con contar el número de historias de usuario que se pueden implementar en una iteración; de esta forma, se sabrá el cupo de historias que se pueden desarrollar en las distintas iteraciones. Usando la velocidad del proyecto controlaremos que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración. Es conveniente reevaluar esta medida cada 3 o 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un nuevo "Reléase Plan".
- *Programación en Parejas:* La metodología XP aconseja la programación en parejas pues incrementa la productividad y la calidad del Software desarrollado. El trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. De esta forma se consigue un código y diseño con gran calidad. Reuniones Diarias: Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto.

## **Fase II: Diseño**

- *Diseños Simples:* La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar.
- *Glosarios de Términos:* Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.
- *Riesgos:* Si surgen problemas potenciales durante el diseño, XP sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.
- *Funcionabilidad extra:* Nunca se debe añadir funcionalidad extra al programa, aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.
- *Re factorizar:* Re factorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. re factorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común rehusar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos.

## **Fase III: Codificación**

Como ya se dijo en la introducción, el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada

cumple la funcionalidad especificada. La codificación debe hacerse ateniendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

#### **Fase IV: Pruebas**

Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando. El uso de los test en XP es el siguiente:

- Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.
- Hay que someter a test las distintas clases del sistema omitiendo los métodos más triviales.
- Se deben crear los test que pasarán los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.
- Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará.
- Como se comentó anteriormente los distintos test se deben subir al repositorio de código acompañados del código que verifican.
- Test de aceptación. Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario.
- Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos.

**Figura 2.7.** Fases de la Metodología XP



*Fuente:* (Bustamante & Rodriguez, 2014).

## 2.8.2. Metodología Kanban

### 2.8.2.1. Breve historia de Kanban

La palabra japonesa Kanban, que se traduce como “letrero”, se ha convertido en sinónimo de programación de la demanda. Kanban tiene sus raíces en los primeros días del sistema de producción de Toyota. A fines de la década de 1940 y principios de la década de 1950, Taiichi Onho desarrollo kanbans para controlar la producción entre procesos e implementar la fabricación justo a tiempo (JIT) en las plantas de fabricación de Toyota en Japón.

Originalmente, Toyota uso Kanban para reducir costos y administrar la utilización de la máquina. Sin embargo, en la actualidad, Toyota continúa utilizando el sistema no solo para administrar costos y flujos, sino también para identificar impedimentos para fluir y oportunidades para la mejora continua. [Gross y McInnis, 2003]

### 2.8.2.2. Definición de Kanban

el Kanban es un sistema de gestión del trabajo en curso (WIP), que sirve principalmente para asegurar una producción continua y sin sobrecargas en el equipo de producción multimedia. El Kanban es un sistema de gestión donde se produce exactamente aquella cantidad de trabajo que el sistema es capaz de asumir.

El Kanban es un sistema de trabajo **just in time**, lo que significa que evita sobrantes innecesarios de stock, que en la gestión de proyectos multimedia equivale a la inversión innecesaria de tiempo y esfuerzo en lo que no necesitaremos (o simplemente es menos prioritario) y evita sobrecargar al equipo.

En la gestión del trabajo en curso con Kanban, se busca un concepto clave como es limitar el trabajo en curso. Está demostrado que, cuanto más trabajo en curso se gestione a la vez, los índices de calidad disminuyen drásticamente.

En la producción de proyectos multimedia, aumentar el trabajo en curso implica aumentar la cantidad de errores que este proyecto multimedia tendrá como consecuencia de la poca capacidad de concentración que los desarrolladores podrán dedicarles a las tareas.

Limitar el trabajo en curso mediante la gestión del trabajo con Kanban también tiene una consecuencia importante y es que disminuimos el tiempo de servicio de una tarea desde que entra al sistema hasta que sale. Disminuyendo la cantidad de trabajo en curso, conseguimos que el enfoque en cada una de las tareas sea mayor y que el tiempo dedicado a todas ellas, sumado, sea menor que el empleado en asumirlas todas de golpe. (Bermejo, 2006).

### **2.8.2.3. Aplicación de Kanban**

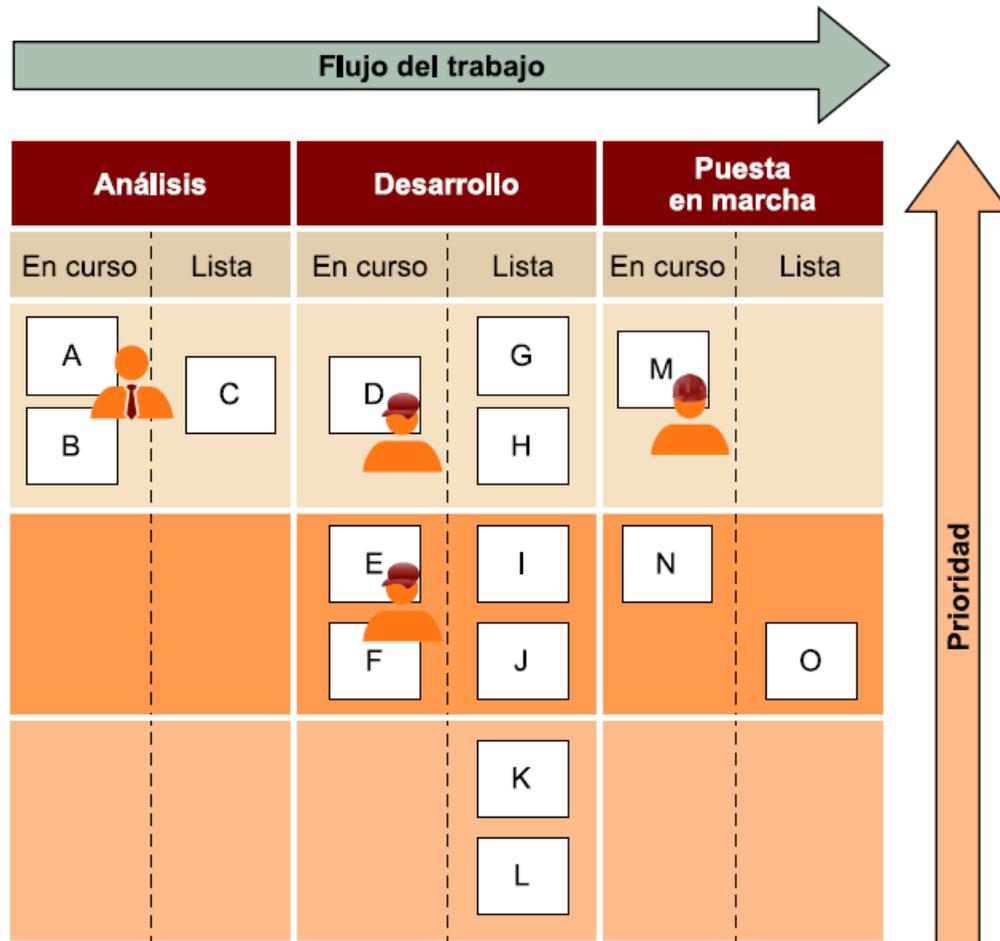
Según el autor (Bermejo, 2006) refleja tres pasos para la aplicación de Kanban, las cuales son:

- *Primer paso*

La forma más habitual de aplicar el Kanban en la producción de productos multimedia es mediante el visual management, es decir, la representación visual del flujo de trabajo mediante paneles que tienen que reflejar la realidad del equipo en cada instante.

Para ello se refleja en el panel Kanban las fases, tareas y el equipo.

**Figura 2.8.** Panel de Kanban



**Fuente:** (Bermejo, 2006)

- *Un panel Kanban para gestionarlos a todos*

En la gestión del trabajo en curso con Kanban, no existe un único modelo de panel adecuado para todos los equipos ni que cumpla todas las necesidades de la empresa. Un error frecuente de aquellos que empiezan a implantar Kanban en el equipo es intentar adoptar las fases de un modelo de panel externo en la realidad de la producción de su equipo. No se trata de cambiar las fases por otras, sino de estudiarlas, comprenderlas y hacerlas visibles. Cuando adoptamos Kanban, el panel tiene que ser construido y mejorado constantemente.

En la gestión ágil de proyectos, una de las claves fundamentales es la iteración constante y la mejora a cada iteración. Como buenos agilistas, nuestro panel Kanban tiene que ser estudiado y juzgado en cada iteración para detectar fases que nos falten o nos sobren.

- *Leyendo un panel Kanban*

En el panel de ejemplo, tenemos tareas que influyen de izquierda a derecha de la siguiente manera:

- Los dos programadores están actualmente produciendo la funcionalidad de D, E y F en simultaneo.
- Mientras tanto, el analista-programador ha hecho el análisis y la estimación de algunas tareas nuevas que les han encomendado, esta estimación es muy importante, puesto que en ella se basa el presupuesto que la empresa hace de las tareas que el equipo lleva a cabo.
- A medida que el tiempo va pasando, los dos programadores van desarrollando a buen ritmo, tienen acabados seis módulos y están programando otros.
- Solo una tarea ha sido registrada como finalizada en este mes (la tarea O) y, por lo tanto, solo un cliente ha recibido su petición con éxito.

## **2.9. MÉTRICAS DE CALIDAD DEL SOFTWARE**

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

Se hará la medición de calidad de software mediante la 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 distintas partes que establece cualquier componente de calidad de software puede ser descrito en términos 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.

**Figura 2.9.** Características de la calidad del producto de software



*Fuente:* (ISO 25000, 2022)

### 2.9.2. Implementación de la ISO/IEC 25010

La implementación de la norma según los autores (Francisco & M., 2021), consideran las siguientes características para la implementación de la norma ISO/IEC 25010 como:

- **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.

- ✓ **Número de entrada de usuarios:** Se cuenta cada entrada de usuario que proporciona datos al sistema.
- ✓ **Número 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.

- ✓ **Número 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.
- ✓ **Número de interfaces externas:** Se cuenta las interfaces legibles por la máquina.

Para calcular el punto de función se utilizará la siguiente formula:

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

Dónde:

**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  $i=14$

- **Fiabilidad**

La fiabilidad es la capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados, donde se considera la fiabilidad de cada módulo o subsistema de forma independiente.

Para calcular la fiabilidad se usa la fórmula:

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

Dónde:

**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.

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.

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

Dónde:

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

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:

$$R_p = \frac{\sum_{i=2}^5 (R_i)}{\sum_{i=2}^5 P_i} = \frac{0,95 + 0,99 + 0,97 + 0,99}{4} = \frac{3,9}{4} = 0,98$$

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

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

De lo cual se puede decir que existe un 8 % 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.

- **Usabilidad**

Capacidad del producto Software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

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

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

Dónde:

**Xi:** Es la sumatoria de valores

**n:** Es el número de preguntas

- **Mantenibilidad.**

Esta característica representa la capacidad del producto Software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas (ISO 25000, 2022).

El índice de madurez se calcula con la siguiente formula:

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

Dónde:

**$M_t$ :** 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.

- **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, está 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.

- **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 procede a calcular la calidad global del sistema.

## **2.10. ESTIMACIÓN DE COSTOS**

### **2.10.1. COCOMO II**

Según el autor [Boehm, 1981], introdujo una jerarquía de modelos de estimación de Software que llevan el nombre COCOMO, por CONstructive COst MOdel: modelo constructivo de costos. El modelo COCOMO original se convirtió en uno de los modelos de estimación de costo más ampliamente utilizados y estudiados en la industria.

La primera versión del modelo COCOMO, COCOMO 81, fue un modelo de tres niveles donde estos reflejaban el detalle del análisis de la estimación del coste. El primer nivel (básico) provee una estimación inicial burda, el segundo nivel la modifica utilizando una serie de multiplicadores de proyecto y proceso, y el nivel más detallado produce estimaciones para las diferentes fases del proyecto. [Sommerville, 2005]

El modelo COCOMO es seguramente el más conocido, ya que como subraya Conte et al., es el “más completo y detalladamente documentado de todos los modelos de esfuerzo”. Este modelo se basa en una estimación previa del tamaño del software en líneas de código (LDC). Este dato sirve como parámetro de las ecuaciones de cálculo de esfuerzo derivadas por B.W. Boehm (Boehm, 1981).

### 2.10.2. Método de Estimación de Costos Nivel Intermedio

Según el autor (Boehm, 1981), COCOMO II es en realidad una jerarquía de modelos de estimación que incluye submodelo básico, intermedio, y avanzado.

- 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.

$$\text{ESFUERZO: } E = a KLDC^b$$

$$\text{TIEMPO: } D = c E^d$$

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.

Formula de esfuerzo:

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

Dónde:

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

**E**, es el esfuerzo aplicado en personas-mes,

**KLDC**, es el número estimado de Líneas de Código expresadas en miles distribuidas para el proyecto.

$\prod_{i=1}^{n=15} M(x_i)$ , representa la productoria de los 15 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  $\prod_{i=1}^{n=15} M(x_i)$ , en cada fase del desarrollo.

Además, existen tres tipos de proyectos en COCOMO, los cuales son:

- **Orgánicos:** relativamente pequeños y sencillos, en los que trabajan pequeños equipos con experiencia, sobre un conjunto de requisitos poco rígidos.
- **Semi acoplados:** proyectos intermedios (en tamaño y complejidad) en los que participan equipos con variados niveles de experiencia, y que deben satisfacer requisitos poco o medio rígidos.
- **Empotrados:** proyectos que deben ser desarrollados en un conjunto de hardware, software y restricciones operativas muy restringido.

**Tabla 2.1.** Valores constantes COCOMO Intermedio

<b>PROYECTO</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>
ORGANICO	3.2	1.05	2.5	0.38
SEMIACOPLADO	3.0	1.12	2.5	0.35
EMPOTRADO	2.8	1.20	2.5	0.32

**Fuente:** (Fenton, 1991).

### 2.10.3. Características de COCOMO

El modelo COCOMO intermedio calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de conductores de costo, que

incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto (Boehm, 1981).

- **Factores de costos:** En el modelo intermedio de COCOMO, el valor de esfuerzo nominal calculado con las ecuaciones anteriores debe ser ajustado de acuerdo a los factores de coste del modelo, que se pueden observar en la tabla 2.2. Cuando un factor se valora como nominal o medio, el valor asignado es siempre 1, es decir, no influye en el coste.

**Tabla 2.2.** Factores de costos

Factores (cost-drivers)	Valor de los factores					
	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra
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
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria			1,00	1,06	1,21	1,56
Volatilidad del hardware		0,87	1,00	1,15	1,30	
Restricciones de tiempo de respuesta		0,87	1,00	1,07	1,15	
Calidad de los analistas	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	1,07	1,00	0,95		
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
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	

**Fuente:** (Fenton, 1991).

#### 2.10.4. Fórmulas de Estimación

##### ➤ Estimación de esfuerzo

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

Dónde:

$a$  y  $b$ , representan los coeficientes COCOMO nivel intermedio

$E$ , es el esfuerzo aplicado en personas-mes,

$KLDC$ , es el número estimado de Líneas de Código expresadas en miles distribuidas para el proyecto.

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

##### ➤ Estimación del tiempo

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

Dónde:

$c$  y  $d$ , representa los coeficientes COCOMO nivel intermedio

$E$ , representa el esfuerzo.

##### ➤ Otras estimaciones

#### Número de personas requerido para el desarrollo

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

Dónde:

E, es el esfuerzo, T es el tiempo.

### Costo total del proyecto

$$CT = NP * T * Sueldo\ mes \quad Bs/\$us$$

Dónde:

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

### Costo por línea de código

$$LDC = \frac{CT}{KLDC} \quad Bs/\$us$$

Dónde:

CT, es el costo total del proyecto

KLDC, líneas de código expresadas en miles

**Tabla 2.3.** Estimación de Costos

FACTOR	ECUACION INTERMEDIO
ESFUERZO (E)	$E = a * (KLDC)^b * ME$
TIEMPO (T)	$T = c * (E)^d$
PERSONAS (P)	$P = E/T$

*Fuente: (Elaboración Propia).*

## 2.11. SEGURIDAD

### 2.11.1. Técnicas de seguridad

- **Niveles de seguridad en el acceso al equipo y el servidor**

La seguridad en el equipo del administrador es un tema importante ya que es donde se ejecuta el software, así mismo la seguridad en el servidor donde se almacena los datos registrados.

- Seguridad en el equipo del usuario (contraseña segura para acceder al equipo).
- Seguridad del servidor web para los datos que se guardan en el mismo.
- Seguridad de la información en relación al servidor y al usuario.

- **Seguridad del software**

La seguridad del software es muy importante que se debe reflejar en todo el proceso del desarrollo del mismo.

- Uso de password

Una contraseña o clave (en inglés password) es una forma de autenticar al acceso de la información secreta para el control del mismo. Por tanto, la contraseña debe ser único y directo a aquellos que tengan el permiso para acceder a la información.

- *Cifrado de datos con Laravel*

El cifrado de Laravel usa OpenSSL para proporcionar cifrado AES-256 y AES-128.

Todos los valores encriptados de Laravel están firmados usando un código de autenticación de mensajes (MAC) para que su valor subyacente no se pueda modificar una vez encriptados.

La encriptación o cifrado de datos protege información secreta de la empresa, la misma se realiza mediante el mecanismo de transformar una cadena en otra. De esta forma se obtiene una cadena encriptada, donde se muestra una cadena con diferentes valores a la original.

- Hash

La fachada de Laravel hash proporciona hash Bcrypt y Argon2 seguro para almacenar contraseñas de usuario.

## **2.12. PRUEBAS AL SOFTWARE**

La fase de pruebas es indispensable en todo proyecto de Software. Ningún producto de Software con un mínimo grado de complejidad está exento de defectos que, al menos en potencia, generan fallas. No es viable entregar un producto sin haber

pasado por un proceso de prueba que proporcione los niveles de confianza necesarios para poder usar el sistema construido. [Cervantes, Gómez, 2019]

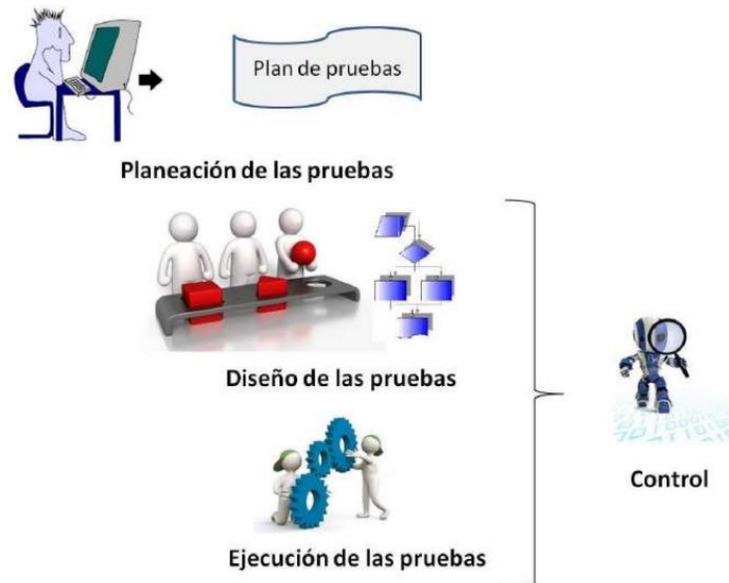
### **2.12.1. Proceso de Pruebas**

El proceso de pruebas puede verse como parte del proceso de desarrollo de Software, o como un proceso aparte. Este proceso, es una actividad compleja que requiere de una planeación. La planeación se documenta en el plan de pruebas, en este documento se especifican las técnicas y métodos a utilizar, así como el ambiente, herramientas necesarias y el calendario previsto.

Durante el diseño de las pruebas se identifican los elementos que deben ser probados. Se especifican las condiciones y datos de prueba en función de los elementos a probar. Se diseña la estructura de las pruebas y el ambiente que requieren. Las pruebas se estructuran, se organizan por grupos y categorías, y se les asignan prioridades según un análisis de riesgos.

La ejecución de las pruebas se lleva a cabo siguiendo un protocolo de pruebas que consta de un procedimiento de prueba y una lista de casos de prueba por ejecutarse. El resultado de las pruebas se reporta en una bitácora de pruebas, conocida como Test Log. Y cuando se encuentra una falla se genera un reporte de falla. Durante el diseño y ejecución de las pruebas se lleva a cabo un control que tiene como objetivos: monitorear el progreso, medir y analizar los resultados, realizar acciones correctivas y tomar decisiones estratégicas (Gómez, Cervantes, & Gonzales, 2019).

**Figura 2.10.** El Proceso de Pruebas



**Fuente:** (Gómez, Cervantes, 2019)

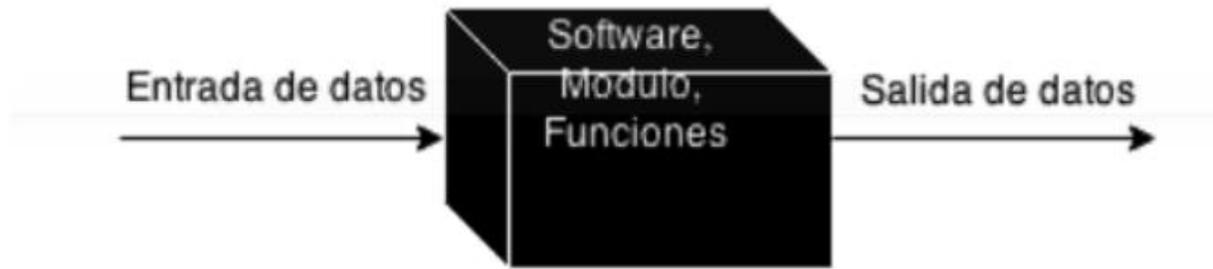
## 2.12.2. Tipos de prueba a usar

### 2.12.2.1. Caja negra

Las pruebas de caja negra son aquellas en las que el producto es visto como una caja sin luz en donde no es posible ver la forma en la que éste lleva a cabo sus funciones, lo único visible son sus interacciones con lo exterior, por lo que lo solo se consideran las entradas y las salidas del Software que se va a probar. Las pruebas de caja negra, o funcionales tienen el punto de vista del usuario porque toman en cuenta los requerimientos, es decir, se prueba que la función requerida se ejecute correctamente, sin importar cómo está implementada.

El propósito de una prueba funcional es buscar discrepancias con la especificación de requerimientos y no demostrar que el programa cumple con ésta cabalmente (Myers, 2004). Para esto se deben considerar condiciones inválidas y entradas inesperadas al momento de diseñarla (Gómez, Cervantes, & Gonzales, 2019).

**Figura 2.11.** Ejemplo de caja negra



**Fuente:** (Maldonado, Alva, & Santa Cruz, 2021).

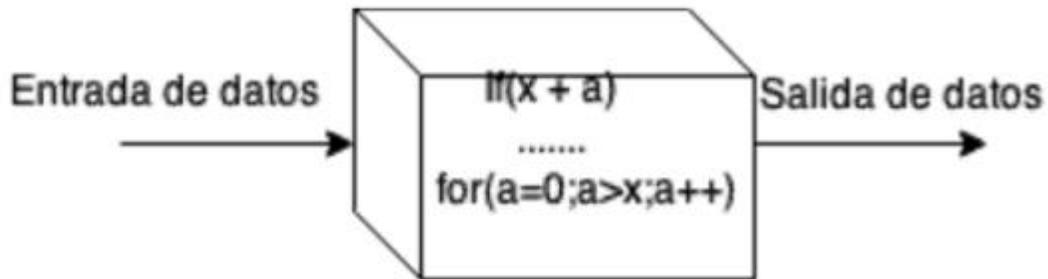
Según los autores (Maldonado, Alva, & Santa Cruz, 2021) mencionan que con los métodos de caja negra se intentan encontrar los errores:

- Funciones incorrectas y faltantes.
- Errores de inicialización y terminación.
- Errores de interfaz.
- Errores en la estructura.

#### **2.12.2.2. Caja blanca**

Las pruebas de caja blanca son aquellas en donde todos los detalles y la forma en la que funciona el producto están disponibles para el diseñador de pruebas, así que puede diseñar casos de prueba para cada uno de estos detalles. El objetivo de las pruebas de caja blanca es verificar que funcionen diferentes casos que puede haber en la ejecución de un programa. Es importante mencionar que en ocasiones es imposible probar absolutamente todos los casos posibles (Gómez, Cervantes, & Gonzales, 2019).

**Figura 2.12.** Ejemplo de caja negra



**Fuente:** (Maldonado, Alva, & Santa Cruz, 2021).

Según los autores (Maldonado, Alva, & Santa Cruz, 2021) mencionan que los métodos de pruebas de caja blanca aportan los siguientes puntos:

- Garantizan que todas las rutas del código se revisan al menos una vez.
- Revisan las condiciones lógicas.
- Revisan las estructuras de datos.

### **2.12.2.3. Pruebas de Estrés**

Con las pruebas de estrés se somete el sistema a cargas pesadas de datos o de actividad en un tiempo corto. Enlistamos también ejemplos de otros tipos de prueba: de seguridad, de usabilidad, de almacenamiento, de configuración, de compatibilidad, de instalación, de confiabilidad, y de recuperación a fallas (Gómez, Cervantes, & Gonzales, 2019).

## **2.13. HERRAMIENTAS**

### **2.13.1. Hardware**

El equipo que cuenta la empresa para el manejo del sistema orientado a la web para la gestión de información cumple con las siguientes características las cuales son:

- Procesador: Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz 3.20 GHz
- Memoria instalada (RAM): 16,0 GB
- Tipo de sistema: sistema operativo de 64 bits, procesador x64

- Disco duro(HDD): 1TB.
- Sistema operativo de Windows 10 Pro

## **2.13.2. Software**

### **2.13.2.1. Lenguajes de programación**

- ***PHP 7***

PHP es un acrónimo recursivo para “PHP: Hypertext Preprocessor”, originalmente Personal Home Page, es un lenguaje interpretado libre, usado originalmente solamente para el desarrollo de aplicaciones presentes y que actuaran en el lado del servidor, capaces de generar contenido dinámico en la World Wide Web. Figura entre los primeros lenguajes posibles para la inserción en documentos HTML, dispensando en muchos casos de archivos externos para eventuales procedimientos de datos. El código es interpretado en el lado del servidor por el módulo PHP, que también genera la página web para ser visualizada en el lado del cliente. El lenguaje evolucionó, pasó a ofrecer funcionalidades en la línea de comandos, y además ganó características adicionales, que posibilitaron usos adicionales del PHP. Es posible instalar el PHP en la mayoría de los sistemas operativos, totalmente de manera gratuita. Siendo competidor directo de la tecnología ASP perteneciente a Microsoft, PHP es utilizado en aplicaciones como Media Wiki, Facebook, Drupal, Joomla, WordPress, Magento y Oscommerce.

El lenguaje surgió a mediados de 1994, como un paquete de programas CGI creados por Rasmus Lerdorf, con el nombre Personal Home Tools, para sustituir un conjunto de scripts Perl que este usaba en el desarrollo de su página personal. En 1997 fue lanzado el nuevo paquete del lenguaje con el nombre de PHP/FI, trayendo la herramienta Forms Interpreter, un interpretador de comandos SQL. Más tarde, Zeev Suraski desarrolló el analizador de PHP 3 que contaba con el primer recurso orientado a objetos, que daba poder de alcanzar algunos paquetes, tenía herencia y daba a los desarrolladores solamente la posibilidad de implementar propiedades y métodos (Arias, 2017).

- **JavaScript**

JavaScript, comúnmente abreviado como JS, es un lenguaje de programación del tipo interpretado. Está basado en el estándar ECMAScript y, a pesar de ser débilmente tipado y dinámico, también se define como un lenguaje orientado a objetos.

JavaScript nació el año 1995, prácticamente de la mano de la WWW, o el Internet comercial. Fue diseñado en un principio por Netscape Communications (hoy, Mozilla Foundation). Integrado originalmente como una especie de plugin en los navegadores de la primera era web, fue muy criticado por su lentitud para procesar código, dado que esta acción generaba un retardo notable en la carga completa de una página. A los pocos años de vida fue destronado por Flash Player, pero tan solo una década después, volvió completamente recargado, y recupero su lugar y labor dentro del desarrollo de sitios web (Luna, 2019).

- **JQuery**

JQuery es una librería de JavaScript muy popular entre los programadores web que permite simplificar significativamente el código (Recio, 2016).

El enfoque de jQuery no consiste solo en una codificación más intuitiva y concisa de los scripts, sino que su filosofía permite concentrarse en el conjunto de elementos que gestiona el DOM. El JavaScript tradicional, en su evolución histórica, se ha tenido que acoplar al Document Object Model. Con jQuery, John Resing ha reconsiderado de alguna forma el JavaScript manteniéndolo como un verdadero lenguaje de programación, basado en el DOM (Aubry & Van, 2017).

- **HTML – 5**

El HyperText Markup Lenguaje (HTML) es un lenguaje de programación. A diferencia de otros lenguajes, no está compuesto por instrucciones, sino por un conjunto de etiquetas que organizan y declaran el propósito de cada contenido del documento.

En el sentido estricto, HTML es un texto escrito con una sintaxis particular que el navegador es capaz de leer y aplicar. Se trata de un lenguaje que fue

creado para compartir a través de Internet no solo el texto incluido en los documentos sino también su formato (Gauchat, 2013).

El lenguaje HTML debe funcionar en diferentes plataformas y navegadores, ha sido desarrollado con la premisa de que cualquier tipo de dispositivo debería ser capaz de usar información de la Web (ordenadores, teléfonos móviles, PDA's, etc.), para ello ha sido necesario un gran consenso entre los interlocutores de la industria cada vez aparecía una nueva versión de HTML, para que así cada nueva versión sea el estándar que sea implementado por los diferentes navegadores Web (Casado, 2019).

### 2.13.3. Framework

- **Laravel 8**

Laravel es el framework PHP más utilizado del momento. Su filosofía es crear código simple y elegante.

En un principio, Laravel no fue creado con el patrón de arquitectura MVC, y su foco estaba puesto principalmente en resolver problemas de autenticación. No obstante, la primera versión incorporaba funcionalidades que fueron bien recibidas, y de forma rápida, por la comunidad de desarrolladores.

La segunda versión tardó menos de seis meses en salir al mercado. Laravel terminó de adoptar el patrón MVC para su arquitectura e incorporó el siguiente slogan:

*“liberándote del código espagueti, Laravel te ayuda a crear aplicaciones maravillosas usando una sintaxis simple y expresiva. El desarrollo debe ser una experiencia creativa que disfrutes, no algo que sea doloroso. Disfruta del aire fresco.”* (Cíceri, 2018).

- **Angular**

Angular es una plataforma que permite desarrollar aplicaciones web en la sección cliente utilizando HTML y JavaScript para que el cliente asuma la mayor parte de la lógica y descargue al servidor con la finalidad de que las aplicaciones ejecutadas a través de Internet sean más rápidas. El hecho de estar mantenido por Google, así como una serie de innumerables razones

técnicas, ha favorecido su rápida adopción por parte de la comunidad de desarrolladores.

Además de mejorar el rendimiento de las aplicaciones web, su utilización en dispositivos móviles esta optimizada ya que, en ellos, los ciclos de CPU y memoria son críticos para su óptimo funcionamiento (Boada & Gómez, 2018).

#### **2.13.4. Base de datos**

- **MySQL 8**

Dentro de las múltiples posibilidades que los sitios dinámicos ofrecen al usuario está la obtención de información almacenada en una base de datos en el servidor.

MySQL es gratuito (aunque también existen licencias comerciales). Por otra parte, los datos se almacenan en archivos que tienen un tamaño mucho menor que otras bases de datos. Además, el motor de datos de MySQL es mucho más rápido, tanto grabando datos como localizándolos y recuperándolos, que el de otras bases de datos. Eso sin contar con que MySQL ofrece una gran seguridad sobre la integridad de los datos almacenados (López, 2014).

#### **2.13.5. Api de Google Maps**

##### **2.13.5.1. Google Maps**

Según la página oficial de (Google Maps, 2018), Google Maps es una aplicación de Google que muestra mapas de todo el planeta e incluso rutas entre diferentes ubicaciones.

- **Funciones principales**

- Navegación: herramientas disponibles para movernos por el mapa.
- Búsquedas: de direcciones, calles, intersecciones y ubicaciones.
- Vistas: modo mapa, satélite, relieve e imágenes 360°.
- Imprimir: los mapas y la información que muestra Google Maps.
- Compartir: difundir la información entre mis contactos u otros dispositivos propios.

- **Vistas del mapa**

- Mapa: una presentación tradicional de carreteras, parques,
- Satélite: imágenes aéreas realizadas por el satélite.
- Relieve: muestra la elevación física, con relieves sombreados y líneas de elevación.

- **API JavaScript de Google Maps**

La API de JavaScript de Maps le permite personalizar mapas con su propio contenido e imágenes para mostrarlos en páginas web y dispositivos móviles. La API de JavaScript de Maps presenta cuatro tipos de mapas básicos (mapa de ruta, satélite, híbrido y terreno) que puede modificar utilizando capas y estilos, controles y eventos, y varios servicios y bibliotecas.

La API de JavaScript de Maps se carga mediante una script etiqueta, que se puede agregar en línea en su archivo HTML o dinámicamente usando un archivo JavaScript separado. Le recomendamos que revise ambos enfoques y elija el que sea más apropiado para la estructura del código de su proyecto.

### **2.13.6. Servicios Api Rest (GET POST PUT DELETE)**

El estilo REST (*Representational State Transfer*) es una forma ligera de crear Servicios Web. El elemento principal en el que se basan estos servicios son las URLs a las que podemos acceder, por ejemplo, mediante el protocolo HTTP, para obtener información o realizar alguna operación. El formato de la información que se intercambie con estas URLs lo decidirá el desarrollador del servicio. Este tipo de servicios acercan los Servicios Web al tipo de arquitectura de la web, siendo especialmente interesantes para su utilización en AJAX (Santamaria, 2013).

#### **2.13.6.1. Fundamentos de REST**

Según el autor (Santamaria, 2013), el término REST proviene de la tesis doctoral de Roy Fielding, publicada en el año 2000, y significa **REpresentational State Transfer**. REST es un conjunto de restricciones que, cuando son aplicadas al diseño de un sistema, crean un estilo arquitectónico de Software. Dicho estilo arquitectónico se caracteriza por:

- Debe ser un sistema cliente-servidor.
- Tiene que ser sin estado, es decir, no hay necesidad de que los servicios guarden las sesiones de los usuarios (cada petición al servicio tiene que ser independiente de las demás).
- Debe soportar un sistema de cachés: la infraestructura de la red debería soportar caché en diferentes niveles
- Debe ser un sistema uniformemente accesible (con una interfaz uniforme): cada recurso debe tener una única dirección y un punto válido de acceso. Los recursos se identifican con URIs, lo cual proporciona un espacio de direccionamiento global para el descubrimiento del servicio y de los recursos.
- Tiene que ser un sistema por capas: por lo tanto, debe soportar escalabilidad
- Debe utilizar mensajes auto descriptivos: los recursos se desacoplan de su representación de forma que puedan ser accedidos en una variedad de formatos, como por ejemplo XML, HTML, texto plano, PDF, JPEG, JSON, etc. Estas restricciones no dictan qué tipo de tecnología utilizar; solamente definen cómo se transfieren los datos entre componentes y qué beneficios se obtienen siguiendo estas restricciones. Por lo tanto, un sistema RESTful puede implementarse en cualquier arquitectura de la red disponible. Y lo que es más importante, no es necesario "inventar" nuevas tecnologías o protocolos de red: podemos utilizar las infraestructuras de red existentes, tales como la Web, para crear arquitecturas RESTful.

- **Recursos**

Un recurso REST es cualquier cosa que sea direccionales a través de la Web. Por direccional nos referimos a recursos que puedan ser accedidos y transferidos entre clientes y servidores. Por lo tanto, un recurso es una correspondencia lógica y temporal con un concepto en el dominio del problema para el cual estamos implementando una solución.

- **Representación**

La representación de los recursos es lo que se envía entre los servidores y clientes. Una representación muestra el estado del dato real almacenado en

algún dispositivo de almacenamiento en el momento de la petición. En términos generales, es un *stream binario*, juntamente con los metadatos que describen como dicho *stream* debe ser consumido por el cliente y/o servidor (los metadatos también pueden contener información extra sobre el recurso, como por ejemplo información de validación y encriptación, o código extra para ser ejecutado dinámicamente).

A través del ciclo de vida de un servicio web, puede haber varios clientes solicitando recursos. Clientes diferentes son capaces de consumir diferentes representaciones del mismo recurso. Por lo tanto, una representación puede tener varias formas, como, por ejemplo, una imagen, un texto, un fichero XML, o un fichero JSON, pero tienen que estar disponibles en la misma URL.

- **Uri**

Una URI, o ***Uniform Resource Identifier***, en un servicio web RESTful es un hiper-enlace a un recurso, y es la única forma de intercambiar representaciones entre clientes y servidores. Un servicio web RESTful expone un conjunto de recursos que identifican los objetivos de la interacción con sus clientes.

El conjunto de restricciones REST no impone que la URLs deban ser hiper-enlaces. Simplemente hablamos de hiper-enlaces porque estamos utilizando la Web para crear servicios web. Si estuviésemos utilizando un conjunto diferente de tecnologías soportadas, una URI RESTful podría ser algo completamente diferente. Sin embargo, la idea de direccionabilidad debe permanecer.

Sin las restricciones REST, los recursos se acceden por su localización: las direcciones web típicas son URLs fijas. Si por ejemplo renombramos un fichero en el servidor, la URI será diferente; si movemos el fichero a un directorio diferente, la URI también será diferente.

## 2.13.6.2. Uniformidad de las interfaces a través de peticiones http

- **Tipo de peticiones HTTP**

Estos coinciden con los verbos de HTTP y son: POST, GET, PUT y DELETE que hacen referencia a las operaciones conocidas como CRUD, respectivamente, create, read, update y delete. A continuación, pasamos a hablar de ellos con más detalle.

- ✓ **Get/Retrieve**

Es el método correspondiente a la operación read, y se encarga de leer un recurso existente. Cuando se lee un recurso este no se modifica por lo que GET es considerado un método seguro. Es además idempotente, ya que siempre que realices GET sobre un mismo recurso el resultado será el mismo.

- ✓ **Post/Create**

Es el método correspondiente a la operación create, y se encarga de crear un nuevo recurso. Estos nuevos recursos creados son subordinados, se crean a partir de un recurso padre.

El método POST no es idempotente, lo cual quiere decir que si aplicas varias veces la misma operación al mismo elemento no vas a obtener siempre el mismo resultado.

- ✓ **Put/Update**

Es el método correspondiente a la operación update, y se encarga de actualizar un recurso. Puede igualmente ser usado para la creación de un recurso, llevando a la duda de cuando usar PUT y cuando POST.

PUT se usa para crear recursos cuando tienes toda la información del recurso que te quieres crear, por lo que no tendrías que crearte el recurso partiendo de un recurso padre como ocurre en POST. Al tratarse de un método que modifica un recurso se considera como no seguro. Sin embargo, a diferencia de POST, PUT si es un método idempotente, ya que siempre que se aplica PUT de la misma forma sobre un mismo recurso obtienes los mismos resultados.

✓ **Delete/Delete**

Es el método correspondiente a la operación delete y se encarga de borrar un determinado recurso. Se necesitará saber el identificador del recurso para poder realizar esta operación, ya que ha de ser realizada sobre el recurso concreto.

La operación de DELETE es idempotente, ya que realizar la misma llamada de DELETE sobre el mismo recurso dará siempre el mismo resultado, la eliminación del mismo, que el resultado ya no exista. DELETE es a su vez un método no seguro, ya que modifica el valor de un recurso, concretamente eliminándolo.

# **CAPÍTULO III**

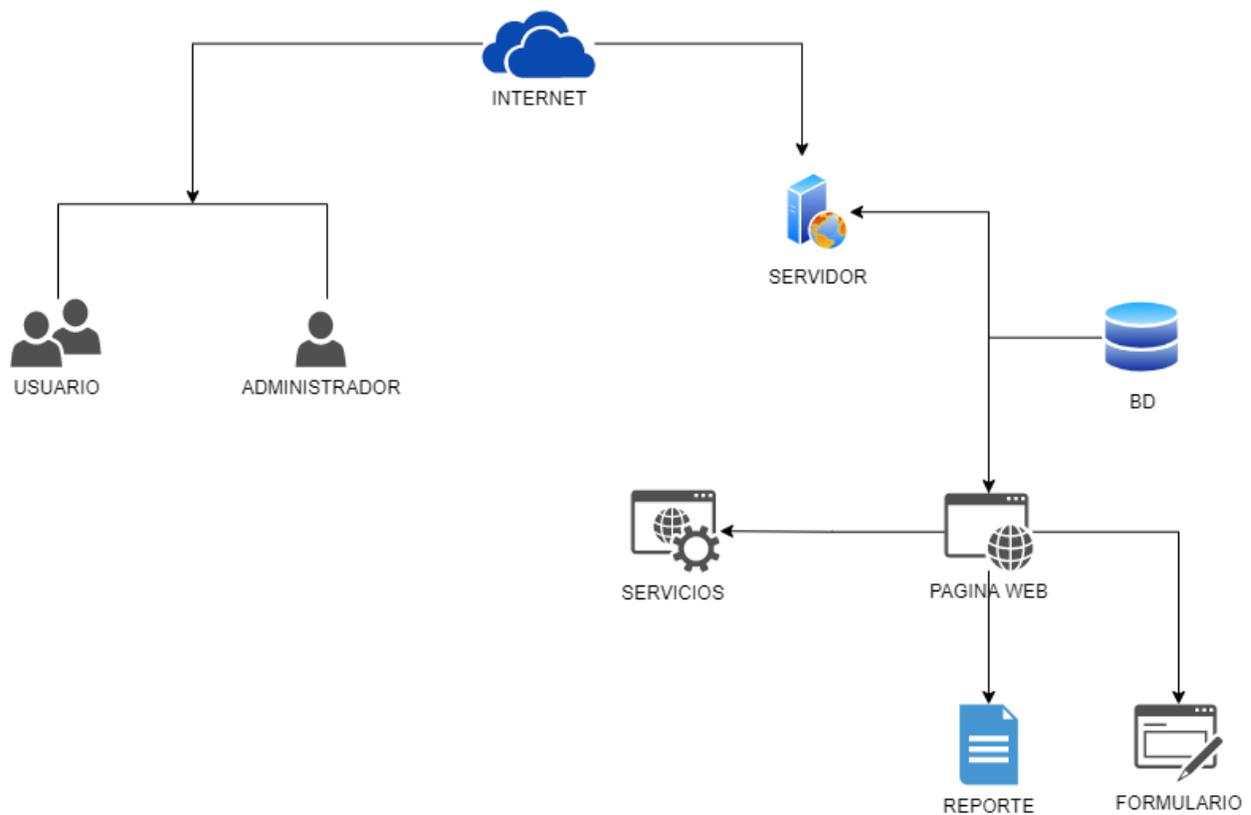
## **MARCO APLICATIVO**

### 3.1. INTRODUCCIÓN

En el presente capítulo se observa en gran magnitud lo que conlleva el análisis, diseño e implementación del sistema web de información geográfica orientada al marketing para las empresas en convenio, para la empresa “FaceBol S.R.L.”, para lo cual se utilizará la metodología ágil de programación extrema XP y para la planificación la metodología Kanban.

### 3.2. ESTRUCTURA O ESQUEMA DEL SISTEMA

Figura 3.1. Arquitectura del sistema

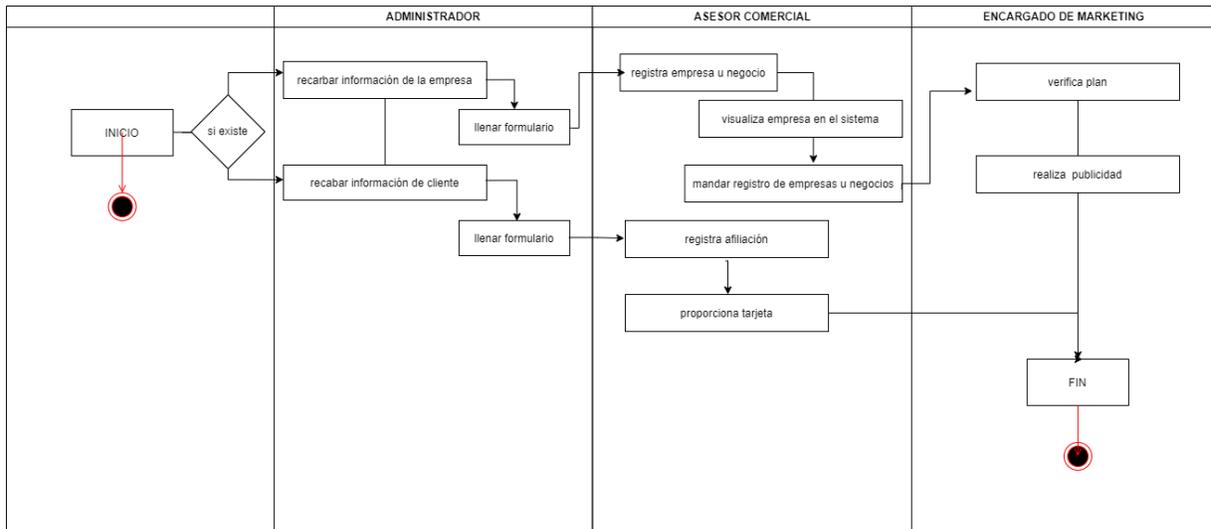


**Fuente:** (Elaboración propia)

### 3.2.1. Funcionalidad del sistema

#### 3.2.1.1. Análisis de la situación actual de la empresa

Figura 3.2. Diagrama de procesos o diagrama de actividades



*Fuente: (Elaboración Propia)*

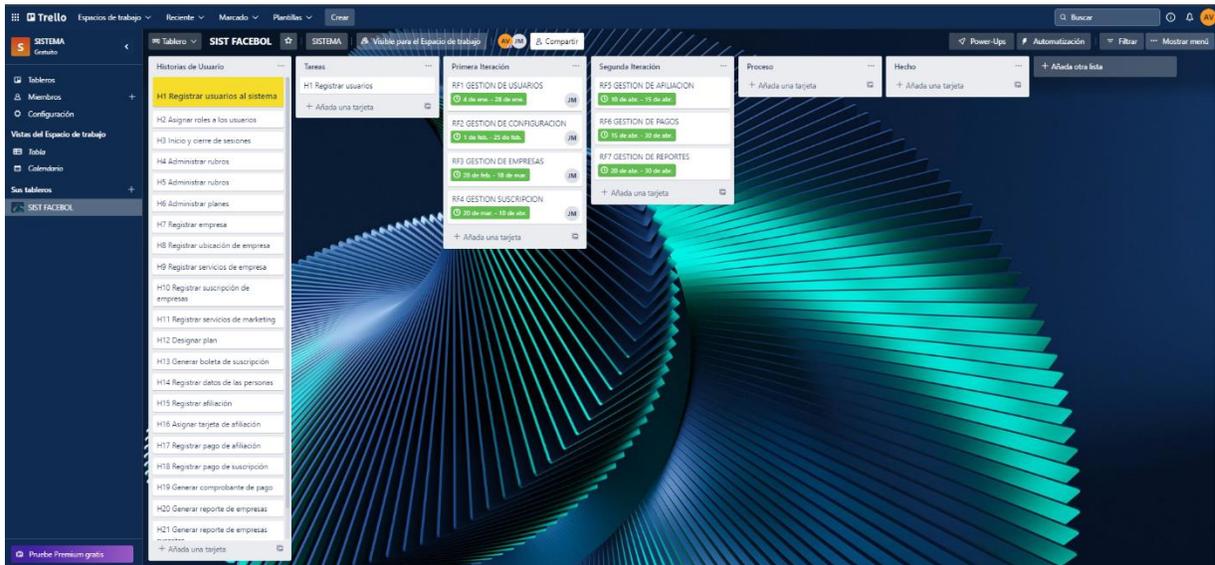
### 3.3. APLICACIÓN DE LA METODOLOGÍA KANBAN

En la aplicación de la metodología Kanban será para la planificación de tareas y funciones del sistema, para lo cual se representa de forma visual.

#### 3.3.1. Implementación del Tablero Kanban

El tablero de Kanban muestra una vista de cuatro columnas para clasificar las tareas de planificación para el desarrollo del sistema, el mismo se implementa gracias a la herramienta "TRELLO".

**Figura 3.3. Muro Kanban**



*Fuente: Elaboración Propia.*

### 3.3.2. Roles Kanban

En la tabla 3.1. se muestra las fases de trabajo identificados en el tablero de Kanban, límite WIP y los responsables que componen el equipo Kanban.

**Tabla 3.1.** Fases de trabajo, limite WIP y responsables

FASE	WIP	RESPONSABLE
TAREAS	4	<b>Analista:</b> Janneth Verónica Ambrocio Mamani
PROCESO	2	<b>Desarrollador:</b> Janneth Verónica Ambrocio Mamani
TEST	2	<b>Desarrollador:</b> Janneth Verónica Ambrocio Mamani

*Fuente: (Elaboración propia).*

### **3.3.3 Fase de Planificación**

- **Análisis de requerimientos**

#### **Requerimientos Funcionales**

Son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas (Sommerville, 2005).

##### **RF1. Gestión de usuarios**

El sistema debe permitir crear, modificar, habilitar y eliminar cuentas de usuarios con los respectivos permisos para acceder al sistema.

##### **RF2. Gestión de configuración**

El sistema debe permitir registrar, modificar, habilitar y eliminar departamentos, rubros y planes.

##### **RF3. Gestión de empresas**

El sistema debe permitir registrar, modificar, habilitar y eliminar información de las empresas en convenio.

##### **RF4. Gestión de suscripción**

El sistema debe permitir crear, modificar habilitar y eliminar suscripción de las empresas en convenio.

##### **RF5. Gestión de afiliación**

El sistema debe permitir registrar, modificar, habilitar, eliminar afiliación de clientes y generar tarjeta de afiliado.

##### **RF6. Gestión de pagos**

El sistema debe permitir crear pagos y generar comprobante de pagos de todas las afiliaciones y convenios.

**RF7. Gestión de reportes**

El sistema debe permitir generar reporte de empresas, reporte de afiliaciones y reporte de suscripciones.

**RF8. Gestión de visitas**

El sistema debe permitir verificar tarjeta para mostrar datos estadísticos de las empresas más visitadas.

**Requerimientos No Funcionales**

Son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema (Sommerville, 2005).

**RNF1.** Los permisos de acceso al sistema podrán ser modificados solamente por el administrador del sistema,

**RNF2.** El sistema deberá tener funcionalidad en diferentes navegadores web de internet.

**RNF3.** Respaldo de la información almacenada en el sistema.

- **Historias de usuario**

**Tabla 3.2.** Historias de usuario

<b>ID</b>	<b>COMO</b>	<b>QUIERO</b>	<b>PARA...</b>
<b>Gestión de Usuarios</b>			
H001	Administrador	Registrar usuarios al sistema	Tener un control de los usuarios
H002	Administrador	Asignar roles a los usuarios	Tener control de acceso de usuarios
H003	Administrador	Inicio y cierre de sesiones	Gestionar el inicio y cierre de sesión en el sistema
<b>Gestión de Configuración</b>			
H004	Administrador	Administrar rubros	Gestionar rubro de las empresas
H005	Administrador	Administrar departamentos	Gestionar departamentos a nivel nacional
H006	Administrador	Administrar planes	Gestionar planes de convenio
<b>Gestión de Empresas</b>			
H007	Administrador	Registrar empresa	Gestiona los datos de la empresa
H008	Administrador	Registrar ubicación de empresa	Tener la ubicación exacta de la empresa
H009	Administrador	Registrar servicios de empresa	Gestionar actividades que brinda la empresa
<b>Gestión de suscripción</b>			
H010	Administrador	Registrar suscripción de	Gestionar empresas en

		empresas	convenio
H011	Administrador	Registrar servicios de marketing	Gestiona servicios de marketing para las empresas
H012	Administrador	Designar plan	Designar plan de convenio de la empresa
H013	Administrador	Generar boleta de suscripción	Tener constancia de las empresas suscritas
<b>Gestión de afiliación</b>			
H014	Administrador	Registrar datos de clientes	Tener control de las personas registradas
H015	Administrador	Registrar afiliación	Gestionar afiliaciones
H016	Administrador	Asignar tarjeta de afiliación	uso de los servicios
<b>Gestión de Pagos</b>			
H017	Administrador	Registrar pago de afiliación	Tener un control en los pagos de afiliación
H018	Administrador	Registrar pago de suscripción	Tener un control en los pagos de empresas en convenio
H019	Administrador	Generar comprobante de pago	Genera comprobante de pagos por lo diferentes ingresos a la empresa
<b>Gestión de Reportes</b>			
H020	Administrador	Generar reporte de empresas	Genera reporte de empresas en convenio
H021	Administrador	Generar reporte de empresas suscritas	Genera reporte de ingresos por pagos

H022	Administrador	Generar reporte de afiliaciones	Tener constancia de los afiliados
<b>Gestión de visitas</b>			
H023	Administrador	Realizar verificación de tarjetas	Tener constancia de ser afiliado
H024	Administrador	Generar datos estadísticos	Tener informe de las visitas a cada empresa

*Fuente: (Elaboración propia).*

### 3.4. APLICACIÓN DE LA METODOLOGÍA XP

#### 3.4.1. Fase de Iteraciones

- **Estimación de Tiempos de la Iteración 1**

**Tabla 3.3.** Estimación de tiempos de la Iteración 1

ID	HISTORIA DE USUARIO	TAREAS	PRIORIZACION	TIEMPO ESTIMADO DIAS
H001	Registrar usuarios al sistema	Modelo de datos lógico. Modelo de datos físico. Usar framework Listar cuentas de usuarios Registrar cuentas de usuario Modificar cuentas de usuarios Eliminar cuentas de usuario Buscar cuentas de usuario	Media	2
H002	Asignar roles a los usuarios	Listar roles Seleccionar roles Modificar roles Habilitar cuentas de usuarios	Media	2

H003	Inicio y cierre de sesiones	Iniciar sesión en el sistema Registrar acceso de usuarios Cerrar sesión en el sistema	Alta	2
H004	Administrar rubros	Listar rubros Registrar rubros Modificar rubros Eliminar rubros Buscar rubros	Alta	2
H005	Administrar departamentos	Listar departamentos Registrar departamento Modificar departamento Eliminar departamento Buscar departamento	Alta	2
H006	Administrar planes	Listar planes Habilitar planes Registrar planes Modificar planes Eliminar planes Buscar planes	Alta	2
H007	Registrar empresa	Listar datos de la empresa Registrar empresa Crear interfaz para el registro de datos de la empresa Modificar empresas Eliminar empresas Buscar empresa	Alta	2
H008	Registrar ubicación de empresa	Crear interfaz para registrar ubicación de empresa	Media	2
H009	Registrar servicios de empresa	Listar servicios de la empresa. Registrar servicios de la	Media	2

		empresa Crear interfaz de servicios de la empresa		
H010	Registrar suscripción de empresas	Registrar empresas suscritas Modificar empresas suscritas Eliminar empresas suscritas Buscar empresas suscritas	Media	2
H011	Registrar servicios de marketing	Listar servicios de marketing Registrar servicios de marketing Modificar servicios de marketing Eliminar servicios de marketing	Media	2
H012	Designar plan	Listar planes de la empresa. Registrar planes de la empresa Modificar planes Eliminar planes	Media	2
H013	Generar boleta de suscripción	Crear boleta de suscripción Descargar boleta de suscripción	Alta	2

*Fuente: (Elaboración propia).*

- **Estimación de Tiempos de la Iteración 2**

**Tabla 3.4.** Estimación de tiempos de la Iteración 2

<b>ID</b>	<b>HISTORIAS DE USUARIO</b>	<b>TAREAS</b>	<b>PRIORIZACION</b>	<b>TIEMPO ESTIMADO EN DIAS</b>
H014	Registrar datos de las personas	Listar datos de personas Registrar personas Modificar datos personas Eliminar personas Buscar personas	Media	2
H015	Registrar afiliación	Listar afiliados Registrar afiliados Modificar datos afiliados Eliminar afiliados Buscar afiliados	Media	2
H016	Asignar tarjeta de afiliación	Listar afiliados Asignar tarjeta Modificar datos afiliados Eliminar afiliados Buscar afiliados	Media	2
H017	Registrar pago de afiliación	Listar pagos Registrar pagos Modificar datos de pago Eliminar pago Buscar pago	Media	2
H018	Registrar pago de suscripción	Listar suscripciones Registrar suscripciones Modificar registro Eliminar registro Buscar registro	Media	2

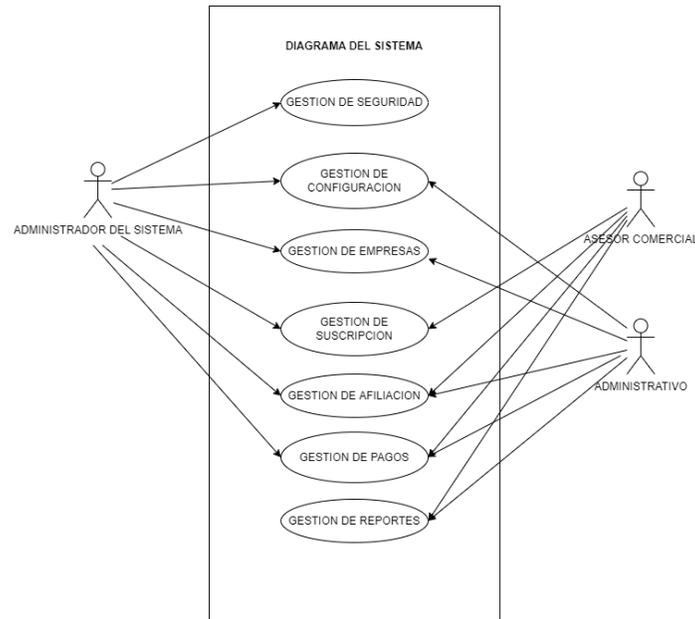
H019	Generar comprobante de pago	Listar pagos Generar comprobante Modificar datos Eliminar comprobante Buscar comprobante	Alta	2
H020	Generar reporte de empresas	Listar reportes Generar reporte Modificar reporte Eliminar reporte Buscar reporte	Alta	2
H021	Generar reporte de empresas suscritas	Listar reportes Generar reporte Modificar reporte Eliminar reporte Buscar reporte	Alta	2
H022	Generar reporte de afiliaciones	Listar reportes Generar reporte Modificar reporte Eliminar reporte Buscar reporte	Alta	2
H023	Realizar verificación de tarjetas	Escanear tarjeta Listar registro de tarjetas Mostrar registro	Alta	2
H024	Generar datos estadísticos	Listar conteo de visitas Mostar gráfico de visitas	Alta	2

**Fuente:** (Elaboración propia).

### 3.4.2. Fase de Diseño

- Diagrama de caso de uso general del sistema

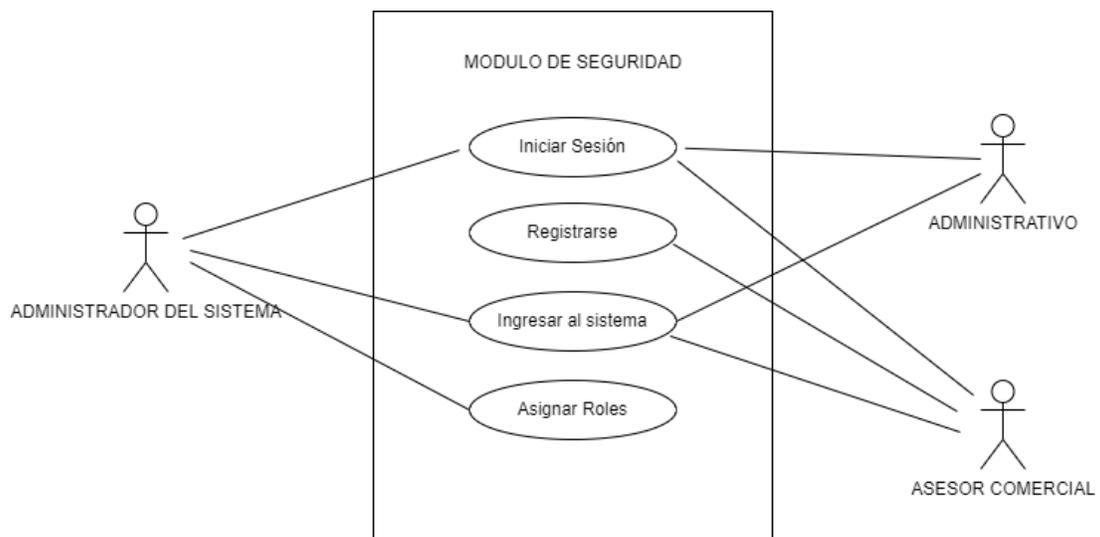
Figura 3.4. Diagrama de caso de uso general del sistema



Fuente: (Elaboración propia).

- Caso de uso del Módulo de Usuario

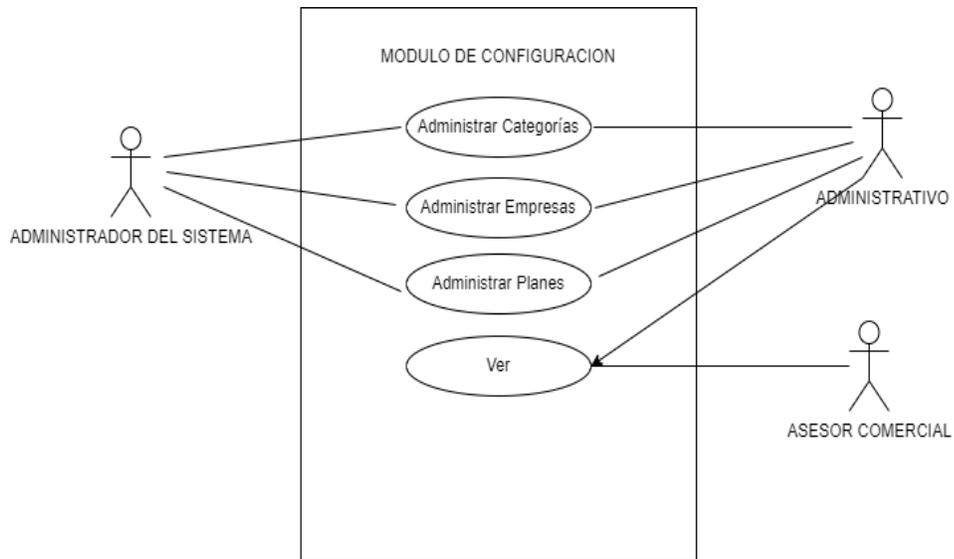
Figura 3.5. Caso de uso gestión de Usuario



Fuente: (Elaboración propia).

- **Caso de uso de gestión de configuración**

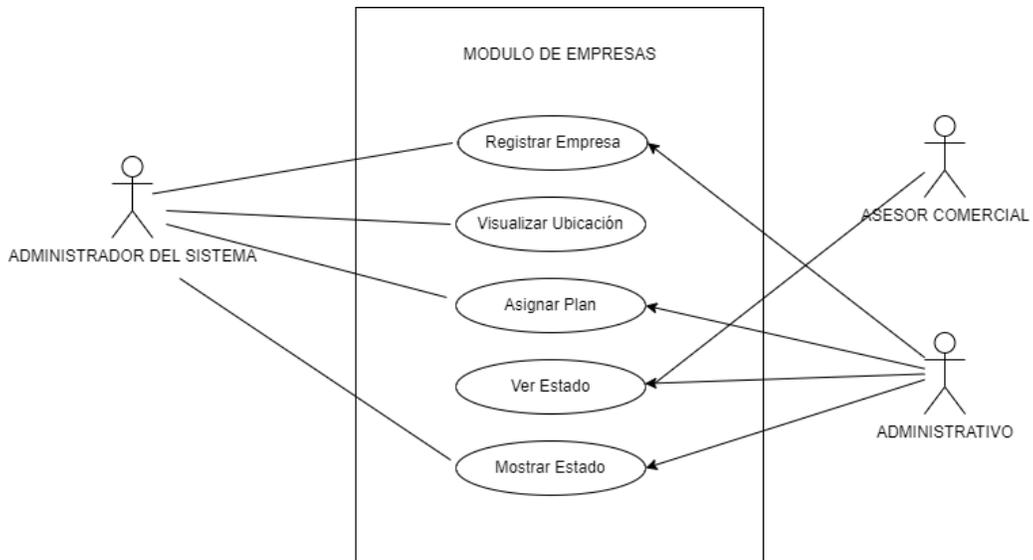
**Figura 3.6.** Caso de uso de gestión de configuración



*Fuente: (Elaboración propia).*

- **Caso de uso de gestión de empresas**

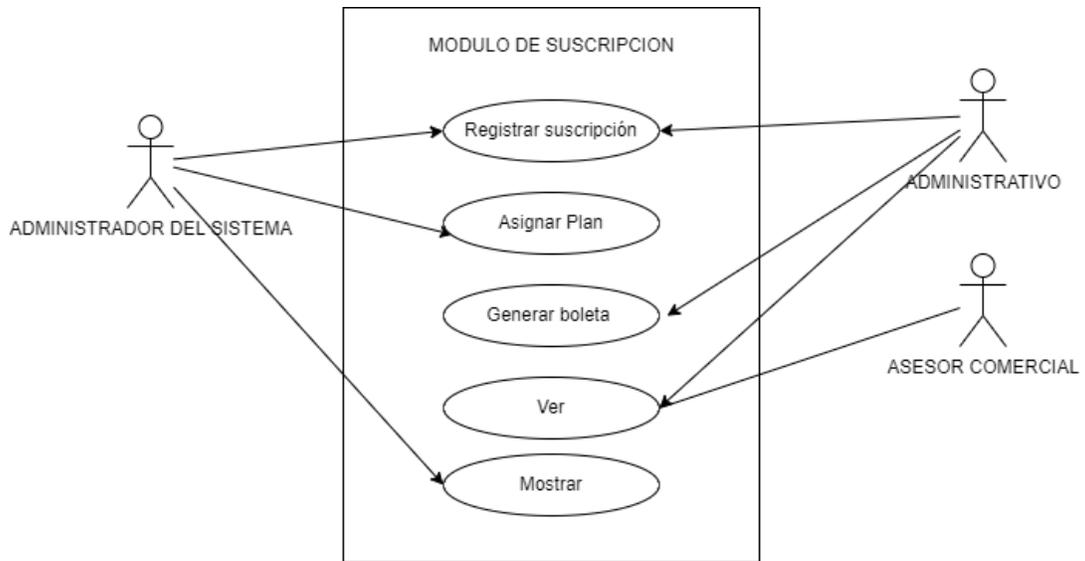
**Figura 3.7.** Caso de uso de gestión de empresas



*Fuente: (Elaboración propia).*

- **Caso de uso de gestión de suscripción**

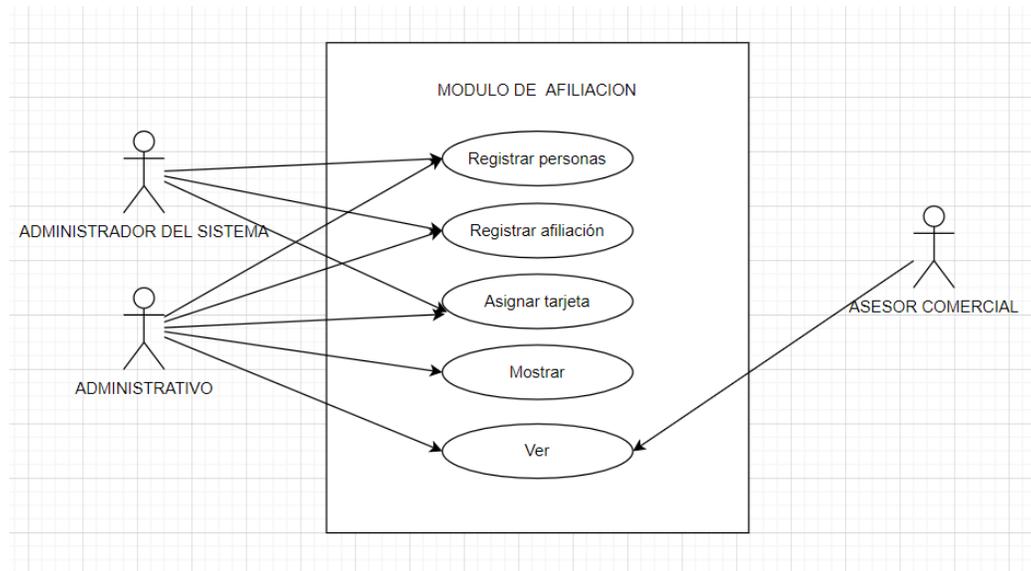
**Figura 3.8.** Caso de uso de gestión de suscripción



*Fuente: (Elaboración propia).*

- **Caso de uso de gestión de afiliación**

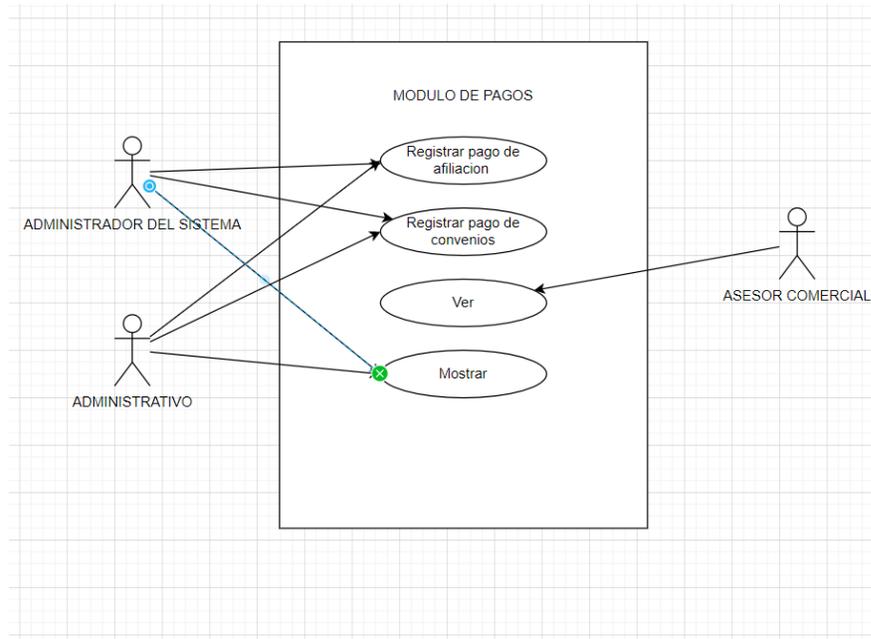
**Figura 3.9.** Caso de uso de gestión de afiliación



*Fuente: (Elaboración propia).*

- **Caso de uso de gestión de pagos**

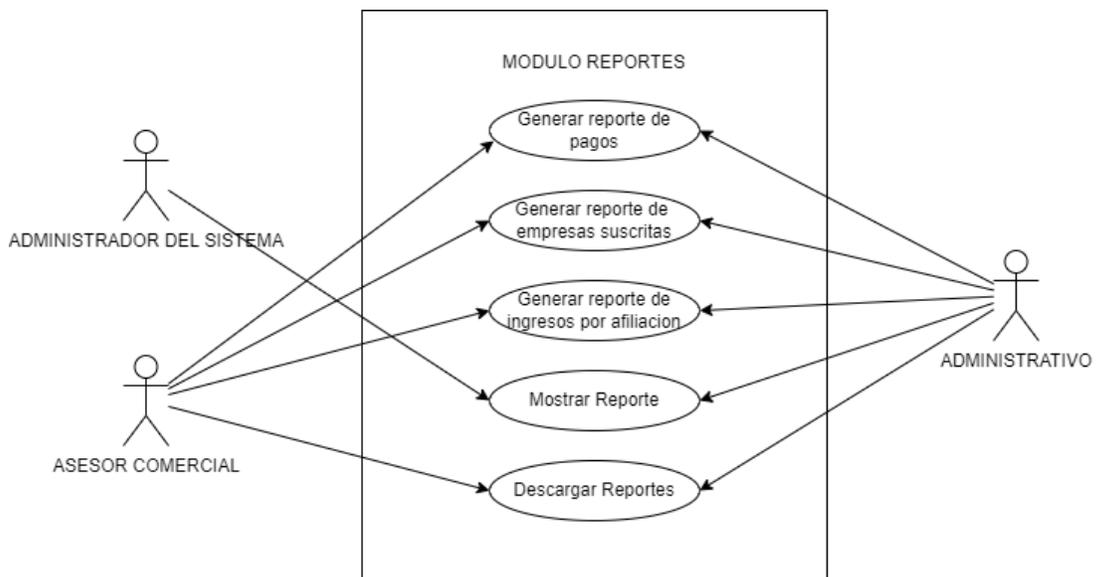
**Figura 3.10.** Caso de uso de gestión de pagos



*Fuente: (Elaboración propia).*

- **Caso de uso de gestión de reportes**

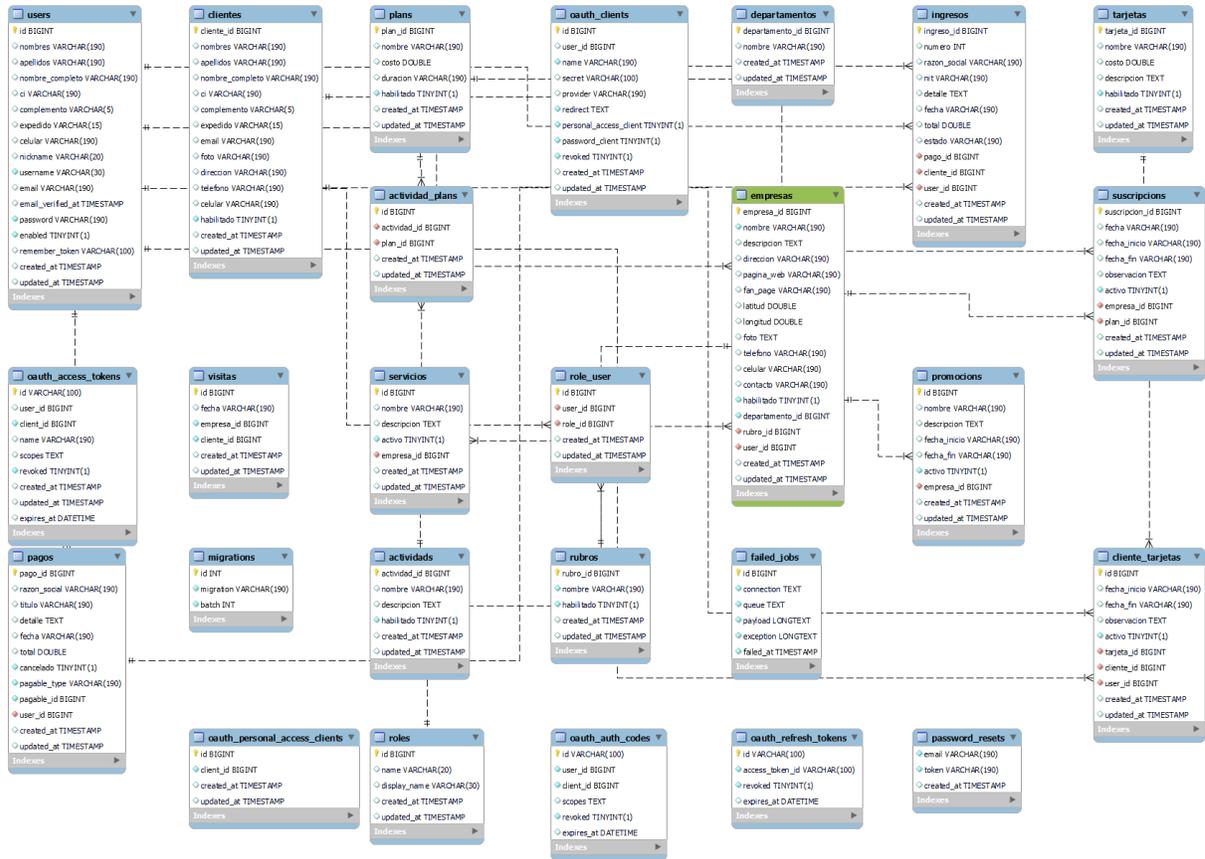
**Figura 3.11.** Caso de uso de gestión de reportes



*Fuente: (Elaboración propia).*

### 3.4.3. Diagrama físico de la Base de Datos del sistema

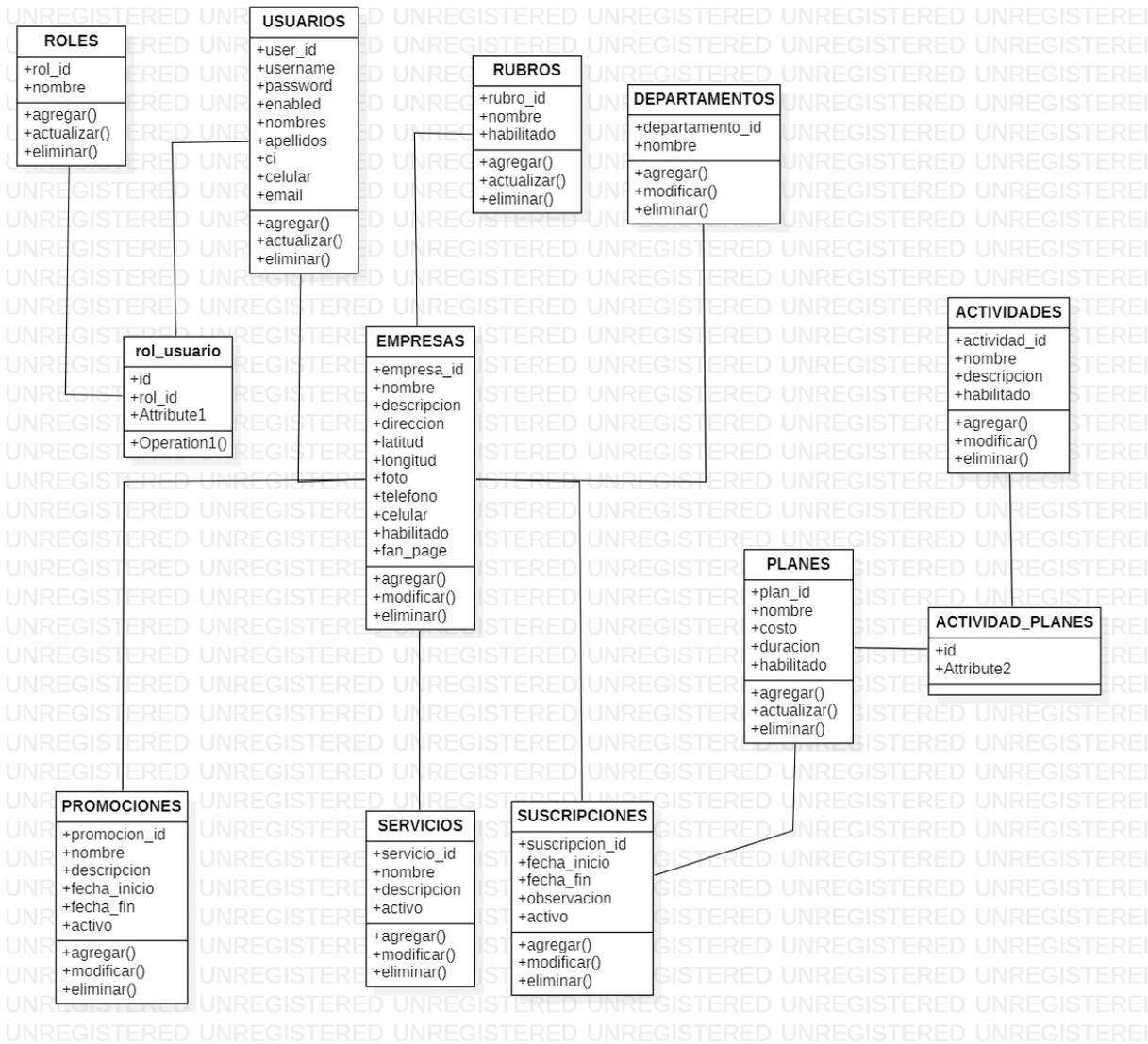
Figura 3.12. Diagrama físico de la Base de Datos del sistema



Fuente: (Elaboración propia).

### 3.4.4. Diagrama de Base de Datos

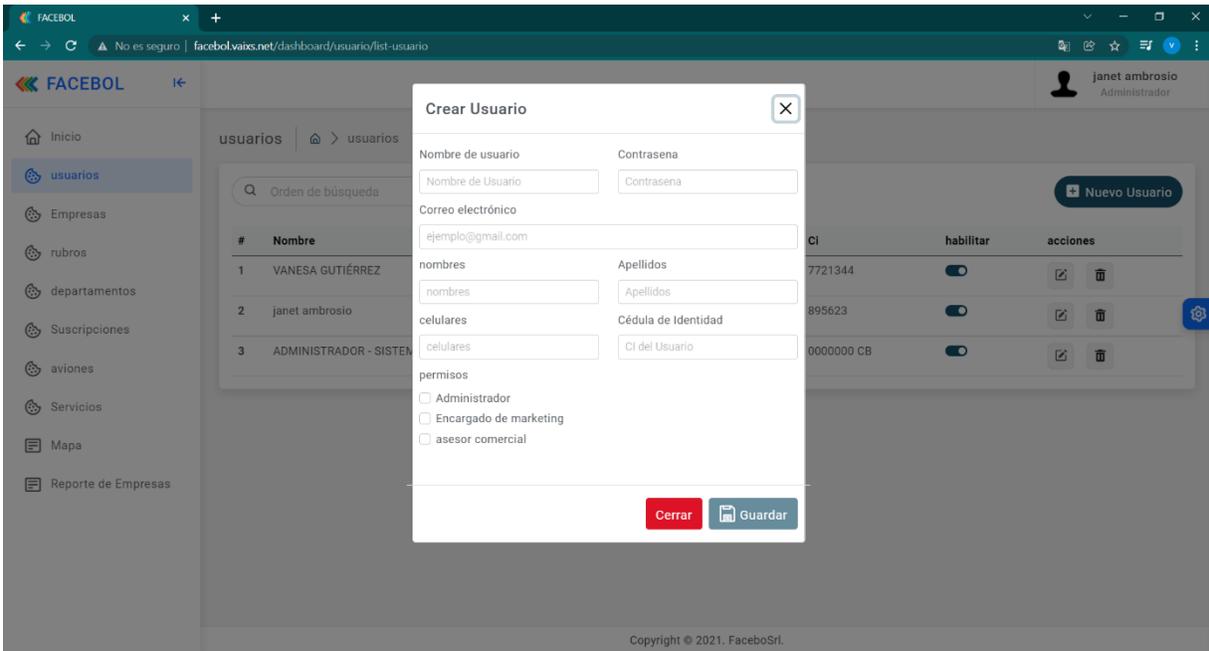
Figura 3.13. Diagrama de Base de Datos



Fuente: (Elaboración propia).

### 3.4.3. Fase de Implementación

Figura 3.14. Registrar usuarios al sistema



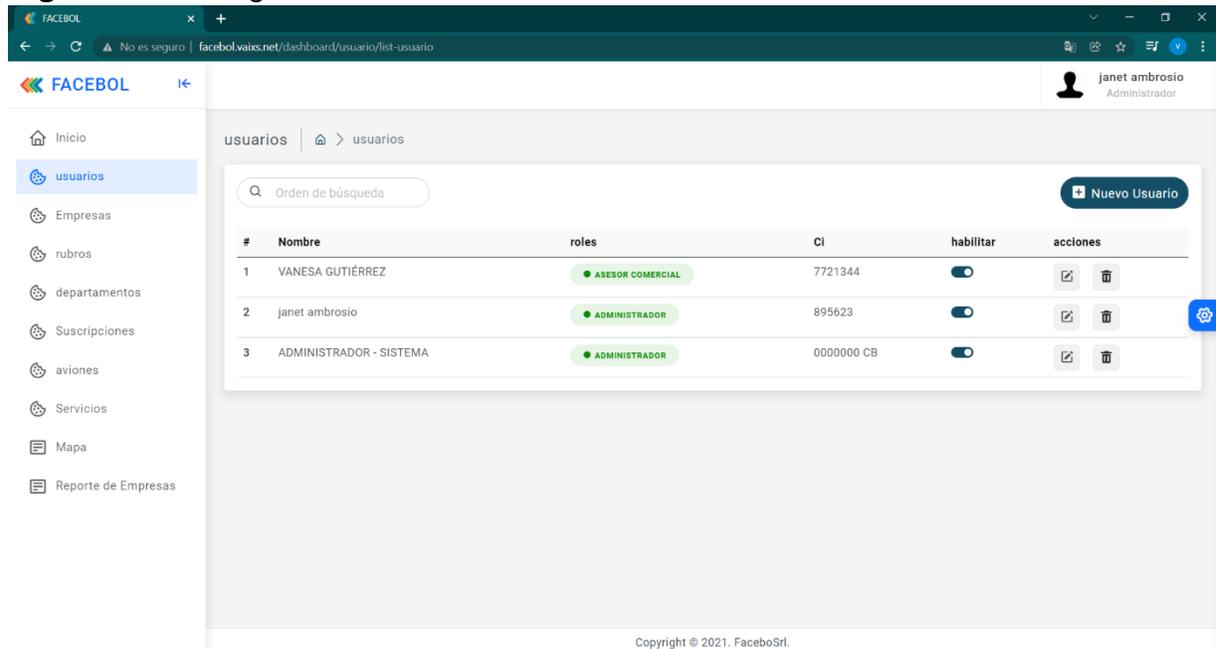
Fuente: (Elaboración propia).

Figura 3.15. Código referente a la vista registro de usuarios

```
...<!DOCTYPE html> == $0
<html lang="en">
  <head>...</head>
  <body class="pace-done modal-open" style="padding-right: 0px;">
    <div class="pace pace-inactive" aria-hidden="true">...</div>
    <app-root _ngghost-daj-c116 ng-version="12.0.2" aria-hidden="true">...</app-root>
    <div class="page-loading" aria-hidden="true">...</div>
    <script src="runtime.js" defer></script>
    <script src="polyfills.js" defer></script>
    <script src="styles.js" defer></script>
    <script src="scripts.js" defer></script>
    <script src="vendor.js" defer></script>
    <script src="main.js" defer></script>
    <svg id="SvgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;" aria-hidden="true">...</svg>
    <div class="overlay-container" aria-hidden="true">...</div>
    <ngb-modal-backdrop style="z-index: 1050;" aria-hidden="true" class="modal-backdrop fade show"></ngb-modal-backdrop>
    <ngb-modal-window role="dialog" tabindex="-1" aria-modal="true" class="d-block modal fade show">...</ngb-modal-window>
  </body>
</html>
```

Fuente: (Elaboración propia).

**Figura 3.16.** Asignar roles a los usuarios



**Fuente:** [Elaboración propia]

**Figura 3.17.** Código referente a la vista asignación de roles.

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body class="pace-done" style == $0
    <div class="pace pace-inactive">...</div>
    <app-root _ngghost-daj-c116 ng-version="12.0.2">...</app-root>
    <div class="page-loading">...</div>
    <script src="runtime.js" defer></script>
    <script src="polyfills.js" defer></script>
    <script src="styles.js" defer></script>
    <script src="scripts.js" defer></script>
    <script src="vendor.js" defer></script>
    <script src="main.js" defer></script>
    <svg id="SvgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;">...</svg>
    <div class="overlay-container">...</div>
  </body>
</html>
```

**Fuente:** (Elaboración propia).

Figura 3.18. Inicio y cierre de sesiones

FaceBol srl.  
¡Hazlo Diferente!

INGRESE SUS CREDENCIALES

Usuario  
Ingresar Usuario

Contraseña  
Ingresar Contraseña

Iniciar Sesión

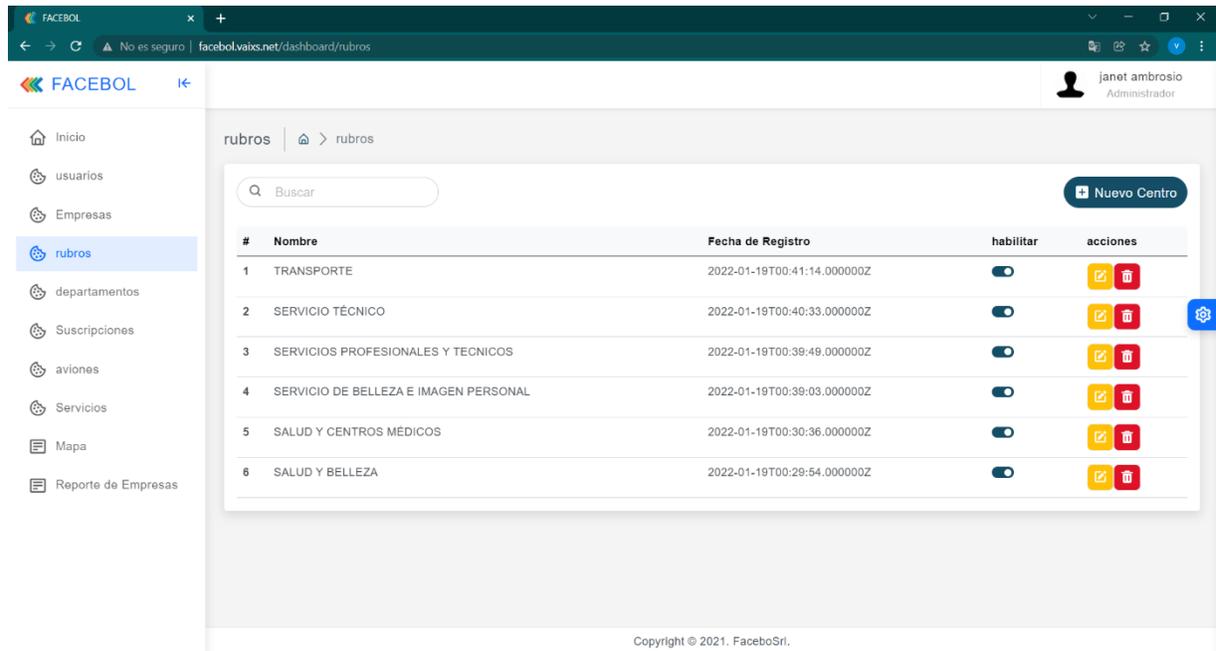
Fuente: (Elaboración propia).

Figura 3.19. Código referente a la vista inicio de sesión

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body class="pace pace-inactive">
    <div class="pace pace-inactive">...</div>
    <app-root _ngcontent-daj-c115 ng-version="12.0.2">
      <router-outlet _ngcontent-daj-c115></router-outlet>
      <app-content-layout _ngcontent-daj-c115 class="ng-star-inserted">
        <router-outlet _ngcontent-daj-c115></router-outlet>
        <app-sign-in _ngcontent-daj-c226 class="ng-star-inserted">
          <div _ngcontent-daj-c226 class="bg-login">
            <div _ngcontent-daj-c226 class="wrapper nav-collapsed">
              <div _ngcontent-daj-c226 class="section-authentication-signin d-flex align-items-center justify-content-center my-5 my-lg-0">
                <div _ngcontent-daj-c226 class="container-fluid">
                  <div _ngcontent-daj-c226 class="row row-cols-1 row-cols-lg-2 row-cols-xl-3">
                    <div class="row row-cols-1 row-cols-lg-2 row-cols-xl-3">
                      <div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </app-sign-in>
        </app-content-layout>
      </app-root>
    <div class="page-loading">...</div>
    <script src="runtime.js" defer></script>
    <script src="polyfills.js" defer></script>
    <script src="styles.js" defer></script>
    <script src="scripts.js" defer></script>
    <script src="vendor.js" defer></script>
    <script src="main.js" defer></script>
    <svg id="svgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;">...</svg>
    <div class="overlay-container">...</div>
  </body>
</html>
```

Fuente: (Elaboración propia).

Figura 3.20. Administrar rubros



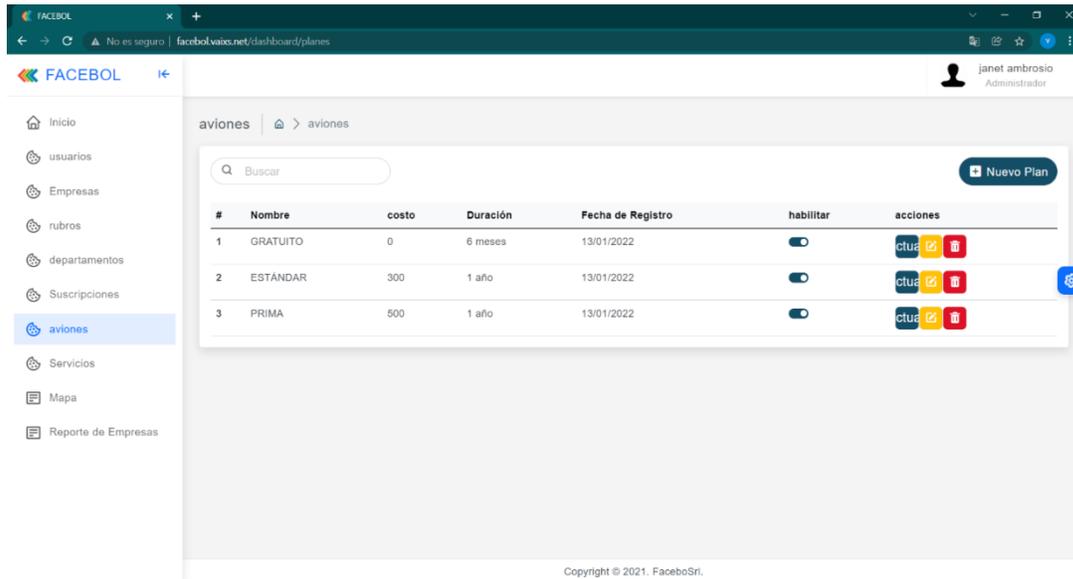
Fuente: (Elaboración propia).

Figura 3.21. Código referente a la vista rubros

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body class="pace-done" style="
    <div class="pace pace-inactive">...</div>
    <app-root _ngghost-daj-c116 ng-version="12.0.2">
      <router-outlet _ngcontent-daj-c116></router-outlet>
      <app-full-layout _ngghost-daj-c114 class="ng-star-inserted">...</app-full-layout>
      <!--container-->
    </app-root>
  <div class="page-loading">...</div> == $0
  <script src="runtime.js" defer></script>
  <script src="polyfills.js" defer></script>
  <script src="styles.js" defer></script>
  <script src="scripts.js" defer></script>
  <script src="vendor.js" defer></script>
  <script src="main.js" defer></script>
  <svg id="SvgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;">...</svg>
  <div class="overlay-container">...</div>
</body>
</html>
```

Fuente: (Elaboración propia).

Figura 3.22. Administrar planes



Fuente: (Elaboración propia).

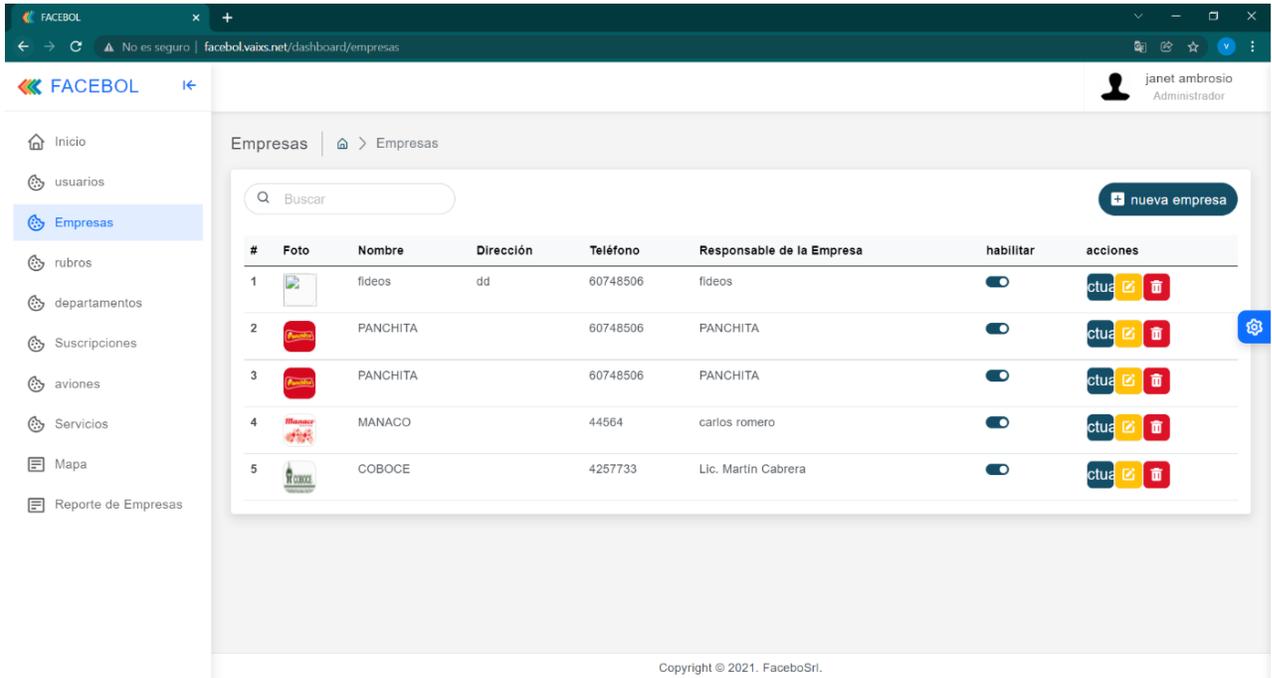
Figura 3.23. Código referente a la vista planes

```

<!DOCTYPE html>
<html lang="en">
<head></head>
<body class="pace-done">
  <div class="pace pace-inactive"></div>
  <app-root _ngcontent-ewb-c116 ng-version="12.0.2">
    <router-outlet _ngcontent-ewb-c116></router-outlet>
    <app-full-layout _ngcontent-ewb-c114>
      <div _ngcontent-ewb-c114 class="wrapper" ng-reflect-ng-class="[object Object]">
        <app-sidebar _ngcontent-ewb-c114></app-sidebar>
        <app-navbar _ngcontent-ewb-c114 _ngcontent-ewb-c109></app-navbar>
        <div _ngcontent-ewb-c114 class="page-wrapper">
          <div _ngcontent-ewb-c114 class="page-content">
            <router-outlet _ngcontent-ewb-c114></router-outlet>
            <app-plan _ngcontent-ewb-c190>
              <div _ngcontent-ewb-c190 class="page-breadcrumb d-none d-sm-flex align-items-center mb-3"></div>
              <div _ngcontent-ewb-c190 class="card">
                <div _ngcontent-ewb-c190 class="card-body">
                  <div _ngcontent-ewb-c190 class="d-lg-flex align-items-center mb-4 gap-3">
                    <div _ngcontent-ewb-c190 class="position-relative"></div>
                    <div _ngcontent-ewb-c190 class="ms-auto"></div>
                  </div>
                  <div _ngcontent-ewb-c190 class="table-responsive"></div>
                </div>
              </app-plan>
            <!-- container -->
            <div _ngcontent-ewb-c114 class="overlay" ng-reflect-ng-class="[object Object]"></div>
          </div>
        </div>
        <app-footer _ngcontent-ewb-c114 _ngcontent-ewb-c108></app-footer>
      </div>
      <app-color-switcher _ngcontent-ewb-c114 _ngcontent-ewb-c112></app-color-switcher>
    </app-full-layout>
  <!-- container -->
</app-root>
<div class="page-loading"></div>
<script src="runtime.js" defer></script>
<script src="polyfills.js" defer></script>
<script src="styles.js" defer></script>
<script src="vendor.js" defer></script>
<script src="main.js" defer></script>
<svg id="SvgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;"></svg>
<script type="text/javascript" async defer id="agmGoogleMapsApiScript" src="https://maps.googleapis.com/maps/api/js?v=quarterly&callback=agmLazyMapsAPILoader&key=AIzaSyDXKdH0ttopPV12H12RnLu1b1jC8RCQod"></script>
</body>
</html> == $0
  
```

Fuente: (Elaboración propia).

Figura 3.24. Registrar empresas



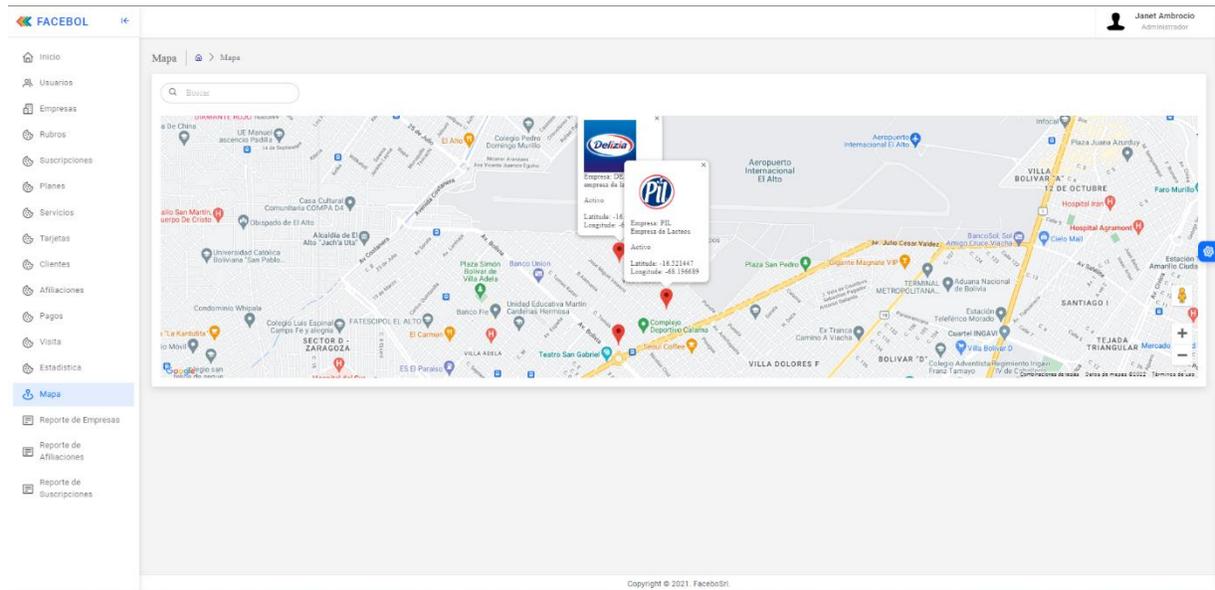
Fuente: (Elaboración propia).

Figura 3.25. Código referente a la vista empresas

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body class="pace-done" style="
    <div class="pace pace-inactive">...</div>
    <app-root_nghost-daj-c116 ng-version="12.0.2">...</app-root>
    <div class="page-loading">...</div> == $0
    <script src="runtime.js" defer></script>
    <script src="polyfills.js" defer></script>
    <script src="styles.js" defer></script>
    <script src="scripts.js" defer></script>
    <script src="vendor.js" defer></script>
    <script src="main.js" defer></script>
    <svg id="SvgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;">...</svg>
    <div class="overlay-container">...</div>
  </body>
</html>
```

Fuente: (Elaboración propia).

**Figura 3.26.** Registrar ubicación de empresas



*Fuente: (Elaboración propia).*

**Figura 3.27.** Código referente a la vista ubicación de empresas

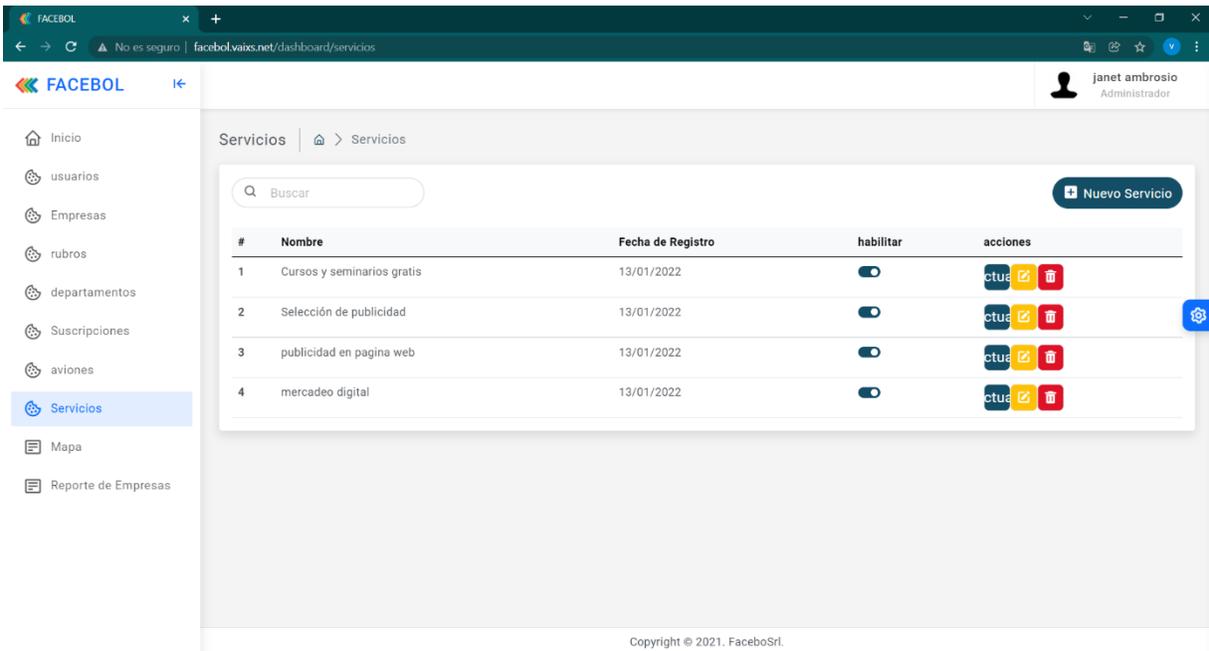
```

<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body class="
    <div class="pace pace-inactive"></div>
    <app-root _ngghost-ewb-c116 ng-version="12.0.2">
      <router-outlet _ngcontent-ewb-c116></router-outlet>
      <div _ngcontent-ewb-c114 class="wrapper" ng-reflect-ng-class="[object Object]">
        <app-sidebar _ngcontent-ewb-c114 _</app-sidebar>
        <app-navbar _ngcontent-ewb-c114 _ngghost-ewb-c109 _</app-navbar>
        <div _ngcontent-ewb-c114 class="page-wrapper">
          <div _ngcontent-ewb-c114 class="page-content-wrapper">
            <div _ngcontent-ewb-c114 class="page-content">
              <router-outlet _ngcontent-ewb-c114></router-outlet>
              <app-empresa _ngghost-ewb-c166>
                <div _ngcontent-ewb-c166 class="page-breadcrumb d-none d-sm-flex align-items-center mb-3"></div> (flex)
                <div _ngcontent-ewb-c166 class="card"></div> (flex)
              </app-empresa>
            </div>
          </div>
        </div>
      <app-footer _ngcontent-ewb-c114 _ngghost-ewb-c108 _</app-footer>
    </div>
    <app-color-switcher _ngcontent-ewb-c114 _ngghost-ewb-c112 _</app-color-switcher>
  </app-full-layout>
</div>
</app-root>
<div class="page-loading"></div>
<script src="runtime.js" defer></script>
<script src="polyfills.js" defer></script>
<script src="styles.js" http://localhost:4200/runtime.js>
<script src="scripts.js" defer></script>
<script src="vendor.js" defer></script>
<script src="main.js" defer></script>
<svg id="SvgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;"></svg>
<script type="text/javascript" async defer id="agmGoogleMapsApiScript" src="https://maps.googleapis.com/maps/api/js?v=guarterly&callback=agmLazyMapsAPILoader&key=AIzaSyDKXkdH0dtqPVL2H12RnUa_1bjcXR0o4"></script>
</body>
</html>

```

*Fuente: (Elaboración propia)*

Figura 3.28. Registrar servicios de empresa



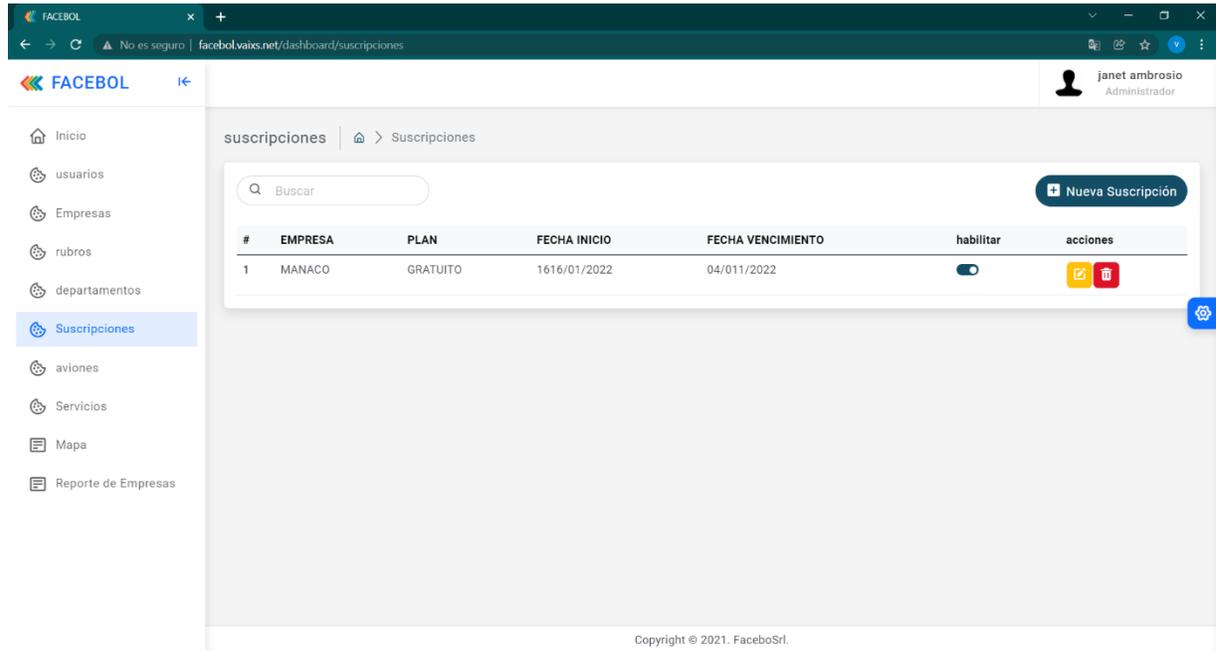
Fuente: (Elaboración propia).

Figura 3.29. Código referente a la vista servicios

```
<!DOCTYPE html>
<html lang="es">
  <head></head>
  <body class="pace pace-done">
    <div class="pace pace-inactive"></div>
    <app-root _ngcontent-ewb-c116 ng-version="12.0.2">
      <router-outlet _ngcontent-ewb-c116></router-outlet>
      <app-full-layout _ngcontent-ewb-c114>
        <div _ngcontent-ewb-c114 class="wrapper ng-reflect-ng-class="[object Object]">
          <app-sidebar _ngcontent-ewb-c114></app-sidebar>
          <app-navbar _ngcontent-ewb-c114 _ngcontent-ewb-c189></app-navbar>
          <div _ngcontent-ewb-c114 class="page-wrapper">
            <div _ngcontent-ewb-c114 class="page-content-wrapper">
              <div _ngcontent-ewb-c114 class="page-content">
                <router-outlet _ngcontent-ewb-c114></router-outlet>
                <app-empresa _ngcontent-ewb-c166>
                  <div _ngcontent-ewb-c166 class="page-breadcrumb d-none d-sm-flex align-items-center mb-3"></div>
                  <div _ngcontent-ewb-c166 class="card">
                    <div _ngcontent-ewb-c166 class="card-body">
                      <div _ngcontent-ewb-c166 class="d-lg-flex align-items-center mb-4 gap-3">
                        <div _ngcontent-ewb-c166 class="position-relative">
                          <div _ngcontent-ewb-c166 class="ms-auto"></div>
                        </div>
                        <div _ngcontent-ewb-c166 class="table-responsive"></div>
                      </div>
                    </div>
                  </div>
                </app-empresa>
                <!-- container -->
                <div _ngcontent-ewb-c114 class="overlay ng-reflect-ng-class="[object Object]"></div>
              </div>
            </div>
          </div>
          <app-footer _ngcontent-ewb-c114 _ngcontent-ewb-c188></app-footer>
        </div>
        <app-color-switcher _ngcontent-ewb-c114 _ngcontent-ewb-c112></app-color-switcher>
      </app-full-layout>
    </app-root>
    <!-- container -->
    <div class="page-loading"></div>
    <script src="runtime.js" defer></script>
    <script src="polyfills.js" defer></script>
    <script src="styles.js" defer></script>
    <script src="scripts.js" defer></script>
    <script src="vendor.js" defer></script>
    <script src="main.js" defer></script>
    <svg id="SvgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;">
    <script type="text/javascript" async defer id="agmGoogleMapsApiScript" src="https://maps.googleapis.com/maps/api/js?v=quarterly&callback=agmLazyMapsAPILoader&key=AIzaSyDjKXkH0tngPVI2H128mJa1b1CxRC0q4"></script>
  </body>
</html>
```

Fuente: (Elaboración propia).

**Figura 3.30.** Registrar suscripción de la empresa



*Fuente: (Elaboración propia).*

**Figura 3.31.** Código referente a la vista suscripción

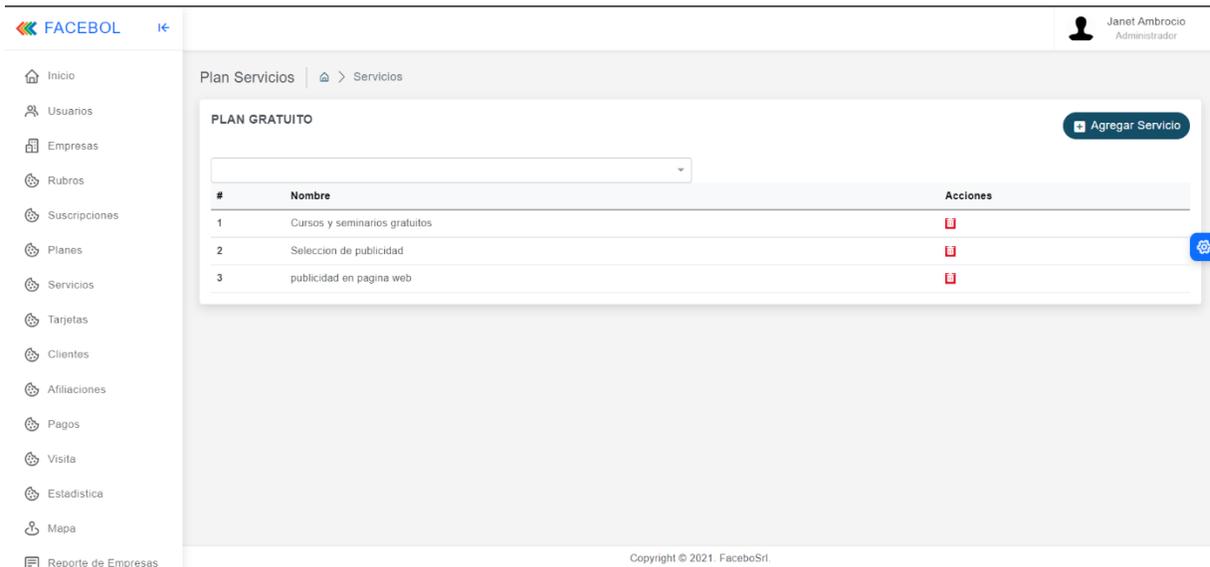
```

<!DOCTYPE html>
<html lang="en">
<head>_</head>
<body class="pace pace-done">
  <div class="pace pace-inactive">_</div>
  <app-root _ngghost-ewb-c116 ng-version="12.0.2">
    <router-outlet _ngcontent-ewb-c116></router-outlet>
    <app-full-layout _ngghost-ewb-c114>
      <div _ngcontent-ewb-c114 class="wrapper" ng-reflect-ng-class="[object Object]">
        <app-sidebar _ngcontent-ewb-c114></app-sidebar>
        <app-navbar _ngcontent-ewb-c114 _ngghost-ewb-c109></app-navbar>
        <div _ngcontent-ewb-c114 class="page-wrapper">
          <div _ngcontent-ewb-c114 class="page-content-wrapper">
            <div _ngcontent-ewb-c114 class="page-content">
              <router-outlet _ngcontent-ewb-c114></router-outlet>
              <app-suscripcion _ngghost-ewb-c196>
                <div _ngcontent-ewb-c196 class="page-breadcrumb d-none d-sm-flex align-items-center mb-3"></div>
                <div _ngcontent-ewb-c196 class="card">
                  <div _ngcontent-ewb-c196 class="card-body">
                    <div _ngcontent-ewb-c196 class="d-lg-flex align-items-center mb-4 gap-3">
                      <div _ngcontent-ewb-c196 class="position-relative"></div>
                      <div _ngcontent-ewb-c196 class="ms-auto">_</div>
                    </div>
                    <div _ngcontent-ewb-c196 class="table-responsive"></div>
                  </div>
                </app-suscripcion>
                <!--container-->
                <div _ngcontent-ewb-c114 class="overlay" ng-reflect-ng-class="[object Object]"></div>
              </div>
            </div>
          </div>
          <app-footer _ngcontent-ewb-c114 _ngghost-ewb-c108></app-footer>
        </div>
        <app-color-switcher _ngcontent-ewb-c114 _ngghost-ewb-c112></app-color-switcher>
      </app-full-layout>
    <!--container-->
  </app-root>
  <div class="page-loading"></div>
  <script src="runtime.js" defer></script>
  <script src="polyfills.js" defer></script>
  <script src="styles.js" defer></script>
  <script src="scripts.js" defer></script>
  <script src="vendor.js" defer></script>
  <script src="main.js" defer></script>
  <svg id="SvgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;"></svg>
  <script type="text/javascript" async defer id="agmGoogleMapsApiScript" src="https://maps.googleapis.com/maps/api/js?v=quarterly&callback=agmLazyMapsAPILoader&key=AIzaSyDKXdh0tqgPV17H12bnJa_1bjCwR0od"></script>
</body>
</html>

```

*Fuente: (Elaboración propia).*

Figura 3.32. Designar plan



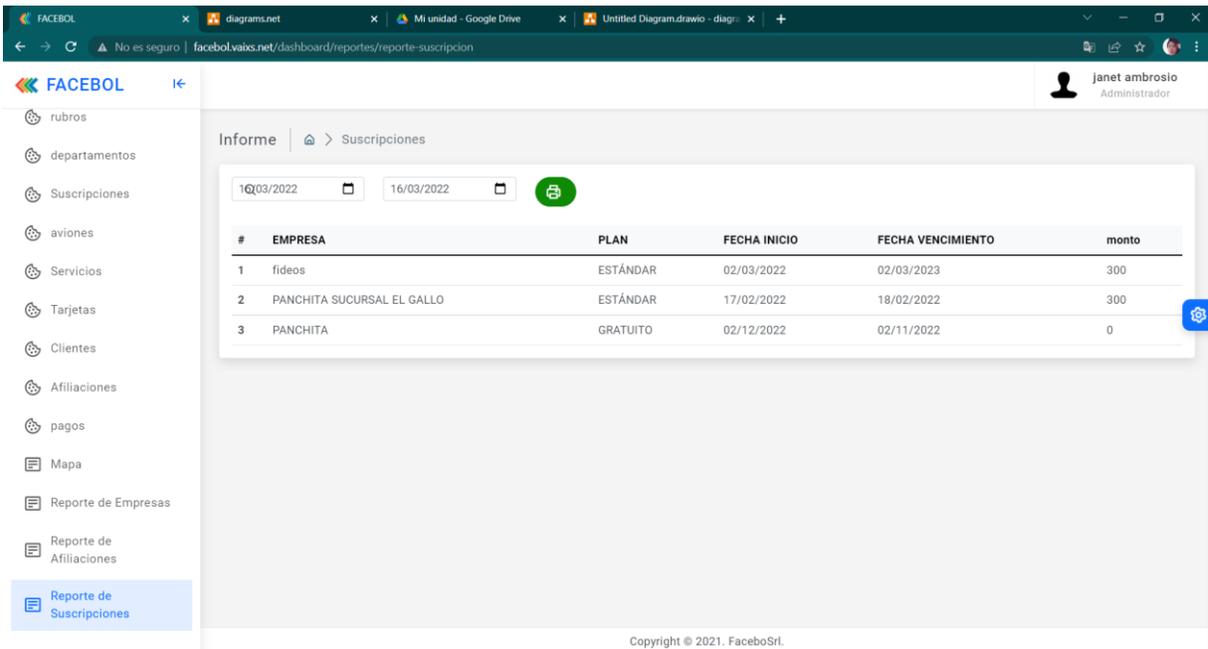
Fuente: (Elaboración propia).

Figura 3.33. Código referente a la vista de planes

```
<!DOCTYPE html>
<html lang="en">
  <head>_</head>
  <body class="pace-done">
    <div class="pace pace-inactive">_</div>
    <app-root _ngghost-ewb-cl16 ng-version="12.0.2">_</app-root>
    <div class="page-loading">_</div> == $0
    <script src="runtime.js" defer></script>
    <script src="polyfills.js" defer></script>
    <script src="styles.js" defer></script>
    <script src="scripts.js" defer></script>
    <script src="vendor.js" defer></script>
    <script src="main.js" defer></script>
    <svg id="SvgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;">_</svg>
    <script type="text/javascript" async defer id="agmGoogleMapsApiScript" src="https://maps.googleapis.com/maps/api/js?v=quarterly&callback=ag
    mLazyMapsAPILoader&key=AIzaSyDlXKdH0dtqgPVl2HI2RnUa_1bjCXRcQ04"></script>
  </body>
</html>
```

Fuente: (Elaboración propia).

Figura 3.34. Generar boleta de suscripción



Fuente: (Elaboración propia).

Figura 3.35. Código referente a la vista de suscripciones

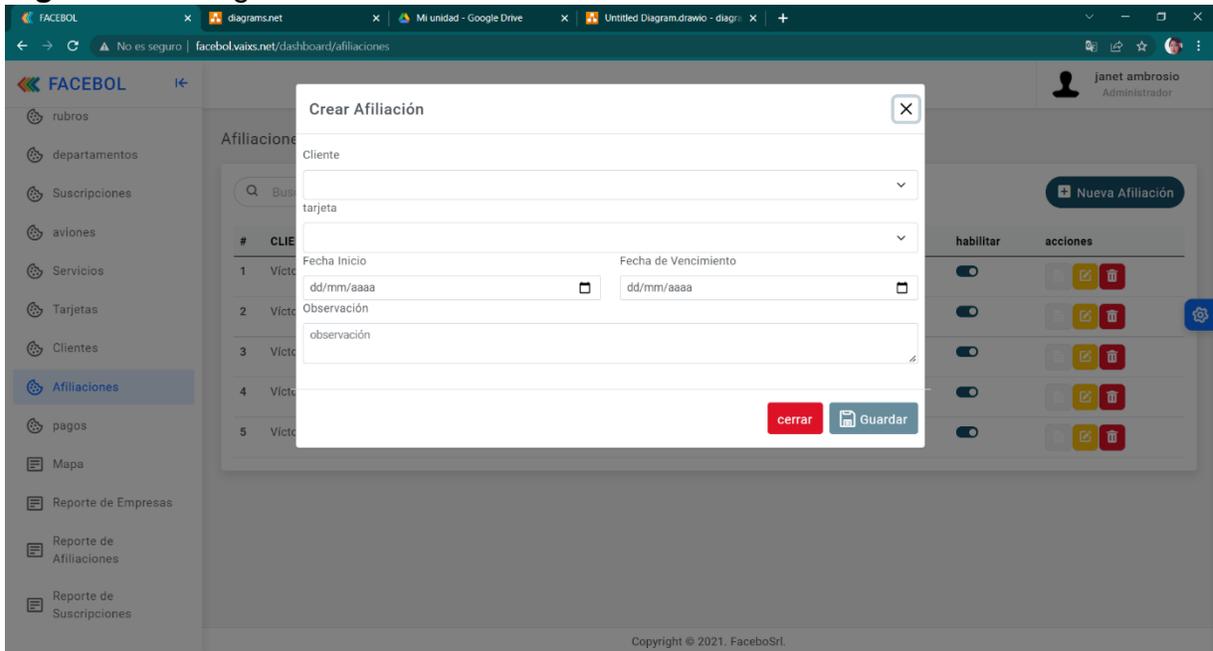
```

<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body class="pace-done">
    <div class="pace pace-inactive"></div>
    <app-root _ngghost-ewb-c116 ng-version="12.0.2">
      <router-outlet _ngcontent-ewb-c116></router-outlet>
      <app-full-layout _ngghost-ewb-c114>
        <div _ngcontent-ewb-c114 class="wrapper" ng-reflect-ng-class="[object Object]">
          <app-sidebar _ngcontent-ewb-c114></app-sidebar>
          <app-navbar _ngcontent-ewb-c114 _ngghost-ewb-c109></app-navbar>
          <div _ngcontent-ewb-c114 class="page-wrapper">
            <div _ngcontent-ewb-c114 class="page-content-wrapper">
              <div _ngcontent-ewb-c114 class="page-content">
                <router-outlet _ngcontent-ewb-c114></router-outlet>
                <app-suscripcion _ngghost-ewb-c196>
                  <div _ngcontent-ewb-c196 class="page-breadcrumb d-none d-sm-flex align-items-center mb-3"></div>
                  <div _ngcontent-ewb-c196 class="card">
                    <div _ngcontent-ewb-c196 class="card-body"></div>
                  </div>
                </app-suscripcion>
              </div>
            </div>
          </div>
          <app-footer _ngcontent-ewb-c114 _ngghost-ewb-c108></app-footer>
        </div>
        <app-color-switcher _ngcontent-ewb-c114 _ngghost-ewb-c112></app-color-switcher>
      </app-full-layout>
    </div>
  </app-root>
  <div class="page-loading"></div>
  <script src="runtime.js" defer></script>
  <script src="polyfills.js" defer></script>
  <script src="styles.js" defer></script>
  <script src="scripts.js" defer></script>
  <script src="vendor.js" defer></script>
  <script src="main.js" defer></script>
  <svg id="SvgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;"></svg>
  <script type="text/javascript" async defer id="agmGoogleMapsApiScript" src="https://maps.googleapis.com/maps/api/js?v=quarterly&callback=
  m1zzyMapsAPILoader&key=AIzaSyDlXKtDdtqPv1zH1zRnJa1bJcXtCRoQ4"></script>
</body>
</html>

```

Fuente: (Elaboración propia).

Figura 3.36. Registrar datos de los clientes



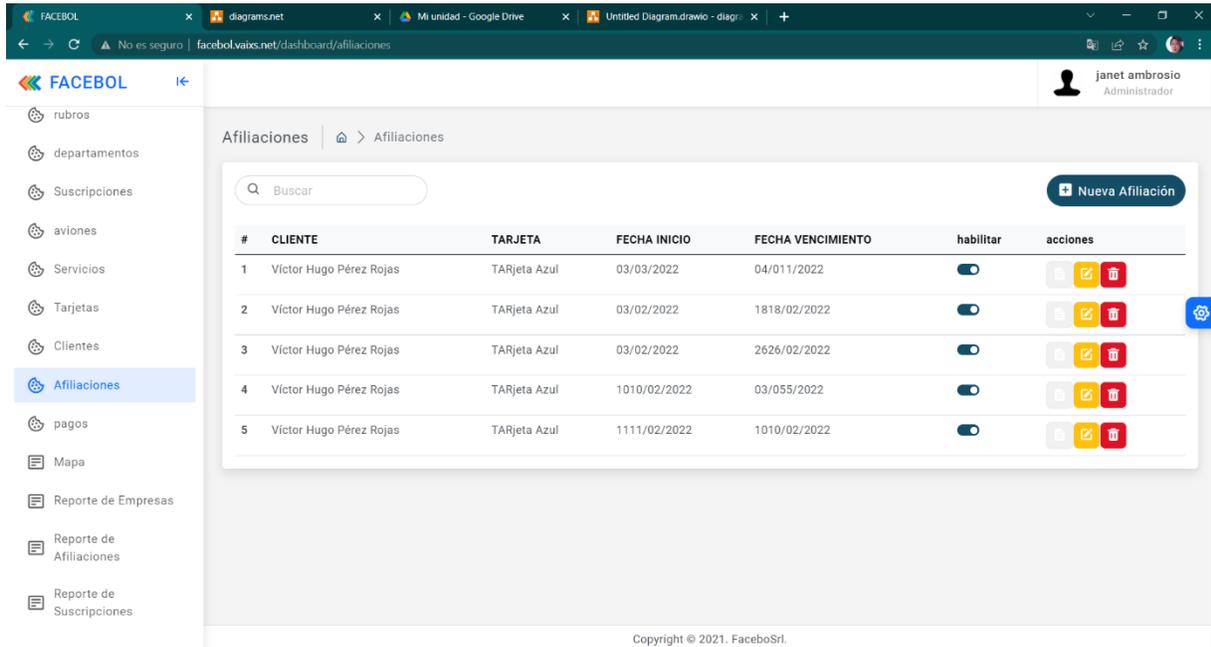
Fuente: (Elaboración propia).

Figura 3.37. Código referente a la vista de clientes

```
<!DOCTYPE html>
<html lang="en">
  <head>_</head>
  <body class="pace-done">
    <div class="pace pace-inactive">_</div>
    <app-root _ngcontent-ewb-c116 ng-version="12.0.2">
      <router-outlet _ngcontent-ewb-c116></router-outlet>
      <app-full-layout _ngcontent-ewb-c114>
        <div _ngcontent-ewb-c114 class="wrapper" ng-reflect-ng-class="[object Object]">
          <app-sidebar _ngcontent-ewb-c114></app-sidebar>
          <app-navbar _ngcontent-ewb-c114 _ngcontent-ewb-c109></app-navbar>
          <div _ngcontent-ewb-c114 class="page-wrapper">
            <div _ngcontent-ewb-c114 class="page-content-wrapper">
              <div _ngcontent-ewb-c114 class="page-content">
                <router-outlet _ngcontent-ewb-c114></router-outlet>
                <app-cliente _ngcontent-ewb-c147>
                  <div _ngcontent-ewb-c147 class="page-breadcrumb d-none d-sm-flex align-items-center mb-3">_</div>
                  <div _ngcontent-ewb-c147 class="card">
                    <div _ngcontent-ewb-c147 class="card-body">_</div>
                  </div>
                </app-cliente>
              </div>
            </div>
          </div>
        <app-footer _ngcontent-ewb-c114 _ngcontent-ewb-c108></app-footer>
      </div>
      <app-color-switcher _ngcontent-ewb-c114 _ngcontent-ewb-c112></app-color-switcher>
    </app-full-layout>
  </app-root>
  <div class="page-loading">_</div>
  <script src="runtime.js" defer></script>
  <script src="polyfills.js" defer></script>
  <script src="styles.js" defer></script>
  <script src="scripts.js" defer></script>
  <script src="vendor.js" defer></script>
  <script src="main.js" defer></script>
  <svg id="SvgJSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;">_</svg>
  <script type="text/javascript" async defer id="agmGoogleMapsApiScript" src="https://maps.googleapis.com/maps/api/js?v=quarterly&callback=
  mLazyMapsAPILoader&key=AIzaSyDkXXdR0dtqgPV12H12RnUa_1bJcX8CQ04"></script>
</body>
</html>
```

Fuente: (Elaboración propia).

Figura 3.38. Registrar afiliación



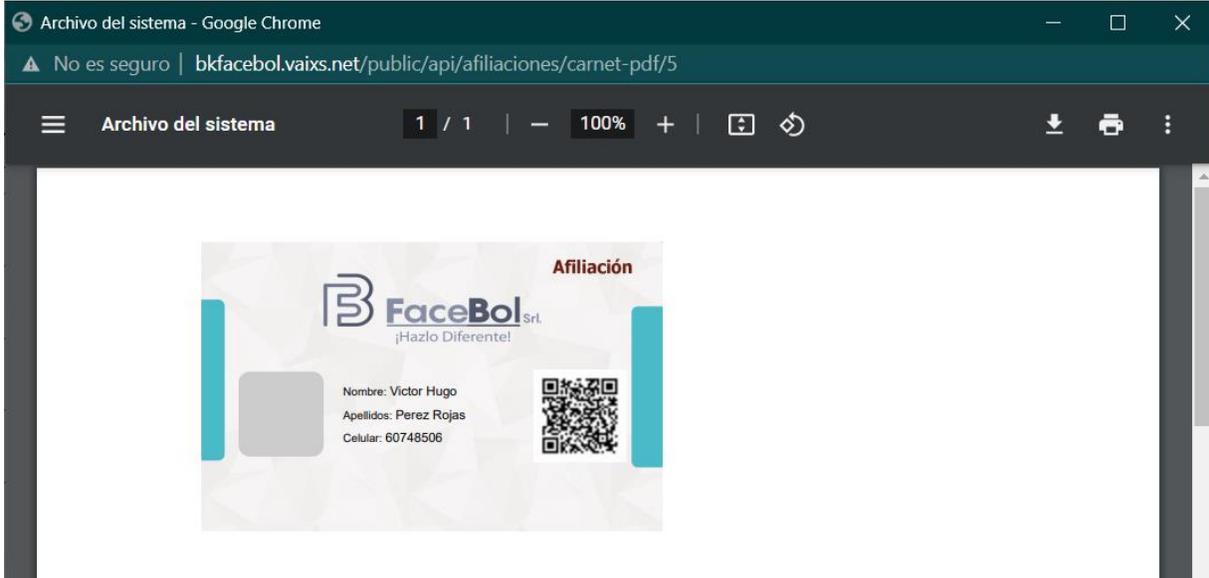
Fuente: (Elaboración propia).

Figura 3.39. Código referente a la vista afiliaciones

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <title>FACEBOL</title>
  </head>
  <body class="pace-done">
    <div class="pace pace-inactive"></div>
    <app-root _ngcontent-ewb-c116 ng-version="12.0.2">
      <router-outlet _ngcontent-ewb-c116></router-outlet>
      <app-full-layout _ngcontent-ewb-c114>
        <div _ngcontent-ewb-c114 class="wrapper" ng-reflect-ng-class="[object Object]">
          <app-sidebar _ngcontent-ewb-c114 _ngcontent-ewb-c189></app-sidebar>
          <app-navbar _ngcontent-ewb-c114 _ngcontent-ewb-c189></app-navbar>
          <div _ngcontent-ewb-c114 class="page-wrapper">
            <div _ngcontent-ewb-c114 class="page-content">
              <router-outlet _ngcontent-ewb-c114></router-outlet>
              <app-afiliacion _ngcontent-ewb-c144>
                <div _ngcontent-ewb-c144 class="page-breadcrumb d-none d-sm-flex align-items-center mb-3"></div>
                <div _ngcontent-ewb-c144 class="card">
                  <div _ngcontent-ewb-c144 class="card-body">
                    <div _ngcontent-ewb-c144 class="d-flex align-items-center mb-4 gap-3">
                      <div _ngcontent-ewb-c144 class="position-relative"></div>
                      <div _ngcontent-ewb-c144 class="ms-auto"></div>
                    </div>
                    <div _ngcontent-ewb-c144 class="table-responsive"></div>
                  </div>
                </div>
              </app-afiliacion>
            </div>
          </div>
        </app-full-layout>
      </app-root>
    </div>
    <div _ngcontent-ewb-c114 class="overlay" ng-reflect-ng-class="[object Object]"></div>
    <app-footer _ngcontent-ewb-c114 _ngcontent-ewb-c188></app-footer>
    <app-color-switcher _ngcontent-ewb-c114 _ngcontent-ewb-c112></app-color-switcher>
  </body>
</html>
```

Fuente: (Elaboración propia).

**Figura 3.40.** Asignar tarjeta de afiliación



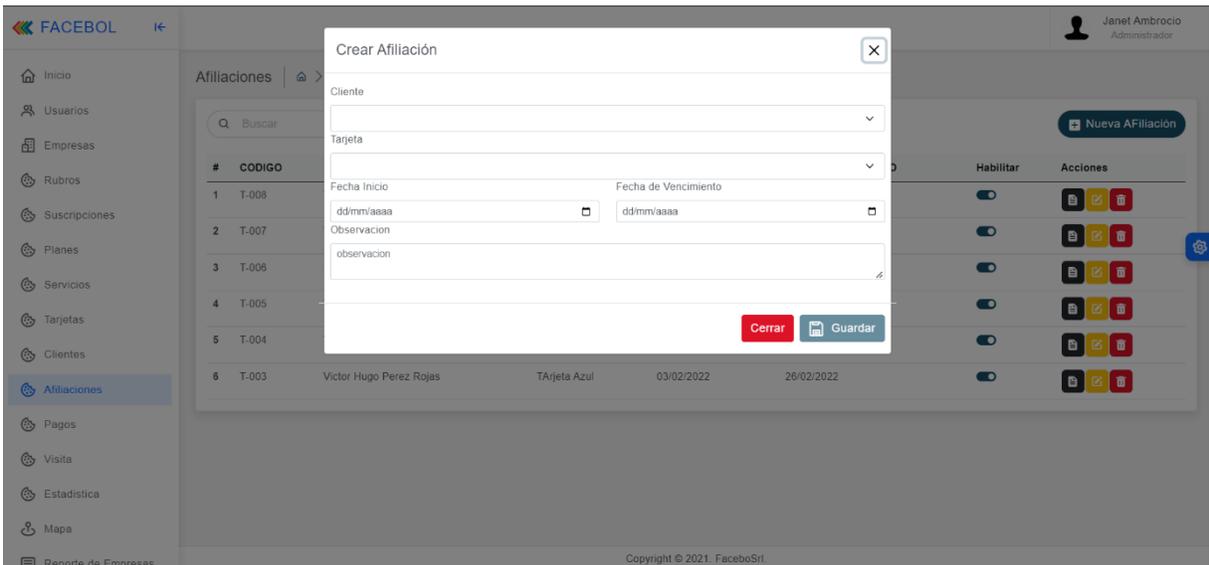
*Fuente: (Elaboración propia).*

**Figura 3.41.** Código referente a la vista tarjetas

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body class="pace pace-inactive">
    <div class="pace pace-inactive">...</div>
    <app-root _ngcontent-ewb-c116 ng-version="12.0.2">
      <router-outlet _ngcontent-ewb-c116>...</router-outlet>
      <app-full-layout _ngcontent-ewb-c114>
        <div _ngcontent-ewb-c114 class="wrapper">
          <app-sidebar _ngcontent-ewb-c114>...</app-sidebar>
          <app-navbar _ngcontent-ewb-c114 _ngcontent-ewb-c109>...</app-navbar>
          <div _ngcontent-ewb-c114 class="page-wrapper">
            <div _ngcontent-ewb-c114 class="page-content-wrapper">
              <div _ngcontent-ewb-c114 class="page-content">
                <router-outlet _ngcontent-ewb-c114>...</router-outlet>
                <app-tarjeta _ngcontent-ewb-c198>
                  <div _ngcontent-ewb-c198 class="page-breadcrumb">...</div>
                  <div _ngcontent-ewb-c198 class="card">
                    <div _ngcontent-ewb-c198 class="card-body">...</div>
                  </div>
                </app-tarjeta>
              </div>
            </div>
          </div>
        </app-full-layout>
      </app-root>
    <div class="page-loading">...</div>
    <script src="runtime.js" defer>...</script>
    <script src="polyfills.js" defer>...</script>
    <script src="styles.js" defer>...</script>
    <script src="scripts.js" defer>...</script>
    <script src="vendor.js" defer>...</script>
    <script src="main.js" defer>...</script>
    <svg id="5vejs5vpl001" width="22" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
  </html>
```

*Fuente: (Elaboración propia).*

Figura 3.42. Registrar pago de afiliación



Fuente: (Elaboración propia).

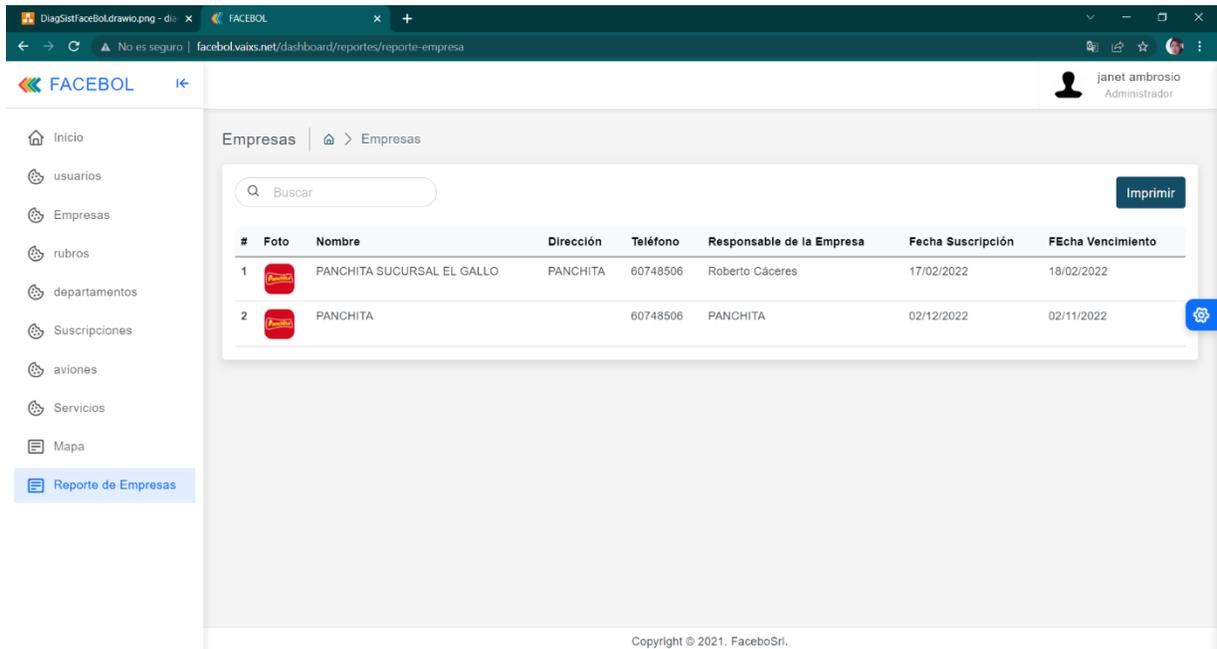
Figura 3.43. Código referente a la vista pagos

```
<!DOCTYPE html>
<html lang="en">
  <head_.../head>
  <body class="pace pace-inactive">
    <div class="pace pace-inactive">
      <app-root_...>
        <router-outlet_...>
          <app-full-layout_...>
            <div_...>
              <app-sidebar_...>
              <app-navbar_...>
              <div_...>
                <div_...>
                  <router-outlet_...>
                    <app-pagos_...>
                      <div_...>
                        <div_...>
                          <div_...>
                            <div_...>
                              <div_...>
                                <table_...>
                                  <thead_...>
                                    <tr_...>
                                      <th_...>
                                      <th_...>
                                      <th_...>
                                      <th_...>
                                      <th_...>
                                      <th_...>
                                    </tr>
                                  </thead>
                                  <tbody_...>
                                </tbody>
                              </table>
                            </div>
                          </div>
                        </div>
                      </app-pagos>
                    </div>
                  <div_...>
                </div>
              </div>
            </app-full-layout>
          </app-root>
        </div>
      </body>
    </html> == $@
```

Fuente: (Elaboración propia).



**Figura 3.46.** Generar reporte de empresas



*Fuente: (Elaboración propia).*

**Figura 3.47.** Código referente a la vista reporte de empresas

```

<!DOCTYPE html>
<html lang="en">
  <head>_</head>
  <body class="pace pace-done">
    <div class="pace pace-inactive"></div>
    <app-root _ngcontent-ewb-c116 ng-version="12.0.2">
      <router-outlet _ngcontent-ewb-c116></router-outlet>
      <app-full-layout _ngcontent-ewb-c116>
        <div _ngcontent-ewb-c114 class="wrapper" ng-reflect-ng-class="[object Object]">
          <app-sidebar _ngcontent-ewb-c114></app-sidebar>
          <app-navbar _ngcontent-ewb-c114 _ngcontent-ewb-c109></app-navbar>
          <div _ngcontent-ewb-c114 class="page-wrapper">
            <div _ngcontent-ewb-c114 class="page-content-wrapper">
              <div _ngcontent-ewb-c114 class="page-content">
                <router-outlet _ngcontent-ewb-c114></router-outlet>
                <app-reporte-empresa _ngcontent-ewb-c225>
                  <div _ngcontent-ewb-c225 class="page-breadcrumb d-none d-sm-flex align-items-center mb-3 noprint"></div>
                  <div _ngcontent-ewb-c225 class="card">
                    <div _ngcontent-ewb-c225 class="card-body">
                      <div _ngcontent-ewb-c225 class="d-lg-flex align-items-center mb-4 gap-3 noprint">
                        <div _ngcontent-ewb-c225 class="position-relative"></div>
                        <div _ngcontent-ewb-c225 class="ms-auto"></div>
                      </div>
                      <div _ngcontent-ewb-c225 class="table-responsive"></div>
                    </div>
                  </div>
                </app-reporte-empresa>
              </div>
            </div>
          </div>
          <app-footer _ngcontent-ewb-c114 _ngcontent-ewb-c108></app-footer>
        </div>
        <app-color-switcher _ngcontent-ewb-c114 _ngcontent-ewb-c112></app-color-switcher>
      </app-full-layout>
    </app-root>
  </div>
  <div class="page-loading"></div>
  <script src="runtime.js" defer></script>
  <script src="polyfills.js" defer></script>
  <script src="styles.js" defer></script>
  <script src="scripts.js" defer></script>
  <script src="vendor.js" defer></script>
  <script src="main.js" defer></script>
  <svg id="svgjsSvg1001" width="2" height="0" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;"></svg>
  <script type="text/javascript" async defer id="agmGoogleMapsApiScript" src="https://maps.googleapis.com/maps/api/js?v=quarterly&callback=agmLazyMapsAPILoader&key=AIzaSyDKKKd0dtaqW12H728nJa_1bjcxBEOod"></script>
</body>
</html>

```

*Fuente: (Elaboración propia).*

Figura 3.48. Generar reporte de empresas suscritas

#	EMPRESA	PLAN	FECHA INICIO	FECHA VENCIMIENTO	monto
1	fideos	ESTÁNDAR	02/03/2022	02/03/2023	300
2	PANCHITA SUCURSAL EL GALLO	ESTÁNDAR	17/02/2022	18/02/2022	300
3	PANCHITA	GRATUITO	02/12/2022	02/11/2022	0

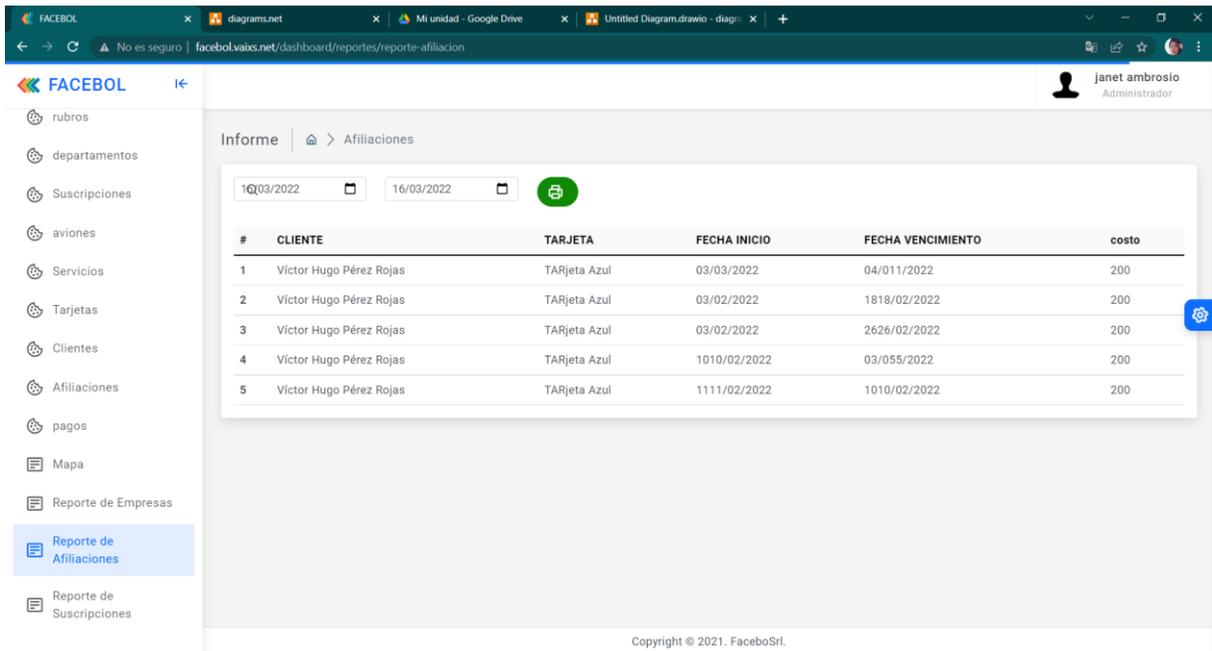
Fuente: (Elaboración propia).

Figura 3.49. Código referente a la vista reporte de suscripciones

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <body class="pace-done">
      <div class="pace pace-inactive">
        <app-root _ngcontent-ewb-c116 ng-version="12.0.2">
          <router-outlet _ngcontent-ewb-c116>
            <app-full-layout _ngcontent-ewb-c114>
              <div _ngcontent-ewb-c114 class="page-wrapper" ng-reflect-ng-class="[object Object]">
                <app-sidebar _ngcontent-ewb-c114>
                  <app-navbar _ngcontent-ewb-c114 _ngcontent-ewb-c189>
                    <div _ngcontent-ewb-c114 class="page-wrapper">
                      <div _ngcontent-ewb-c114 class="page-content-wrapper">
                        <div _ngcontent-ewb-c114 class="page-content">
                          <router-outlet _ngcontent-ewb-c114>
                            <app-report-suscripcion _ngcontent-ewb-c227>
                              <div _ngcontent-ewb-c227 class="page-breadcrumb d-none d-sm-flex align-items-center mb-3">
                                <div _ngcontent-ewb-c227 class="card">
                                  <div _ngcontent-ewb-c227 class="card-body">
                                    </div>
                                  </div>
                                </div>
                              <div _ngcontent-ewb-c114 class="overlay" ng-reflect-ng-class="[object Object]">
                                </div>
                              </div>
                            </div>
                          <app-footer _ngcontent-ewb-c114 _ngcontent-ewb-c188>
                            </div>
                        </div>
                      <app-color-switcher _ngcontent-ewb-c114 _ngcontent-ewb-c112>
                        </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </app-root>
      </div>
    </body>
  </html>
```

Fuente: (Elaboración propia).

**Figura 3.50.** Generar reporte de afiliaciones



*Fuente: (Elaboración propia).*

**Figura 3.51.** Código referente a la vista reporte de afiliaciones

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>FACEBOL</title>
  </head>
  <body class="pace pace-inactive">
    <div class="app-root _ngcontent-ewb-c116 ng-version="12.0.2">
      <router-outlet _ngcontent-ewb-c116></router-outlet>
      <app-full-layout _ngcontent-ewb-c114>
        <div _ngcontent-ewb-c114 class="wrapper" ng-reflect-ng-class="[object Object]">
          <app-sidebar _ngcontent-ewb-c114></app-sidebar>
          <app-navbar _ngcontent-ewb-c114 _ngcontent-ewb-c109></app-navbar>
          <div _ngcontent-ewb-c114 class="page-wrapper">
            <div _ngcontent-ewb-c114 class="page-content-wrapper">
              <div _ngcontent-ewb-c114 class="page-content">
                <router-outlet _ngcontent-ewb-c114></router-outlet>
                <app-reportes-afiliacion _ngcontent-ewb-c224>
                  <div _ngcontent-ewb-c224 class="page-breadcrumb d-none d-sm-flex align-items-center mb-3"></div>
                  <div _ngcontent-ewb-c224 class="card">
                    <div _ngcontent-ewb-c224 class="card-body">
                      <div _ngcontent-ewb-c224 class="d-lg-flex align-items-center mb-4 gap-3"></div>
                      <div _ngcontent-ewb-c224 class="table-responsive">
                        <table border="1">
                          <thead>
                            <tr>
                              <th>#</th>
                              <th>CLIENTE</th>
                              <th>TARJETA</th>
                              <th>FECHA INICIO</th>
                              <th>FECHA VENCIMIENTO</th>
                              <th>costo</th>
                            </tr>
                          </thead>
                          <tbody>
                            <tr>
                              <td>1</td>
                              <td>Victor Hugo Pérez Rojas</td>
                              <td>TARjeta Azul</td>
                              <td>03/03/2022</td>
                              <td>04/011/2022</td>
                              <td>200</td>
                            </tr>
                            <tr>
                              <td>2</td>
                              <td>Victor Hugo Pérez Rojas</td>
                              <td>TARjeta Azul</td>
                              <td>03/02/2022</td>
                              <td>1818/02/2022</td>
                              <td>200</td>
                            </tr>
                            <tr>
                              <td>3</td>
                              <td>Victor Hugo Pérez Rojas</td>
                              <td>TARjeta Azul</td>
                              <td>03/02/2022</td>
                              <td>2626/02/2022</td>
                              <td>200</td>
                            </tr>
                            <tr>
                              <td>4</td>
                              <td>Victor Hugo Pérez Rojas</td>
                              <td>TARjeta Azul</td>
                              <td>1010/02/2022</td>
                              <td>03/055/2022</td>
                              <td>200</td>
                            </tr>
                            <tr>
                              <td>5</td>
                              <td>Victor Hugo Pérez Rojas</td>
                              <td>TARjeta Azul</td>
                              <td>1111/02/2022</td>
                              <td>1010/02/2022</td>
                              <td>200</td>
                            </tr>
                          </tbody>
                        </table>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
          <app-footer _ngcontent-ewb-c114 _ngcontent-ewb-c108></app-footer>
          <app-color-switcher _ngcontent-ewb-c114 _ngcontent-ewb-c112></app-color-switcher>
        </div>
      </div>
    </div>
    <div class="page-loading"></div>
    <script src="runtime.js" defer></script>
    <script src="polyfills.js" defer></script>
    <script src="styles.js" defer></script>
    <script src="scripts.js" defer></script>
    <script src="vendor.js" defer></script>
    <script src="main.js" defer></script>
    <svg id="SvgjsSvg1001" width="2" height="2" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:svgjs="http://svgjs.com/svgjs" style="overflow: hidden; top: -100%; left: -100%; position: absolute; opacity: 0;"></svg>
    <script type="text/javascript" async defer id="ag40001e4apsAPIscript" src="https://maps.googleapis.com/maps/api/js?v=quarterly&callback=jQueryMapsAPIloader&key=AIzaSyBxkddt0tngEPlv1zrh0la_ibjcdC00d"></script>
  </body>
</html>

```

*Fuente: (Elaboración propia).*

### 3.4.4. Fase de Pruebas

#### Pruebas de aceptación de Iteración 1

Tabla 3.5. Prueba de Aceptación Iteración 1

ID	HISTORIA DE USUARIO	RESULTADO ESPERADO	RESULTADO
H001	Registrar usuarios al sistema	Validar formulario registro con campos requeridos	Correcto
		Encriptar la contraseña del usuario	Correcto
		Mostrar mensaje de confirmación de usuario registrado	Correcto
		Editar información del usuario	Correcto
		Habilitar usuario	Correcto
		Dar de baja usuario	Correcto
H002	Asignar roles a los usuarios	Mostrar los roles para seleccionar	Correcto
		Cambiar rol de usuario	Correcto
H003	Inicio y cierre de sesiones	Inicio de sesión con usuario y contraseña correctos	Correcto
		Inicio de sesión con usuario y contraseña incorrectos, mostrar mensaje.	Correcto

		Ocultar texto de la contraseña al introducir los datos del usuario	Correcto
H004	Administrar rubros	Mostrar formulario para añadir rubro	Correcto
		Mostrar un listado de los rubros registrados	Correcto
		Mostrar opción de búsqueda de rubros	Correcto
H005	Administrar departamento	Mostrar formulario para añadir departamento	Correcto
		Mostrar un listado de los departamentos registrados	Correcto
		Mostrar opción de búsqueda de departamentos	Correcto
H006	Administrar planes	Mostrar formulario para añadir plan	Correcto
		Mostrar un listado de los planes registrados	Correcto
		Mostrar opción de búsqueda de plan	Correcto
H007	Registrar empresas	Visualizar un formulario para registrar empresa	Correcto
		Modificar datos de la empresa	Correcto
		Mostrar listado de empresas registradas	Correcto

H008	Registrar ubicación de empresas	Visualizar un formulario para registrar la ubicación de la empresa	Correcto
		Modificar datos de ubicación de la empresa	Correcto
H009	Registrar servicios de empresa	Visualizar un formulario para registrar servicios de empresa	Correcto
		Modificar servicios de la empresa	Correcto
		Mostrar listado de servicios de la empresa	Correcto
H010	Registrar suscripción de empresas	Visualizar un formulario para registrar suscripción	Correcto
		Modificar suscripción	Correcto
		Mostrar listado de suscripción de empresas	Correcto
H011	Registrar servicios de marketing	Visualizar un formulario para registrar servicios de marketing	Correcto
		Modificar servicios	Correcto
		Mostrar listado de servicios	Correcto

H012	Designar plan	Visualizar formulario de plan	Correcto
		Mostrar planes	Correcto
H013	Generar boleta de suscripción	Boleta de suscripción	Correcto
		Mostrar opción para imprimir boleta	Correcto

*Fuente: (Elaboración propia).*

### Pruebas de Aceptación Iteración 2

**Tabla 3.6.** Prueba de Aceptación Iteración 2

ID	HISTORIA DE USUARIO	RESULTADO ESPERADO	RESULTADO
H014	Registrar datos de cliente	Mostrar formulario de registro para datos de clientes	Correcto
		Mostrar registro de los clientes	Correcto
H015	Registrar afiliación	Mostrar formulario para registrar afiliación	Correcto
		Mostrar registro de afiliados	Correcto
H016	Asignar tarjeta de afiliación	Mostrar formulario para registrar tarjeta	Correcto
		Mostrar registro de tarjetas	Correcto

		Mostrar opción para imprimir tarjeta	Correcto
H017	Registrar pago de afiliación	Mostrar listado de pagos por afiliación	Correcto
		Generar boleta de afiliación	Correcto
H018	Registrar pago de suscripción	Visualizar formulario de registro de pagos por suscripción	Correcto
		Mostrar listado de pagos por suscripción	Correcto
		Generar boleta de pago por suscripción	Correcto
H019	Generar comprobante de pago	Mostrar comprobante de pagos	Correcto
		Mostrar una opción para imprimir comprobante	Correcto
H020	Generar reporte de empresas	Mostrar reporte general de empresas registradas	Correcto
		Mostrar una opción para imprimir reporte	Correcto
H0201	Generar reporte de empresas suscritas	Mostrar reporte general de empresas suscritas	Correcto
		Mostrar una opción para imprimir reporte	Correcto

H022	Generar reporte de afiliaciones	Mostrar reporte general de afiliados	Correcto
		Mostrar una opción para imprimir reporte	Correcto
H023	Realizar verificación de tarjetas	Mostrar tarjeta para escanear	Correcto
H024	Generar datos estadísticos	Mostrar visitas con la cantidad de visitas	Correcto

**Fuente:** (Elaboración propia).

**CAPÍTULO IV**  
**METRICAS DE CALIDAD, COSTO Y**  
**SEGURIDAD**

## **4.1. INTRODUCCIÓN**

En el presente capítulo se muestran los resultados de las pruebas de evaluación y resultados de acuerdo a métricas de calidad de software, análisis de costos, seguridad en el software y pruebas al software.

## **4.2. MÉTRICA DE CALIDAD**

### **4.2.1. Factores de calidad ISO/IEC 25000**

Se hará la medición de calidad de software mediante la 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 distintas partes que establece cualquier componente de calidad de software puede ser descrito en términos 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.

#### **4.2.1.1. Implementación de la ISO/IEC 25010**

- **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.

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.

- ✓ **Número de entrada de usuarios:** Se cuenta cada entrada de usuario que proporciona datos al sistema.

- ✓ **Número de salidas de usuario:** Se refiere cada salida que proporciona el sistema al usuario. Entre estos pueden ser 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, que sean parte de una base de datos, o un archivo independiente.
- ✓ **Numero de interfaz externas:** se cuenta las interfaces legibles por la máquina.

Para calcular el punto de función se utilizará la siguiente formula:

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

Dónde:

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 4.1.** Parámetros de Medida y su Cantidad

<b>PARÁMETROS DE MEDIDA</b>	<b>CANTIDAD</b>
Número de entrada de usuario	20
Número de salidas de usuario	18
Número de peticiones de usuario	20
Número de archivo	12
Número de interfaz externa	3

*Fuente: (Elaboración propia).*

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

**Tabla 4.2.** Parámetros del punto de función.

PARAMETROS DE MEDIDA	CANTIDAD	FACTOR DE PONDERACION	TOTAL
N° de entradas de usuario	20	*5	100
N° de salidas de usuario	18	*5	90
N° de peticiones de usuario	20	*4	90
Número de archivo	12	*9	108
Número de interfaz externa	3	*6	18
<b>CUENTA TOTAL</b>			<b>= 406</b>

*Fuente: (Elaboración propia)*

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

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

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

Dónde:

$$PF = 406 * (0,65 + 0,01 * 55)$$

$$PF = 406 * 1,2$$

$$PF = 487,2$$

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

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

$$PF_{\max} = 406 * (0,65 + 0.01 * 70)$$

$$PF_{\max} = 406 * (0,65 + 0.01 * 70)$$

$$PF_{\max} = 406 * 1,35$$

$$PF_{\max} = 548,1$$

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

$$\text{Adecuacion funcional} = \frac{487,2}{548,1} * 100\%$$

$$\text{Adecuacion funcional} = 0,88 * 100\%$$

$$\text{Adecuacion funcional} = 88\%$$

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.

- **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:

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

Dónde:

**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 10 min continuas se logró llenar la siguiente tabla:

**Tabla 4.3.** Valores de fiabilidad de cada modulo

N°	MODULOS	$\lambda$	t	R(t)
1	Modulo Usuarios	0.001	4Hrs	0.95
2	Modulo Configuración	0.022	4Hrs	0.95
3	Modulo Empresas	0.025	4Hrs	0.99
4	Modulo Suscripción	0.018	4Hrs	0.97
5	Modulo Pagos	0.012	4Hrs	0.90
6	Modulo Reportes	0.018	4Hrs	0.98

*Fuente: (Elaboración propia)*

La fiabilidad del sistema estaría dada por la formula.

$$Fiabilidad = R_s * R_p$$

Dónde:

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

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:

$$R_p = \frac{\sum_{i=2}^5 (R_i)}{\sum_{i=2}^5 P_i} = \frac{0,95 + 0,99 + 0,97 + 0,90 + 0,98}{5} = \frac{4,79}{5} = 0,958$$

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

$$Fiabilidad = R_s * R_p = 0,95 * 0,958 = 0,91 = 91\%$$

De lo cual se puede decir que existe un 9 % de probabilidad de que el sistema presente algún fallo cuando se exceda un tiempo de uso continuo, debido a que puedan existir fallas.

- **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:

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

Dónde:

$X_i$ : Es la sumatoria de valores

$n$ : Es el número de preguntas

Para responder a las preguntas, se debe considerar la siguiente tabla:

**Tabla 4.4.** Parámetros de medición

<b>ESCALA</b>	<b>VALOR</b>
Muy Bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

**Fuente:** (Elaboración propia)

Calculamos la usabilidad con la ecuación anterior:

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

$$FU = [0.94 * 100]$$

$$FU = 94\%$$

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

- **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:

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

Dónde:

$M_t$ : 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 4.5.** 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:

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

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

$$IMS = 0,88 * 100\% = \mathbf{88\%}$$

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.

- **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, está 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 4.6.** Factor de portabilidad

<b>FACTOR DE PORTABILIDAD</b>	<b>VALOR EN %</b>
Puede ser transferido de un entorno a otro	90
Se puede adaptar a otros ambientes con factibilidad	100

Es fácil de instalar	100
<b>TOTAL</b>	<b>96%</b>

*Fuente: (Elaboración propia)*

- **Calidad Global**

Una vez calculado 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 4.7.** Resultados

ATRIBUTO	VALOR DE (%)
FUNCIONALIDAD	88
FIABILIDAD	91
USABILIDAD	94
MANTENIBILIDAD	88
PORTABILIDAD	96
<b>CALIDAD GLOBAL</b>	<b>457/5 = 91,4 %</b>

*Fuente: (Elaboración propia)*

### 4.3. ANALISIS DE COSTOS

#### 4.3.1. Aplicación del COCOMO II Nivel Intermedio

Para estimar el costo del desarrollo de software se aplicó COCOMO II nivel intermedio.

**Tabla 4.8.** Factores de Costos

Factores (cost-drivers)	Valor de los factores					
	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra
<b>Atributos 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	1,21	1,56
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 los analistas	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	1,07	1,00	0,95		
Calidad de los 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	

**Fuente:** (Elaboración propia)

### I. Estimación de esfuerzo

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

$$E = 3,2 * 4^{1,05} * 0,53 = 7,34 \text{ Personas/mes}$$

### II. Estimación del tiempo

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

$$\text{Entonces: } T = 2,5 * 7,34^{0,38} = 5,33 \Rightarrow 5 \text{ Meses}$$

Con el dato anterior hallaremos el resto de variables

### III. Otras estimaciones

#### Número de personas requerido para el desarrollo

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

Entonces:  $NP = \frac{7,34}{5,33} = 1.37 \Rightarrow 2 \text{ Personas}$

#### Costo total del proyecto

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

Entonces:  $CT = 2 * 5 * 2250 = 22.500\text{Bs.}$

#### Costo por línea de código

$$LDC = \frac{CT}{KLDC} \text{ Bs.}$$

Entonces:  $LDC = \frac{22.500}{4} = 5625 \text{ Bs.}$

#### 4.3.2. Costos del Hardware

La empresa FaceBol S.R.L. cuenta con el equipo completo de servidor para subir el presente sistema. Para ello se cuenta con dos equipos, primero equipo de escritorio para el área de administración y segundo un equipo portátil para el área comercial.

Otros costos:

- **Costo del software:** se describe lo siguiente:

**Tabla 4.9.** Tabla general de costos

<b>Costos del software</b>	<b>Bs.</b>
Software desarrollado	5625
<b>Costo de servidor</b>	
Hosting dominio	389
<b>Otros costos</b>	
Internet	219
<b>TOTAL</b>	<b>6.233 Bs</b>

*Fuente: (Elaboración Propia)*

- **Costo total del sistema**

**Tabla 4.10.** Costo total del sistema

<b>Descripción de costos</b>	<b>Total (Bs)</b>
Costo total de Software desarrollado	6.233
Costo total del proyecto	22.500
<b>TOTAL</b>	<b>28.733 Bs.</b>

*Fuente: (Elaboración Propia)*

#### **4.4. SEGURIDAD**

##### **4.4.1. Seguridad Web**

La seguridad en Web tiene 3 etapas primarias:

- Seguridad de la computadora del usuario.

- El usuario debe tener una contraseña al momento de ingresar a su equipo como medida de seguridad.
- El equipo del usuario debe tener antivirus, para proteger la información en su equipo.
- El equipo debe tener activado el firewall, para un control de amenazas externas.
- Seguridad del servidor Web.
- Seguridad con contraseña segura para el acceso a la base de datos y el servidor web.
- Seguridad de la información que viaja entre el servidor Web y el usuario.
- Cifrar datos confidenciales con contraseñas fuertes.
- Actualizar el firmware.
- Actualizar antivirus.
- Realizar copias de seguridad de la información.

#### **4.4.2. Seguridad de password con Bcrypt**

Una contraseña o clave (en inglés password) es una forma de autenticar al acceso de la información secreta para el control del mismo. Por tanto, la contraseña debe ser único y directo a aquellos que tengan el permiso para acceder a la información.

- Hash

La fachada de Laravel hash proporciona hash Bcrypt y Argon2 seguro para almacenar contraseñas de usuario.

#### **4.4.3. Seguridad con Cifrado de datos con Laravel**

El cifrado de Laravel usa OpenSSL para proporcionar cifrado AES-256 y AES-128.

Todos los valores encriptados de Laravel están firmados usando un código de autenticación de mensajes (MAC) para que su valor subyacente no se pueda modificar una vez encriptados.

La encriptación o cifrado de datos protege información secreta de la empresa, la misma se realiza mediante el mecanismo de transformar una cadena en otra. De

esta forma se obtiene una cadena encriptada, donde se muestra una cadena con diferentes valores a la original.

#### 4.4.4. Niveles de acceso al sistema

**Tabla 4.11.** Nivel de acceso al sistema de parte de los usuarios

OPERACIONES	ADMINISTRADOR	EMPRESA	ASESOR COMERCIAL	ASESOR DE MARKETING
ALTAS	✓			
BAJAS	✓			
MODIFICACIONES	✓		✓	✓
CONSULTAS Y REPORTES	✓	✓	✓	✓

*Fuente: (Elaboración Propia)*

### 4.5. PRUEBAS AL SOFTWARE

#### 4.5.1. Prueba de Caja Blanca

Para las pruebas de caja blanca se proporciona una medición cuantitativa de la complejidad lógica del programa. La complejidad ciclomática de un grado de flujo  $V(G)$  establece el número de caminos independientes.

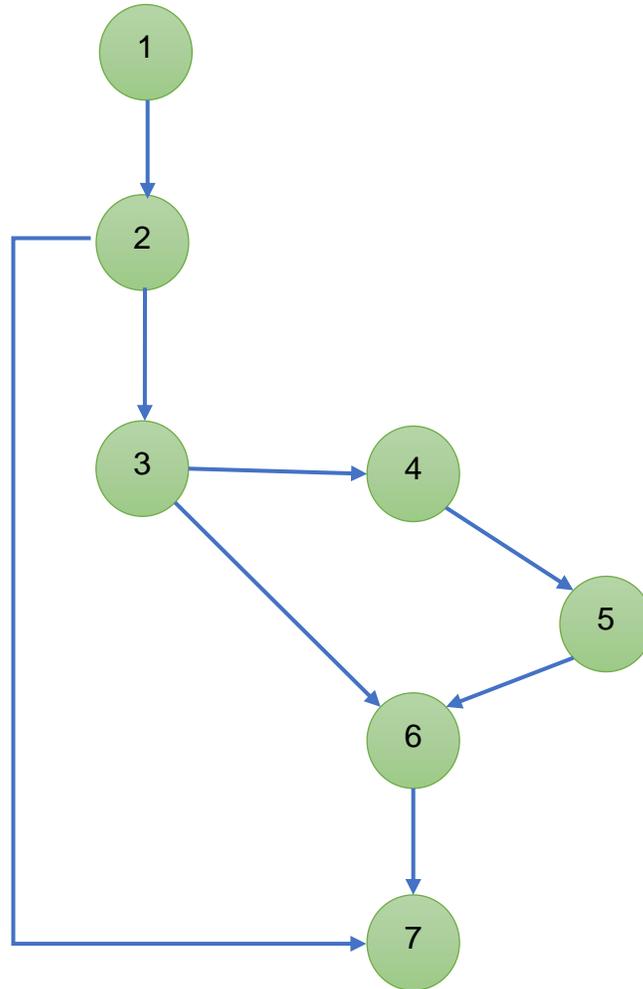
- El número de regiones del gráfico de flujo.

Los casos de prueba son los siguientes:

- $V(G) = A - N + 2$ , donde A es el número de aristas y N es el número de nodos.

## ❖ Gestión de Usuarios

Figura 4.1. Grafo de Flujo del Módulo Usuario



**Fuente:** (Elaboración Propia)

Dónde:

- Página iniciar sesión (1)
- Verifica ingreso de Usuario y Contraseña (2)
- Carga mensaje de campos obligatorios (3)
- Verificación de Usuarios y Contraseña (4)
- Carga permisos según el rol (5)
- Vuelve a cargar la página de iniciar sesión (6)
- Fin de ciclo (7)

Empleamos la formula siguiente:

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

Reemplazamos valores obtenidos:

$$V(G) = 8 - 7 + 2$$

$$V(G) = 3$$

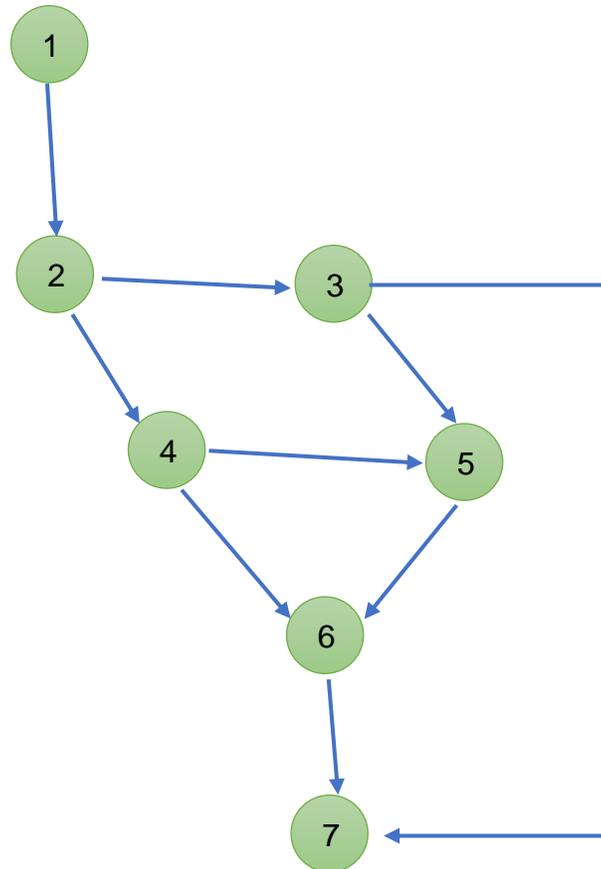
**Tabla 4.12.** Casos de prueba

<b>N° Caminos</b>	<b>Ruta</b>	<b>Casos de prueba</b>
1	1,2,3,4,5,6,7	Se verifica si el usuario ingreso todos los datos, si no ingreso alguno de los datos el sistema carga mensaje de campos obligatorios.
2	1,2,3,6,7	El sistema verifica los datos ingresados, si es correcto identifica el tipo de usuario y carga sus permisos correspondientes, permitiéndole ingresar al sistema.
3	1,2,7	El sistema verifica los datos ingresados, si es incorrecto vuelve a cargar la página de iniciar sesión, denegando acceso al sistema.

**Fuente:** (Elaboración propia)

## ❖ Gestión de Empresas

Figura 4.2. Grafo de flujo del Módulo Empresas



**Fuente:** (Elaboración Propia)

Dónde:

- Iniciar sesión (1)
- Página empresa (2)
- Registra datos de empresa (3)
- Carga mensaje de campos obligatorios (4)
- Verificación de campos llenados (5)
- Muestra empresa registrada (6)
- Fin de ciclo (7)

Empleamos la formula siguiente:

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

Reemplazamos valores obtenidos:

$$V(G) = 9 - 7 + 2$$

$$V(G) = 4$$

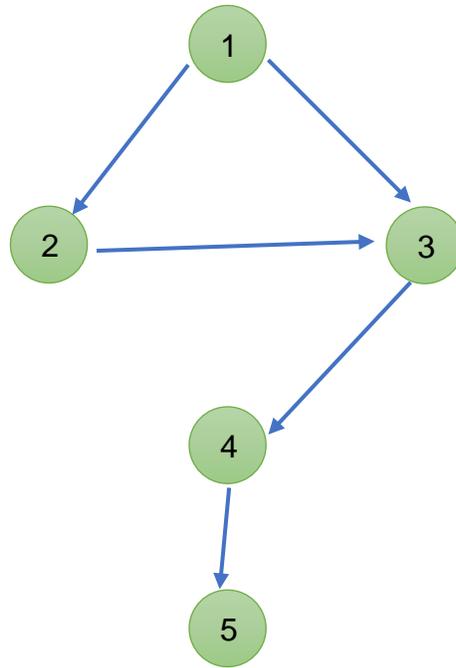
**Tabla 4.13.** Casos de prueba

<b>N° Caminos</b>	<b>Ruta</b>	<b>Casos de prueba</b>
1	1,2,3,5,6,7	Ingresar como usuario empresa,.
2	1,2,3,5,6,7	El sistema registra los datos ingresados, si es correcto identifica el tipo de empresa y carga sus permisos correspondientes, permitiéndole ingresar al sistema.
3	1,2,4,5,6,7	El sistema verifica que los campos de registro estén llenos, si no esta no permite registrar empresa.
4	1,2,3,7	El sistema verifica los datos ingresados, si es incorrecto vuelve a cargar la página de empresa.

**Fuente:** (Elaboración propia)

## ❖ Gestión de Reportes

Figura 4.3. Grafo de flujo del Módulo Reportes



**Fuente:** (Elaboración Propia)

Dónde:

- Página reportes (1)
- Mostrar listado de empresas (2)
- Actualizar vista de reportes (3)
- Imprimir reporte (4)
- Fin de ciclo (5)

Empleamos la formula siguiente:

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

Reemplazamos valores obtenidos:

$$V(G) = 5 - 5 + 2$$

$$V(G) = 2$$

**Tabla 4.14.** Casos de prueba

N° Caminos	Ruta	Casos de prueba
1	1,2,3,4,5	Se ingresa al módulo de reportes.
2	1,3,4,5	Se genera reporte, para imprimir

*Fuente: (Elaboración propia)*

#### 4.5.2. Prueba de Caja Negra

Se trata de un enfoque que intenta descubrir diferentes tipos de errores que no se encuentran con los métodos de caja blanca.

Análisis de Valores Limites

**Tabla 4.15.** Caso de Prueba al Módulo Usuarios

N°	Propósito de prueba	Caso de prueba			
		Dato de entrada 1	Dato de entrada 2	Dato de entrada 3	Dato de entrada 4
1	Que el módulo entregue resultados erróneos.	Ingresar datos nulos en la tabla	Ingresar una cadena no válida, tal que devuelva error.	Ingresar un email no válida, tal que devuelva error.	Ingresar un nombre ya existente, tal que no permita registrar repetidos.

*Fuente: (Elaboración propia)*

**Tabla 4.16.** Caso de Prueba al Módulo Empresas

N°	Propósito de prueba	Caso de prueba			
		Dato de entrada 1	Dato de entrada 2	Dato de entrada 3	Dato de entrada 4
1	Que el módulo entregue resultados erróneos.	Ingresar datos nulos en la tabla	Ingresar una cadena no válida, tal que devuelva error.	Ingresar un email no válida, tal que devuelva error.	Ingresar una empresa ya existente, tal que devuelva dato existente.

*Fuente: (Elaboración propia)*

**Tabla 4.17.** Caso de Prueba al Módulo Reportes

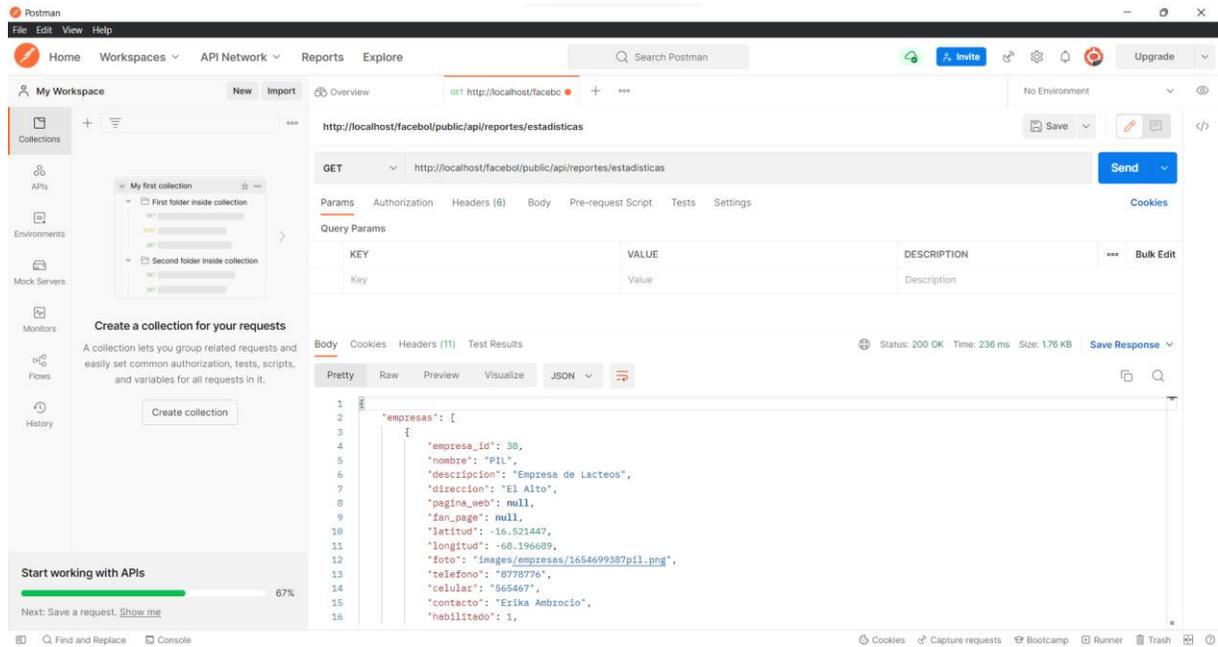
N°	Propósito de prueba	Caso de prueba			
		Dato de salida 1	Dato de salida 2	Dato de salida 3	Dato de salida 4
1	Que el módulo entregue resultados erróneos.	Verificar listado de todas las empresas.	Actualizar listado de registros nuevos.	Generar reporte de empresas.	Imprimir reporte, tal que no se muestre todos los registros, actualizar tabla.

*Fuente: (Elaboración propia)*

#### 4.5.3. Prueba de Estrés

La prueba de estrés nos permitirá medir el comportamiento de nuestro sistema bajo condiciones de carga extrema. Para esto usaremos la herramienta Postman, que nos permitirá simular varias peticiones, para ello se realizó la prueba de estrés en el módulo empresa, afiliación y suscripción.

**Figura 4.4. Prueba al Módulo Empresa**



**Fuente:** (Elaboración propia)

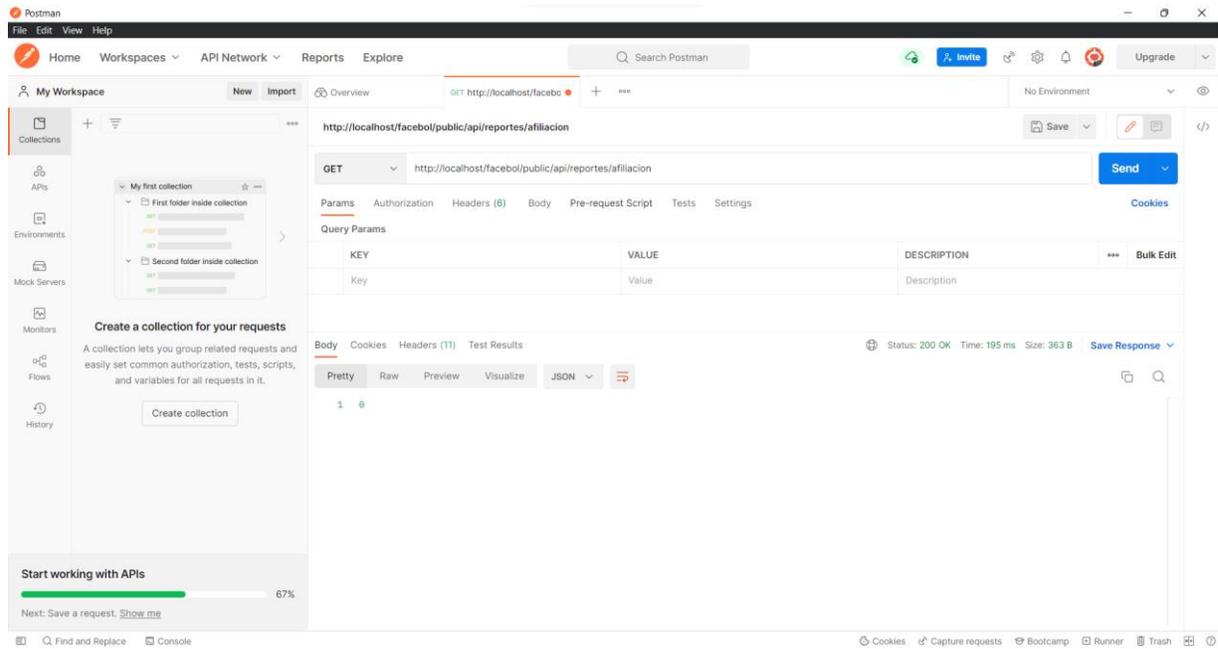
En la prueba de suscripciones se tiene un status óptimo de 200k, sin margen de error, con un rendimiento de 171 ms

**Tabla 4.18. Resultados**

Nº	Muestra KB	Status	Rendimiento ms
1	363	200 OK	171
2	363	200 OK	204

**Fuente:** (Elaboración propia)

**Figura 4.5.** Prueba al Módulo de afiliación



**Fuente:** (Elaboración propia)

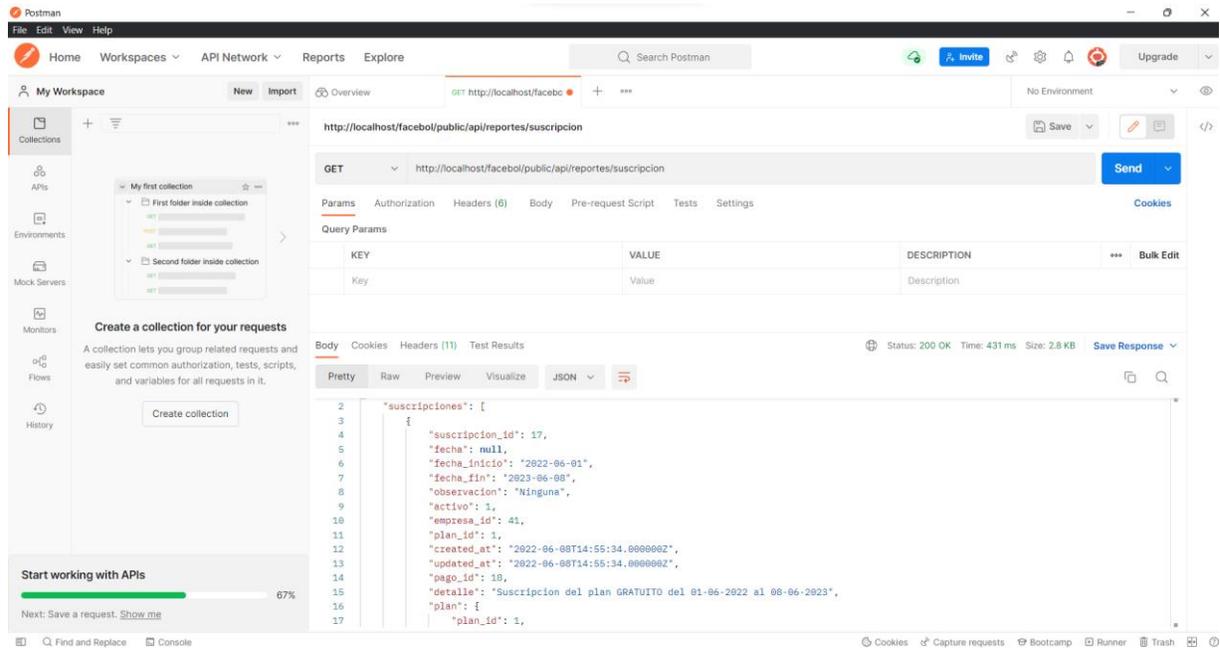
En la prueba de suscripciones se tiene un status óptimo de 200k, sin margen de error, con un rendimiento de 174 ms.

**Tabla 4.19.** Resultados

Nº	Muestra KB	Status	Rendimiento ms
1	363	200 OK	174
2	363	200 OK	204

**Fuente:** (Elaboración propia)

**Figura 4.6.** Prueba al Módulo de Suscripciones



*Fuente: (Elaboración propia)*

En la prueba de suscripciones se tiene un status óptimo de 200k, sin margen de error, con un rendimiento de 431 ms.

**Tabla 4.20. Resultados**

Nº	Muestra KB	Status	Rendimiento ms
1	2.8	200 OK	431
2	2.8	200 OK	651

*Fuente: (Elaboración propia)*

**CAPITULO V**  
**CONCLUSIONES Y**  
**RECOMENDACIONES**

## **5.1. CONCLUSIONES Y RECOMENDACIONES**

Con los resultados obtenidos al implementar el SISTEMA WEB DE INFORMACIÓN GEOGRÁFICA ORIENTADA AL MARKETING PARA LAS EMPRESAS EN CONVENIO PARA FACEBOL S.R.L. se tiene el siguiente análisis:

### **5.1.1. CONCLUSIONES**

- Se llegó a realizar el análisis de la situación actual de la institución, para tener referencia de su estado.
- Se logró esquematizar la estructura de funcionamiento del sistema, para tener una idea de los procesos que cumple para su ejecución.
- Se logró centralizar la información recolectada en una base de datos, para una mejor gestión de nuevos registros al sistema.
- Se logró desarrollar los módulos del sistema para mejorar la organización en el trabajo del sistema.
- Se logró hacer uso de APIs de Google Maps, para georreferenciar a las empresas en convenio, vista en mapa de todas las entidades.
- Se logró satisfactoriamente implementar el sistema y sus respectivas pruebas de software.
- Se logró automatizar la generación de reportes, con información que pueda utilizar la empresa y los administradores para una mejor gestión de información.

### **5.1.2. RECOMENDACIONES**

- Se sugiere incorporar módulo al sistema para la gestión de publicidad de las empresas.
- Se recomienda para trabajos futuros implementar PostgreSQL, para tener un mayor espacio en el almacenamiento de datos.
- Se recomienda un encargado para el soporte técnico del sistema.
- Se propone insertar firma digital a los comprobantes de afiliación y convenios, para agilizar el trabajo de la empresa.

# **REFERENCIAS BIBLIOGRAFICAS**

## REFERENCIAS BIBLIOGRAFICAS

- Abud, F. M. (2012). Normas ISO 25000. *Revista Upiicsa*. Recuperado el 16 de Noviembre de 2021, de ISO IEC 25010: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- Arias, M. Á. (2017). *Aprende Programación con PHP y MySQL*. 2da Edición. IT.
- Asociación de Investigación, e. S. (2006). *Modelo Ágil y Programación Extrema*. AISI.
- Aubry, C., & Van, L. L. (2017). *jQuery*. 3ra Edición. ENI.
- Beck, K., & Andrés, C. (2012). *Programación Extrema*. Bostón: 2da Edición. Adsison-Wesley.
- Bermejo, M. (2006). Creative Commons.
- Bermejo, M. (2010). *El Kanban*. Creative Commons.
- Boada, O. M., & Gómez, G. J. (2018). *El Gran Libro de Angular*. Marcombo.
- Boehm, B. (1981). *Modelo CO.Co.Mo*. Moreno.
- Bustamante, D., & Rodriguez, J. (2014). *Metodología XP*. Barinas.
- Casado, R. (2019). *Introduccion a HTML*. Universidad Salamanca.
- Cíceri, V. M. (2018). *Introducción a Laravel*. Buenos Aires: Six Ediciones.
- FaceBol SRL. (2018). *FaceBol SRL*. Obtenido de FaceBol SRL: <https://www.facebolsrl.com/>
- Fenton, N. (1991). *Cocomo*. Bindex.
- Florián, G. B. (2007). *Manual Rápido de CSS*. Editorial Web.
- Francisco, J., & M., J. (2021). *implementacion de la norma ISO 25010*. Edit. Film.
- Garcia, P. F. (2022). *Procesos y Métodos de Modelado para la Ingeniería Web y Web Semántica*. GRIAL.
- Gauchat, J. D. (2013). *El Gran Libro de HTML5, CSS3 y JavaScript*. 2da Edición. Marcombo.
- Gómez, F. M., Cervantes, O. J., & Gonzales, P. P. (2019). *Fundamentos de Ingeniería de Software*. México: Litoprocess S.A. de C. V.
- Google Maps. (2018). *Google Maps*. Obtenido de ¿Que es Google Maps?: [www.google.com/mumaps](http://www.google.com/mumaps)
- ISO 25000. (2022). *Normas ISO 25000*. Recuperado el 16 de Noviembre de 2021, de ISO IEC 25010: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- Lapiedra, A. R., Fore, z. J., & Martinez, C. L. (2021). *Introducción a la Gestion de Sistemas de Informacion Geográfica*. Publicaciones de la Universitat Jaume I.
- López, Q. J. (2014). *Domine PHP y MySQL*. 2da Edición. RA-MA.
- Luna, F. O. (2019). *JavaScript: Aprende a Programar en el Lenguaje de la Web*. Edición Claudio Peña.

Maldonado, R. I., Alva, Z. J., & Santa Cruz, A. R. (2021). *Pruebas de Software*. Bagua.

Morea, R. M., & Huerta, R. J. (2005). *Sistemas de Informacion Geográfica*. GUD.

Morea, R. M., & Huerta, R. J. (2005). *Sistemas de Información Geográfica*. Works.

Olaya, V. (2014). *Sistemas de Información Geográfica*. Creative Common Atribución.

Recio, G. J. (2016). *HTML5, CSS3 y JQuery, Curso Práctico*. RA-MA.

Rodríguez, L. J., & Olivella, R. (2006). *Introducción a los Sistemas de Información Geográfica*. FUOC.

Santamaria, R. (2013). *Introducción a los Servicios Web RESTful*. IA ALL.

Selman, H. (2017). *Marketing Digital*. Ibukku.

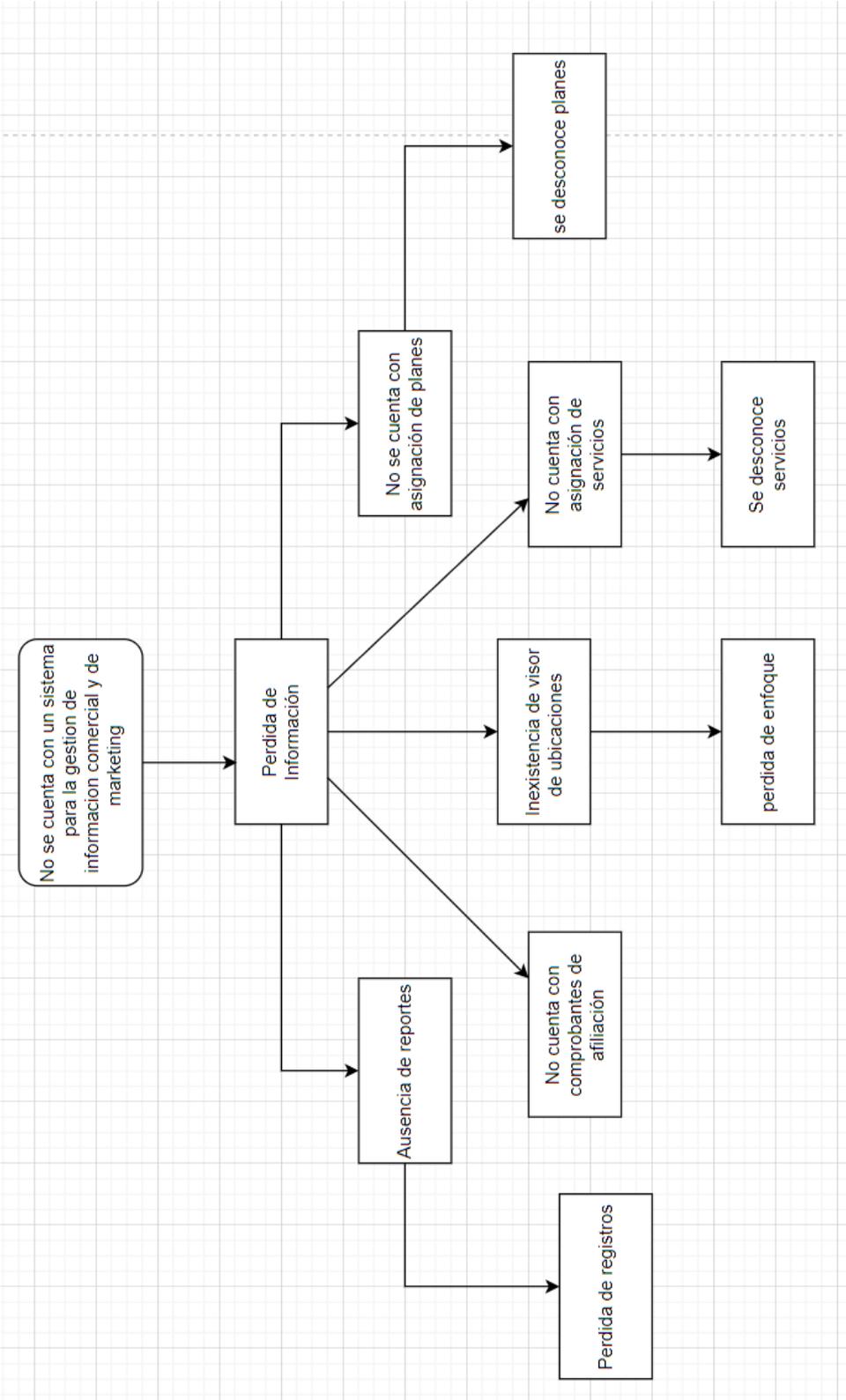
Shum, X. Y. (2019). *Marketing Digital*. 2da Ed. Ediciones de la U.

Sommerville, I. (2005). *Ingenieria de Software*. 7ma Edición. Pearson Educación.

Taylor, O. (2022). *Laravel*. Obtenido de The PHP Framework for Web Artisans: <https://laravel.com/>

**ANEXOS**

ANEXO A. ARBOL DE PROBLEMAS



**ANEXO B.**

**AVAL DE CONFORMIDAD**  
**Empresa FaceBol SRL**

El Alto, 10 de marzo de 2022

Señor:  
ING. ENRIQUE FLORES BALTAZAR.  
TUTOR METODOLÓGICO  
TALLER DE GRADO II  
Presente. –

**REF. AVAL DE CONFORMIDAD**

Distinguido tutor metodológico:  
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

**TITULO:** "SISTEMA WEB DE INFORMACIÓN GEOGRÁFICA ORIENTADA AL MARKETING PARA LAS EMPRESAS EN CONVENIO"  
**CASO:** FACEBOL SRL  
**MODALIDAD:** PROYECTO DE GRADO  
**UNIV.:** JANNETH VERONICA AMBROCIO MAMANI  
**REGISTRO UNIVERSITARIO:** 200002659  
**CEDULA DE IDENTIDAD:** 7322279 OR.

De tal forma cabe recalcar que el SISTEMA/PROYECTO satisface los requerimientos de la institución, de esta forma se dio cumplimiento de los objetivos del presente.

En cuanto certifico, en honor a la verdad, para fines consiguientes del interesado para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente,

  
Luis Fernando Ilaquita Fernández  
INGENIERO DE SISTEMAS  
R.N.I. 25.507

.....  
Ing. Luis Fernando Ilaquita Fernández  
GERENTE GENERAL  
FACEBOL SRL



## **ANEXO C.**

**ANEXO D.**

**AVAL DE CONFORMIDAD**  
(TUTOR ESPECIALISTA)

El Alto, 31 de mayo de 2022

Señor:

**Ing. Enrique Flores Baltazar**  
**TALLER DE GRADO II**  
Presente. –

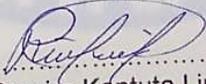
REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:  
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: "SISTEMA WEB DE INFORMACIÓN GEOGRÁFICA ORIENTADA AL MARKETING PARA LAS EMPRESAS EN CONVENIO"  
CASO: FACEBOL SRL  
MODALIDAD: PROYECTO DE GRADO  
Univ.: JANNETH VERONICA AMBROCIO MAMANI  
Registro Universitario: 200002659  
Cedula de Identidad: 7322279 OR.

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,

  
.....  
Ing. Ramiro Kantuta Limachi.  
TUTOR ESPECIALISTA

**ANEXO E.**

**AVAL DE CONFORMIDAD**

(TUTOR REVISOR)

El Alto, marzo de 2022

Señor:  
Ing. Enrique Flores Baltazar.  
**TUTOR METODOLÓGICO**  
**TALLER DE GRADO II**  
Presente. –

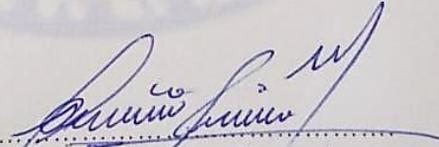
**REF. AVAL DE CONFORMIDAD**

Distinguido tutor metodológico:  
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: "SISTEMA WEB DE INFORMACIÓN GEOGRÁFICA ORIENTADA AL  
MARKETING PARA LAS EMPRESAS EN CONVENIO"  
CASO: FACEBOL SRL  
MODALIDAD: PROYECTO DE GRADO  
Univ.: JANNETH VERONICA AMBROCIO MAMANI  
Registro Universitario: 200002659  
Cedula de Identidad: 7322279 OR.

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,



Ing. Rolman Cordero Mamani.  
TUTOR REVISOR

# MANUAL DE USUARIO

## Administrador

### ❖ Vista para acceder al sistema e iniciar sesión

1. Ingresar nombre de usuario
2. Ingresar contraseña
3. Iniciar sesión

FaceBol Srl.  
¡Hazlo Diferente!

INGRESE SUS CREDENCIALES

Usuario

Ingresar Usuario

Contraseña

Ingresar Contraseña

Iniciar Sesión

1

2

## ❖ Vista de inicio de la pagina

Copyright © 2021. FaceboSri.

## ❖ Vista de usuarios

Usuarios | Usuarios

Buscar usuario

Nuevo Usuario

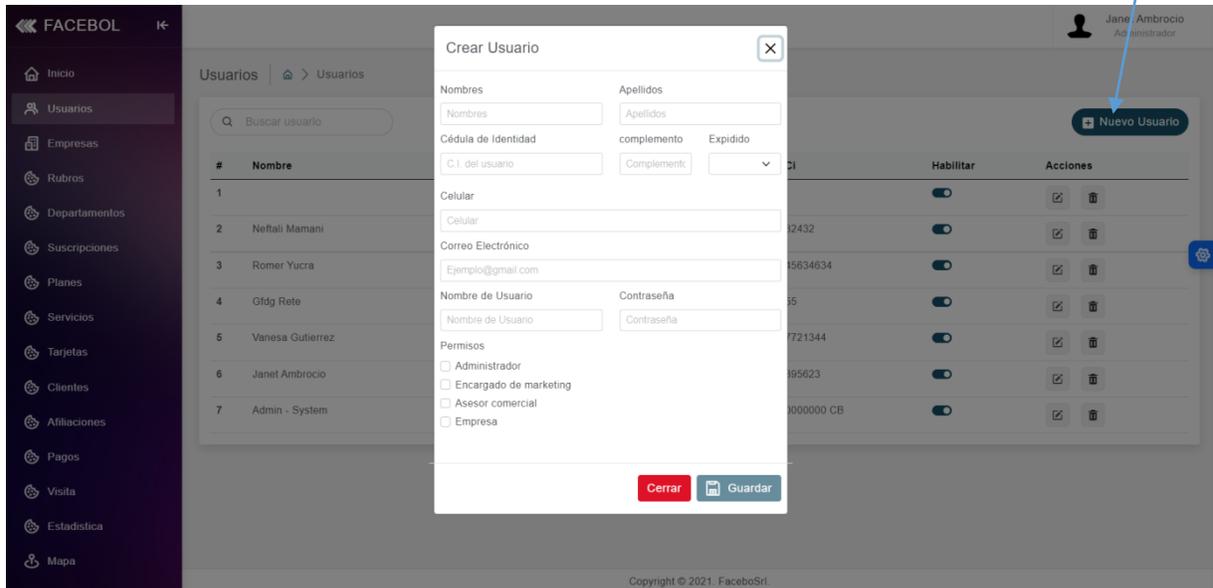
#	Nombre	Roles	Ci	Habilitar	Acciones
1		EMPRESA		<input checked="" type="checkbox"/>	
2	Neftali Mamani	ADMINISTRADOR	32432	<input checked="" type="checkbox"/>	
3	Romer Yucra	ADMINISTRADOR	45634634	<input checked="" type="checkbox"/>	
4	Gldg Rete	ENCARGADO DE MARKETING	55	<input checked="" type="checkbox"/>	
5	Vanesa Gutierrez	ASESOR COMERCIAL	7721344	<input checked="" type="checkbox"/>	
6	Janet Ambrocio	ADMINISTRADOR	895623	<input checked="" type="checkbox"/>	
7	Admin - System	ADMINISTRADOR	0000000 CB	<input checked="" type="checkbox"/>	

localhost:49965/dashboard/usuario/list-usuario

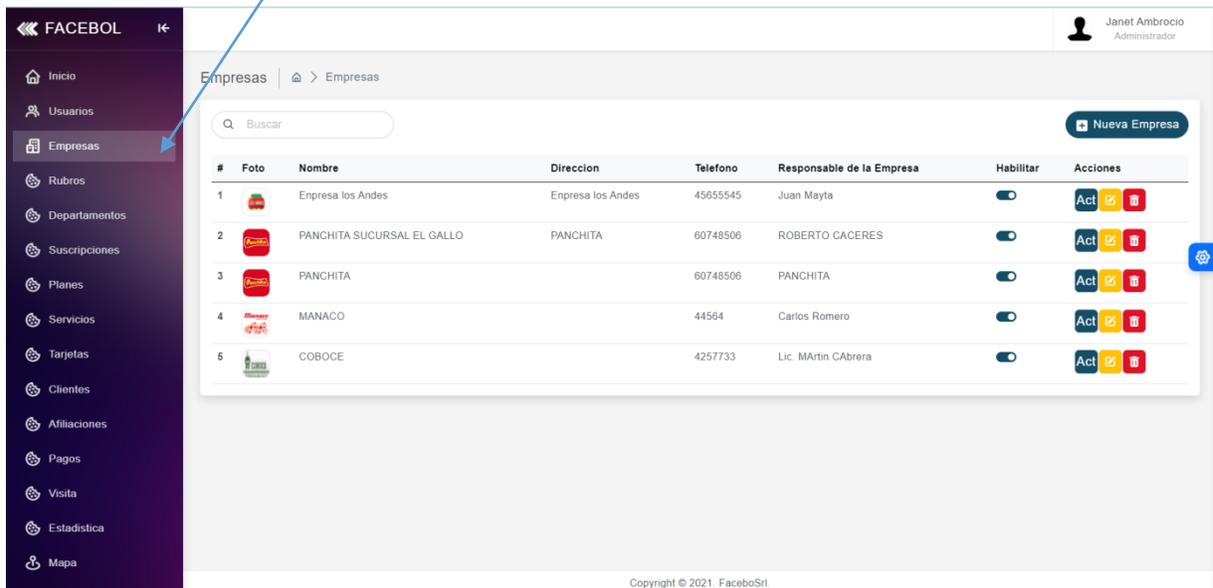
Copyright © 2021. FaceboSri.

❖ Formulario para crear un nuevo usuario.

Clic en el botón



❖ Vista de empresas registradas



## ❖ Formulario para crear empresa

Clic en el botón

Crear Empresa

Datos de la Empresa | Ubicación | Foto | Servicios

Nombre de Usuario:

Contraseña:

Correo Electrónico:

Nombre:

Descripción:

Página Web:

Dirección:

Teléfono:

Responsable:

Rubro:

Cerrar Guardar

## ❖ Formulario para ubicar empresa en el mapa

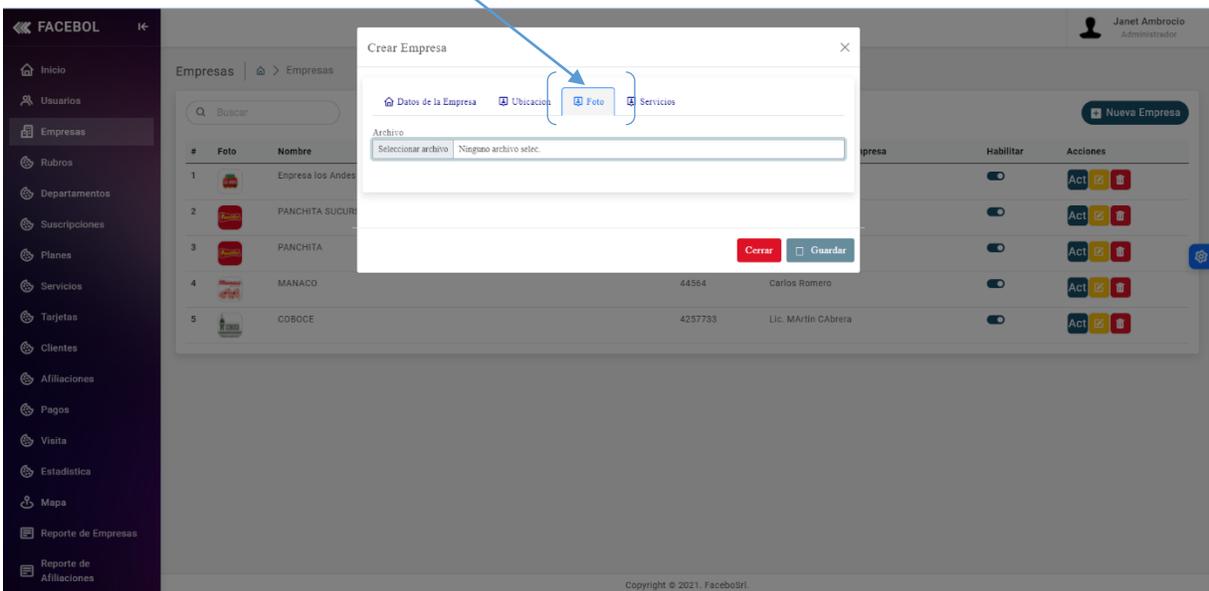
Crear Empresa

Datos de la Empresa | Ubicación | Foto | Servicios

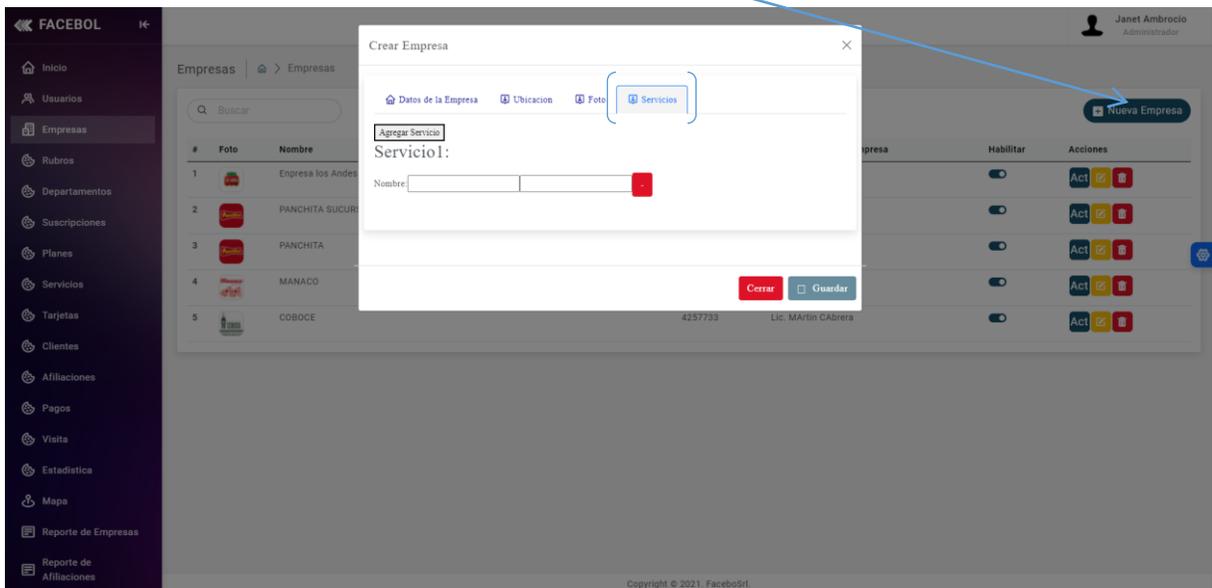
Mapa de Google Maps con una ubicación marcada.

Cerrar Guardar

## ❖ Formulario para subir logo de la empresa



## ❖ Formulario para registrar servicios



## ❖ Formulario para agregar promoción de las empresas

Clic en el botón

FACEBOL

Inicio  
Usuarios  
Empresas  
Rubros  
Departamentos  
Suscripciones  
Planes  
Servicios  
Tarjetas  
Clientes  
Afilaciones  
Pagos  
Visita  
Estadística  
Mapa  
Reporte de Empresas  
Reporte de Afilaciones

Empresa > Promociones

Nombre Empresa: Empresa los Andes  
Descripción Empresa: Empresa los Andes

Nombre  
Descripción  
Fecha Inicio  
Fecha de Vencimiento

Cerrar Guardar

Agregar Promoción

#	Nombre	Descripción	Fecha de Inicio	Fecha de Final	Habilitar	Acciones
---	--------	-------------	-----------------	----------------	-----------	----------

Copyright © 2021. FaceboSrl.

## ❖ Vista de los rubros registrados

FACEBOL

Inicio  
Usuarios  
Empresas  
Rubros  
Departamentos  
Suscripciones  
Planes  
Servicios  
Tarjetas  
Clientes  
Afilaciones  
Pagos  
Visita  
Estadística  
Mapa  
Reporte de Empresas  
Reporte de Afilaciones

Rubros > Rubros

Buscar

Nuevo Rubro

#	Nombre	Fecha de Registro	Habilitar	Acciones
1	TRANSPORTE	2022-01-19T00:41:14.000000Z	On	✍️ 🗑️
2	SERVICIO TECNICO	2022-01-19T00:40:33.000000Z	On	✍️ 🗑️
3	SERVICIOS PROFESIONALES Y TECNICOS	2022-01-19T00:39:49.000000Z	On	✍️ 🗑️
4	SERVICIO DE BELLEZA E IMAGEN PERSONAL	2022-01-19T00:39:03.000000Z	On	✍️ 🗑️
5	SALUD Y CENTROS MEDICOS	2022-01-19T00:30:36.000000Z	On	✍️ 🗑️
6	SALUD Y BELLEZA	2022-01-19T00:29:54.000000Z	On	✍️ 🗑️
7	ROPA Y ACCESORIOS	2022-01-19T00:29:29.000000Z	On	✍️ 🗑️
8	RESTAURANTES Y SNAKCS	2022-01-19T00:26:19.000000Z	On	✍️ 🗑️
9	PRODUCTOS	2022-01-19T00:24:49.000000Z	On	✍️ 🗑️
10	MASCOTAS Y SERVICIOS VETERINARIOS	2022-01-19T00:24:14.000000Z	On	✍️ 🗑️
11	INSTITUCIONES	2022-01-19T00:23:47.000000Z	On	✍️ 🗑️
12	INMOBILIARIA	2022-01-19T00:23:21.000000Z	On	✍️ 🗑️
13	HOSPEDAJE Y TURISMO	2022-01-19T00:14:23.000000Z	On	✍️ 🗑️

Copyright © 2021. FaceboSrl.

## ❖ Formulario para crear un nuevo rubro

Clic en el botón

FACEBOL

Rubros

Crear Rubro

Nombre

nombre

Cerrar Guardar

#	Nombre	Fecha de Registro	Habilitar	Acciones
1	TRANSPORTE	2022-01-19T00:40:33.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
2	SERVICIO TECNICO	2022-01-19T00:39:49.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
3	SERVICIOS PROFESIONALES Y TECNICOS	2022-01-19T00:39:03.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
4	SERVICIO DE BELLEZA E IMAGEN PERSONAL	2022-01-19T00:30:36.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
5	SALUD Y CENTROS MEDICOS	2022-01-19T00:29:54.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
6	SALUD Y BELLEZA	2022-01-19T00:29:29.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
7	ROPA Y ACCESORIOS	2022-01-19T00:29:29.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
8	RESTAURANTES Y SNACKS	2022-01-19T00:26:19.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
9	PRODUCTOS	2022-01-19T00:24:49.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
10	MASCOTAS Y SERVICIOS VETERINARIOS	2022-01-19T00:24:14.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
11	INSTITUCIONES	2022-01-19T00:23:47.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
12	INMOBILIARIA	2022-01-19T00:23:21.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
13	HOSPEDAJE Y TURISMO	2022-01-19T00:14:23.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>

Janet Ambrocio  
Administrador

Nuevo Rubro

Copyright © 2021. FaceboSrl.

## ❖ Vista de departamentos registrados

FACEBOL

Departamento

Departamentos

Buscar

Nuevo Departamento

#	Nombre	Fecha de Registro	Habilitar	Acciones
1	LA PAZ	2022-01-13T14:19:23.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
2	COCHABAMBA	2022-01-13T14:19:23.000000Z	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>

Janet Ambrocio  
Administrador

Nuevo Departamento

Copyright © 2021. FaceboSrl.

## ❖ Formulario para registrar un nuevo departamento

The screenshot shows the 'FACEBOL' dashboard with the 'Departamentos' menu item selected. A modal window titled 'Crear Departamento' is open, featuring a text input field labeled 'Nombre' with the placeholder text 'nombre'. Below the input field are two buttons: 'Cerrar' (red) and 'Guardar' (blue). In the background, a table lists existing departments:

#	Nombre	Habilitar	Acciones
1	LA PAZ	<input checked="" type="checkbox"/>	
2	COCHABAMBA	<input checked="" type="checkbox"/>	

The user 'Janet Ambrocio Administrador' is logged in, and the copyright notice 'Copyright © 2021. FaceboSrl.' is visible at the bottom.

## ❖ Vista de suscripciones de las empresas

The screenshot shows the 'FACEBOL' dashboard with the 'Suscripciones' menu item selected. The main content area displays a table of subscriptions:

#	EMPRESA	PLAN	FECHA INICIO	FECHA VENCIMIENTO	Habilitar	Acciones
1	PANCHITA SUCURSAL EL GALLO	ESTANDAR	17/02/2022	18/02/2022	<input checked="" type="checkbox"/>	

The user 'Janet Ambrocio Administrador' is logged in, and the copyright notice 'Copyright © 2021. FaceboSrl.' is visible at the bottom.

## ❖ Formulario para crear una nueva suscripción

Clic en el botón

FACEBOL

Inicio  
Usuarios  
Empresas  
Rubros  
Departamentos  
Suscripciones  
Planes  
Servicios  
Tarjetas  
Clientes  
Afilaciones  
Pagos  
Visita  
Estadística  
Mapa  
Reporte de Empresas  
Reporte de Afilaciones

suscripciones

Buscar

# EMPRESA

1 PANCHITA SUCURSAL EL GALL

Crear Suscripción

Empresa

Plan

Fecha Inicio dd/mm/aaaa

Fecha de Vencimiento dd/mm/aaaa

Observación

Cerrar Guardar

Nueva Suscripción

Habilitar Acciones

Janet Ambrocio Administrador

Copyright © 2021. FaceboSrl.

## ❖ Vista de los planes registrados

FACEBOL

Inicio  
Usuarios  
Empresas  
Rubros  
Departamentos  
Suscripciones  
Planes  
Servicios  
Tarjetas  
Clientes  
Afilaciones  
Pagos  
Visita  
Estadística  
Mapa  
Reporte de Empresas  
Reporte de Afilaciones

Planes

Buscar

Nuevo Plan

#	Nombre	Costo	Duracion	Fecha de Registro	Habilitar	Acciones
1	2334	244	1	29/05/2022	<input checked="" type="checkbox"/>	Act
2	GRATUITO	0	6 meses	13/01/2022	<input checked="" type="checkbox"/>	Act
3	ESTANDAR	300	1 año	13/01/2022	<input checked="" type="checkbox"/>	Act
4	PREMIUM	500	1 año	13/01/2022	<input checked="" type="checkbox"/>	Act

Janet Ambrocio Administrador

Copyright © 2021. FaceboSrl.

## ❖ Formulario para crear nuevo plan

Clic en el botón

The screenshot shows the 'Planes' management interface. A modal form titled 'Crear Plan' is open, allowing the user to enter details for a new plan. The form includes fields for 'Nombre', 'Costo', and 'Duracion'. Below the form are 'Cerrar' and 'Guardar' buttons. In the background, a table lists existing plans, and a 'Nuevo Plan' button is visible in the top right corner of the table area, which is highlighted by a callout box.

#	Nombre	Costo	Habilitar	Acciones
1	2334	244	<input checked="" type="checkbox"/>	Act
2	GRATUITO	0	<input checked="" type="checkbox"/>	Act
3	ESTANDAR	300	<input checked="" type="checkbox"/>	Act
4	PREMIUM	500	<input checked="" type="checkbox"/>	Act

## ❖ Vista de los servicios que brinda la empresa

The screenshot shows the 'Servicios' management interface. A blue arrow points from the 'Servicios' menu item in the left sidebar to the main content area. The main area displays a table of services with columns for '#', 'Nombre', 'Fecha de Registro', 'Habilitar', and 'Acciones'. A 'Nuevo Servicio' button is located in the top right corner of the table area.

#	Nombre	Fecha de Registro	Habilitar	Acciones
1	Cursos y seminarios gratuitos	13/01/2022	<input checked="" type="checkbox"/>	Act
2	Selección de publicidad	13/01/2022	<input checked="" type="checkbox"/>	Act
3	publicidad en pagina web	13/01/2022	<input checked="" type="checkbox"/>	Act
4	marketing digital	13/01/2022	<input checked="" type="checkbox"/>	Act

## ❖ Formulario para crear nuevo servicio

Clic en el botón

The screenshot shows the 'FACEBOL' interface with a sidebar menu on the left. The main content area is titled 'Servicios' and contains a table with the following data:

#	Nombre	Habilitar	Acciones
1	Cursos y seminarios gratuitos	<input checked="" type="checkbox"/>	Act. [icon] [icon]
2	Selección de publicidad	<input checked="" type="checkbox"/>	Act. [icon] [icon]
3	publicidad en pagina web	<input checked="" type="checkbox"/>	Act. [icon] [icon]
4	marketing digital	<input checked="" type="checkbox"/>	Act. [icon] [icon]

A modal form titled 'Crear Servicio' is open in the center, with fields for 'Nombre' (containing 'nombre') and 'Descripcion' (containing 'descripcion'). The form has 'Cerrar' and 'Guardar' buttons. A blue callout box with the text 'Clic en el botón' points to a 'Nuevo Servicio' button in the top right corner of the main interface.

## ❖ Vista de las tarjetas registradas

The screenshot shows the 'FACEBOL' interface with the sidebar menu on the left. The main content area is titled 'Tarjetas' and contains a table with the following data:

#	Nombre	Costo	Descripcion	Fecha de Registro	Habilitar	Acciones
1	tarjeta Blanca	100	tarjeta por el dia de la madre	01/06/2022	<input checked="" type="checkbox"/>	[icon] [icon]
2	Tarjeta Azul	200	tarjeta azul	23/02/2022	<input checked="" type="checkbox"/>	[icon] [icon]

A blue arrow points from the 'Tarjetas' menu item in the sidebar to the 'Nuevo Tarjeta' button in the top right corner of the main interface.

## ❖ vista de clientes registrados

FACEBOL

Inicio  
Usuarios  
Empresas  
Rubros  
Departamentos  
Suscripciones  
Planes  
Servicios  
Tarjetas  
**Clientes**  
Afiliaiones  
Pagos  
Visita  
Estadística  
Mapa  
Reporte de Empresas  
Reporte de Afiliaiones

Janet Ambrocio  
Administrador

Cientes > Cientes

Buscar

Nuevo Cliente

#	Foto	Cliente	CI	Direccion	Correo	Telefono	Celular	Acciones
1		Veronica Mamani	5656767	trf	fr@gmail.com	54654	5654	
2		Janneth Ambrocio Mamani	33366775	El Alto		4456577	567878898o	
3		Victor Hugo Perez Rojas	4845944	Calle 1 Zona Licenciada	victorvirtual@gmail.com	+59160748506	60748506	

Copyright © 2021. FaceboSRI.

## ❖ Formulario para crear nuevo cliente

Clic en el botón

FACEBOL

Inicio  
Usuarios  
Empresas  
Rubros  
Departamentos  
Suscripciones  
Planes  
Servicios  
Tarjetas  
**Clientes**  
Afiliaiones  
Pagos  
Visita  
Estadística  
Mapa  
Reporte de Empresas  
Reporte de Afiliaiones

Janet Ambrocio  
Administrador

Cientes > Cientes

Buscar

Nuevo Cliente

Crear Clientes

Nombres: nombre  Apellidos: apellidos

C.I.: C.I. Cliente  complemento  Expidido

Telefono: Telefono  Celular: Telefono

Direccion: Direccion

Correo: Correo Electrónico

Cerrar Guardar

#	Foto	Cliente	CI	Direccion	Correo	Telefono	Celular	Acciones
1		Veronica Mamani				54654	5654	
2		Janneth Ambrocio Mamani				4456577	567878898o	
3		Victor Hugo Perez Rojas			@gmail.com	+59160748506	60748506	

Copyright © 2021. FaceboSRI.

## ❖ Vista de registros de afiliación

FACEBOL

Inicio  
Usuarios  
Empresas  
Rubros  
Departamentos  
Suscripciones  
Planes  
Servicios  
Tarjetas  
Clientes  
**Afilaciones**  
Pagos  
Visita  
Estadística  
Mapa  
Reporte de Empresas  
Reporte de Afilaciones

Janet Ambrocio  
Administrador

Afilaciones > Afilaciones

Buscar

Nueva Afiliación

#	CLIENTE	TARJETA	FECHA INICIO	FECHA VENCIMIENTO	Habilitar	Acciones
1	Veronica Mamani	tarjeta Blanca	022/06/2022	3030/06/2022	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
2	Veronica Mamani	Tarjeta Azul	044/06/2022	3030/06/2022	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
3	Janneth Ambrocio Mamani	Tarjeta Azul	011/06/2022	1616/06/2022	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
4	Victor Hugo Perez Rojas	Tarjeta Azul	033/03/2022	011/04/2022	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
5	Victor Hugo Perez Rojas	Tarjeta Azul	033/02/2022	1818/02/2022	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
6	Victor Hugo Perez Rojas	Tarjeta Azul	033/02/2022	2626/02/2022	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>

Copyright © 2021. FacebolSrl.

## ❖ Formulario para crear nueva afiliación

FACEBOL

Inicio  
Usuarios  
Empresas  
Rubros  
Departamentos  
Suscripciones  
Planes  
Servicios  
Tarjetas  
Clientes  
**Afilaciones**  
Pagos  
Visita  
Estadística  
Mapa  
Reporte de Empresas  
Reporte de Afilaciones

Janet Ambrocio  
Administrador

Afilaciones > Afilaciones

Buscar

Nueva Afiliación

Crear Afiliación

Cliente

Tarjeta

Fecha Inicio

Fecha de Vencimiento

Observacion

Cerrar Guardar

#	CLIENTE	TARJETA	FECHA INICIO	FECHA VENCIMIENTO	Habilitar	Acciones
1	Veronica Mamani				<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
2	Veronica Mamani				<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
3	Janneth Ambrocio Mamani				<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
4	Victor Hugo Perez Rojas				<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
5	Victor Hugo Perez Rojas				<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
6	Victor Hugo Perez Rojas	Tarjeta Azul	033/02/2022	2626/02/2022	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>

Copyright © 2021. FacebolSrl.

## ❖ Vista de tarjeta generada

The screenshot displays the FACEBOL system interface. On the left, a dark sidebar menu lists various options, with 'Afiliaciones' highlighted. The main content area shows a list of clients under the heading 'Afiliaciones'. A modal window titled 'Archivo del sistema' is open, displaying a generated membership card for 'Veronica Mamani'. The card includes the FACEBOL logo, the text 'Afiliación', and a QR code. The card details are: Nombre: Veronica, Apellido: Mamani, Celular: 5654. To the right of the modal, there is a 'Nueva Afiliación' button and a table of actions for each client, with a blue arrow pointing to the first row.

#	N Comprobante	detalle	Costo	Fecha de Registro	Acciones
1	15	Pago por afiliacion del 02-06-2022 al 30-06-2022	100	01/06/2022	[icon]
2	14	Suscripcion del plan GRATUITO del 06-06-2022 al 30-06-2022	0	01/06/2022	[icon]
3	13	Pago por afiliacion del 04-06-2022 al 30-06-2022	200	01/06/2022	[icon]
4	12	Pago por afiliacion del 01-06-2022 al 16-06-2022	200	01/06/2022	[icon]
5	11	Suscripcion del plan GRATUITO del 28-01-2022 al 29-06-2022	0	31/05/2022	[icon]
6	10	Pago por afiliacion del 03-03-2022 al 01-04-2022	200	02/03/2022	[icon]

## ❖ Vista de pagos registrados en el sistema

The screenshot displays the FACEBOL system interface. On the left, a dark sidebar menu lists various options, with 'Pagos' highlighted. The main content area shows a table of registered payments under the heading 'Cobranzas > Pagos'. The table has columns for '#', 'N Comprobante', 'detalle', 'Costo', 'Fecha de Registro', and 'Acciones'. A blue arrow points from the 'Pagos' menu item to the table.

#	N Comprobante	detalle	Costo	Fecha de Registro	Acciones
1	15	Pago por afiliacion del 02-06-2022 al 30-06-2022	100	01/06/2022	[icon]
2	14	Suscripcion del plan GRATUITO del 06-06-2022 al 30-06-2022	0	01/06/2022	[icon]
3	13	Pago por afiliacion del 04-06-2022 al 30-06-2022	200	01/06/2022	[icon]
4	12	Pago por afiliacion del 01-06-2022 al 16-06-2022	200	01/06/2022	[icon]
5	11	Suscripcion del plan GRATUITO del 28-01-2022 al 29-06-2022	0	31/05/2022	[icon]
6	10	Pago por afiliacion del 03-03-2022 al 01-04-2022	200	02/03/2022	[icon]

## ❖ Generar comprobante de afiliación o convenio

Archivo del sistema - Google Chrome

Archivo del sistema

COMPROBANTE DE AFILIACION Nro: 15

Razon social: Veronica Mamani Fecha: 01-06-2022

N.	DETALLE	TOTAL
1	Pago por afiliacion del 02-06-2022 al 30-06-2022	100.00
TOTAL		100.00

Cajero: Janet Ambrocio

Entregue conforme Recibi conforme

Fecha de Registro	Acciones
1/06/2022	[icon]
1/05/2022	[icon]
12/03/2022	[icon]

Copyright © 2021. FaceboSrl.

## ❖ Vista para escanear código QR

FACEBOL

Acerque un código QR para escanear.

#	Cliente	Fecha Ingreso
1	Victor Hugo Perez Rojas	2022-06-02 09:47:38
2	Victor Hugo Perez Rojas	2022-06-02 09:47:38
3	Victor Hugo Perez Rojas	2022-06-02 09:47:38
4	Victor Hugo Perez Rojas	2022-06-02 09:47:38
5	Veronica Mamani	2022-06-02 13:10:25
6	Veronica Mamani	2022-06-02 13:10:26
7	Veronica Mamani	2022-06-02 13:10:27
8	Veronica Mamani	2022-06-02 13:10:28

Copyright © 2021. FaceboSrl.

## ❖ Vista de las visitas realizadas a las empresas

The screenshot shows the FACEBOL dashboard with the 'Estadístico' view selected. The left sidebar contains a menu with 'Estadística' highlighted. The main content area displays a table with the following data:

#	Nombre Empresa	Cantidad
1	COBOCE	8
2	MANACO	14
3	PANCHITA	36
4	PANCHITA SUCURSAL EL GALLO	0
5	Empresa los Andes	3

The dashboard also includes a search bar, a user profile for 'Janet Ambrocio', and a footer with 'Copyright © 2021. FaceboSrl'.

## ❖ Vista en el mapa geográfico de las empresas registradas

The screenshot shows the FACEBOL dashboard with the 'Mapa' view selected. The left sidebar contains a menu with 'Mapa' highlighted. The main content area displays a map of La Paz, Bolivia, with several red location pins. A pop-up window for 'LOS ANDES' is visible, showing the company name and coordinates: 'Empresa: Empresa los Andes', 'Latitude: -16.525561', and 'Longitude: -68.186217'. The dashboard also includes a search bar, a user profile for 'Janet Ambrocio', and a footer with 'Copyright © 2021. FaceboSrl'.

## ❖ Vista de reporte de las empresas activas

The screenshot displays the FACEBOL web application interface. On the left, a dark sidebar menu contains various navigation options, with 'Reporte de Empresas' highlighted and an arrow pointing to it. The main content area shows a 'Vista previa del Reporte' window. The report is titled 'REPORTE DE EMPRESAS' and is dated '02-06-2022'. It features a table with the following data:

#	Nombre empresa	Direccion	Telefono	Responsable de la empresa	Fecha de suscripción	Fecha de vencimiento
1	PANCHITA SUCURSAL EL GALLO	PANCHITA	60748506	ROBERTO CACERES	2022-02-17	2022-02-18

The background interface shows the 'Empresas' section with a search bar and a list of companies, including 'PANCHITA SUCURSAL EL GALLO'. The top right corner displays the user 'Janet Ambrocio, Administrador' and a date range for subscriptions from 17/02/2022 to 18/02/2022. A 'Copyright © 2021, FaceBoSrl' notice is visible at the bottom.

## ❖ vista de reporte de afiliación

The screenshot displays the FACEBOL web application interface. On the left, a dark sidebar menu contains various navigation options, with 'Reporte de Afiliaciones' highlighted and an arrow pointing to it. The main content area shows a 'Vista previa del Reporte' window. The report is titled 'REPORTE DE AFILIACIÓN' and is dated '02-06-2022'. It features a table with the following data:

#	Cliente	Tarjeta	Fecha Inicio	Fecha Vencimiento	Costo
1	Veronica Mamani	tarjeta Blanca	2022-06-02	2022-06-30	100

The background interface shows the 'Reporte' section with a search bar and a list of clients, including 'Veronica Mamani'. The top right corner displays the user 'Janet Ambrocio, Administrador' and a date range for subscriptions from 02/06/2022 to 02/06/2022. A 'Copyright © 2021, FaceBoSrl' notice is visible at the bottom.

❖ Vista de reporte de todas las empresas suscritas

The screenshot displays the FACEBOL application interface. On the left is a dark sidebar menu with various options. The main content area shows a 'Reporte' section for 'Suscripciones'. A modal window titled 'Vista previa del Reporte' is open, showing a PDF preview of a report. The report includes the FACEBOL logo, the title 'REPORTE DE SUSCRIPCIONES', the date '02-06-2022', and a table with the following data:

#	Empresa	Plan	Fecha de inicio	Fecha de vencimiento	Monto
1	PANCHITA SUCURSAL EL GALLO	ESTANDAR	2022-02-17	2022-02-18	300

The background interface shows a list of subscriptions with columns for 'EMPRESA', 'FECHA VENCIMIENTO', and 'Monto'. The user 'Janet Ambrocio' is logged in as an administrator.

❖ Vista de estadísticas

The screenshot displays the 'Estadístico' view in the FACEBOL application. The sidebar menu on the left has 'Estadística' highlighted. The main content area shows a date range filter set to '08/06/2022' for both 'Fecha Inicio' and 'Fecha Fin'. Below the filter is a table with the following data:

#	Nombre Empresa	Cantidad
1	PIL	0
2	TOYOTA	0

To the right of the table is a bar chart with a y-axis from 0 to 1.0 and x-axis labels for 'PIL' and 'TOYOTA'. The chart shows zero bars for both categories. The user 'Janet Ambrocio' is logged in as an administrator.