

UNIVERSIDAD AUTÓNOMA GABRIEL RENE MORENO  
Facultad de Ingeniería en Ciencias de la Computación  
y  
Telecomunicaciones



# Universidad Autónoma Gabriel René Moreno

## GRUPO 11

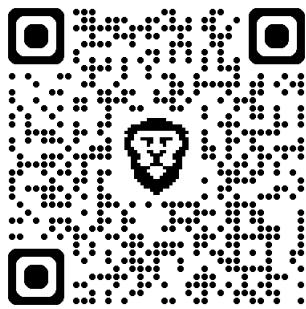
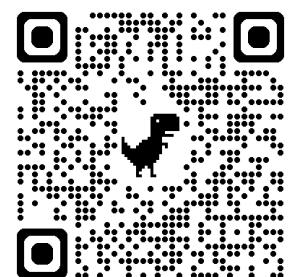
### “SOFTWARE DE GESTIÓN Y SEGUIMIENTO DE MERCADERÍAS PUBLICITARIOS”

**Docente:** Ing. Rolando Martínez

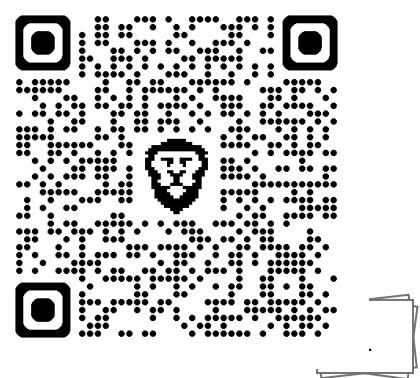
**Materia:** Ingeniería de Software 2

#### Integrantes:

- |                                |           |
|--------------------------------|-----------|
| 1. Machuca Yavita Miguel Angel | 216063558 |
| 2. Torrez Garcia Jorge Luis    | 218137761 |



Gestion II-2024 Santa Cruz - Bolivia



## 1. Contenido

1. PAPS.....	14
1.1. Ámbito del proyecto.....	15
1.1.1. Introducción.....	15
1.1.2. Antecedentes.....	15
Objetivos.....	17
Objetivo general .....	17
Objetivos específicos .....	17
1.2. Métricas.....	18
1.2.1. Google encontrar mi dispositivo.....	18
Métricas orientadas a tamaño .....	20
Métricas orientadas a la función.....	20
1.2.2. GURTAM.....	22
Métricas orientadas a tamaño .....	23
Métricas orientadas a la función.....	23
Driv.in.....	25
métricas orientadas a tamaño.....	25
Métricas orientadas a la función.....	26
Tabla de Métricas de los tres proyectos medidos .....	27
1.3. Estimaciones del proyecto.....	28

Tamaño del proyecto .....	28
1.4.    Objetivos .....	30
Objetivo general .....	30
Objetivos específicos .....	31
Requerimientos funcionales .....	31
No Funcionales: .....	32
1.5.    Restricciones .....	33
1.5.1. Restricciones Tecnológicas .....	33
1.5.2. Restricciones Presupuestarias.....	33
1.5.3. Restricciones de Tiempo .....	33
1.5.4. Restricciones de Infraestructura .....	34
1.6.    Interfaces Externas .....	35
1.6.1. Interacciones con Software.....	35
1.6.2. Interacciones con Personas.....	35
1.6.3. Interacciones con Hardware .....	35
1.7.    Estimación del proyecto .....	35
1.7.1. Valor esperado.....	35
1.7.2. COCOMO II.....	36
1.7.3. Historias de usuario (Planning Poker) .....	37
1.7.4. 1. El cliente navega hacia el sitio web donde se podrá registrar. ....	37

1.8.	Gestión de riesgos .....	40
1.9.	Planificación del tiempo.....	42
1.9.1.	Diagrama de Gantt.....	43
1.9.2.	Diagrama de PERT .....	44
1.10.	Tabla de recursos.....	45
1.11.	Organización interna .....	46
1.11.1.	Descentralizada Democrática .....	46
1.11.2.	Descentralizada Controlada (DC): .....	47
1.11.3.	Centralizada Controlada (CC): .....	48
1.12.	Mecanismos de seguimiento y control .....	48
	Reunión Técnica Formal (RTF) y Diseño de Formulario (DF) .....	48
•	Dayli SCRUM .....	49
•	Sprint Review .....	50
1.13.	Tabla de recursos.....	52
<b>2.</b>	<b>Modelo de Desarrollo de Software .....</b>	<b>54</b>
2.1.	PROCESO DE DESARROLLO DE SOFTWARE “SCRUM”.....	55
2.1.1.	Requisitos Iniciales.....	55
2.1.2.	Equipos de Desarrollo .....	56
2.1.3.	Definición del Patrón de Desarrollo .....	56
2.1.4.	Duración del Sprint.....	56

2.1.5. Herramientas de Gestión de Proyecto .....	56
2.1.6. Modelo de contexto inicial .....	57
2.1.7. Modelo de Arquitectura Inicial.....	57
2.1.8. Base de datos inicial .....	58
2.1.9. Producto Backlog Inicial .....	58
2.2. Sprint 3 .....	60
2.2.1. Objetivos del Sprint .....	60
2.2.2. Patrón de desarrollo por Historia de Usuario .....	60
Historias de Usuario .....	60
Contexto del Sistema.....	62
Sprint backlog.....	62
Diseño .....	64
Sprint Retrospective .....	66
<b>3. Manual de Calidad (SQA).....</b>	<b>67</b>
3.1. Introducción .....	68
3.2. Antecedente.....	68
3.3. Objetivos .....	69
3.3.1. objetivo general .....	69
3.3.2. objetivos específicos.....	69
3.3.3. Misión.....	70

3.3.4. Visión .....	70
3.4. Política de calidad .....	71
3.5. SLOGAN.....	71
3.6. Plan de Aseguramiento de Calidad de Software (SQAP) .....	72
3.7. Estándares .....	73
3.8. Gestión .....	74
3.8.1. Organización.....	74
3.8.2. Tareas.....	75
3.8.3. Actividades .....	76
3.8.4. Responsables .....	80
3.9. Evolución del SQAP .....	83
3.9.1. Control de cambio .....	83
3.9.2. Cuando hacer modificaciones.....	84
3.9.3. Implantación de modificaciones .....	84
3.9.4. Publicación del SQAP .....	85
3.10. Documentación .....	85
3.11. Plan de Verificación y Validación .....	88
3.11.1.Ciclo de Vida de Verificación y Validación.....	89
3.11.2.Informe de Verificación y Validación .....	91
Reporte sumario de fase V&V.....	92

Reporte de Anomalías .....	93
Reporte Final de V&V.....	94
3.12. Documentación del Usuario (UD) .....	95
3.12.1.Modo Instruccional.....	95
3.12.2.Modo de Referencia.....	95
3.13. Estándares, prácticas y convenciones .....	97
3.13.1.Estandar de Codificación.....	97
3.13.2.Estandar de Comentarios .....	99
Estándares para documentación .....	99
3.13.3.Responsables de verificar el cumplimiento .....	99
3.14. Revisiones y auditorias .....	99
3.14.1.Revisión de los Requisitos de Software (SRR): .....	100
3.14.2.Revisión del Diseño Preliminar (PDR): .....	100
3.14.3.Revisión del Diseño Crítico (CDR):.....	101
3.14.4.Revisión del Plan de Verificación y Validación (SVVPR): .....	101
3.14.5.Auditoria Funcional:.....	102
3.14.6.Auditoria Física (PA): .....	103
3.14.7.Auditorías del Proceso (IPA): .....	103
3.15. Revisiones de gestión:.....	104
3.15.1.Revisar el ajuste al proceso: .....	105

3.15.2.Revisión Técnica Formal (RTF):.....	106
3.15.3.Gestión de Configuración.....	109
3.15.4.Gestión de Problemas y Acciones Correlativas.....	109
<b>3.16. Herramientas, técnicas y metodologías .....</b>	<b>111</b>
3.16.1.Control del Código .....	112
3.16.2.Control de Medios .....	112
3.16.3.Medio de Almacenamiento.....	113
3.16.4.Proceso de copias de seguridad .....	113
3.16.5.Puntos de Control .....	113
3.16.6.Control de Suministradores y Subcontratos .....	113
3.16.7.Recolección, Mantenimiento y Retención de Registros.....	114
<b>4. Productividad basada en Herramientas CASE.....</b>	<b>115</b>
4.1. ChatGpt .....	116
4.1.1. La Integración de ChatGPT en el Desarrollo de Software .....	116
4.2. Cursor .....	118
4.3. Jira .....	120
<b>5. Aspectos Legales.....</b>	<b>122</b>
5.1. Apertura de una empresa en Santa Cruz.....	123
5.1.1. FUNDAEMPRESA .....	123
5.1.2. NIT 124	

5.1.3. Gobierno Municipal.....	125
5.1.4. Caja Nacional de Salud.....	125
5.1.5. AFPs .....	125
5.1.6. Ministerio de Trabajo .....	127
5.2. Derecho de Autor .....	127
5.2.1. SENAPI .....	127
<b>6. Infraestructura para la Producción del Software.....</b>	<b>130</b>
6.1. Gestión de Configuración de Software .....	131
6.1.1. Introducción.....	131
6.1.2. Alcance .....	131
6.1.3. Objetivos.....	131
6.1.4. Líneas Base .....	132
6.1.5. Tareas de la configuración del software .....	132
6.2. Herramientas para el Control de Versiones .....	133
6.2.1. Herramienta de control de versiones .....	134
6.3. Herramientas utilizadas en el proyecto .....	135
6.3.1. GitHub .....	135
6.3.2. Visual Studio Code.....	136
<b>7. Desarrollo del Sitio Web.....</b>	<b>137</b>
7.1. Sitio web.....	138

7.1.1. Información Principal .....	138
7.1.2. Logo.....	138
7.1.3. Planificación .....	138
7.1.4. Diseño y Prototipo.....	139
7.1.4.1. Inicio .....	139
7.1.4.2. Servicios.....	139
7.1.4.3. Nosotros .....	140
7.1.4.4. Proyectos:.....	140
7.1.4.5. Contactos.....	141
7.1.5. Arquitectura de la Página Web.....	142
7.1.6. Chatbot .....	142
<b>8. Estudio de Mercado .....</b>	<b>144</b>
8.1. Objetivo del Estudio.....	145
8.1.1. Mercado Objetivo.....	145
8.2. Demografía:.....	145
8.3. Área del mercado.....	146
8.3.1. Extensión Geográfica.....	146
8.3.2. Población .....	146
8.4. Investigación de Mercado .....	146
8.4.1. Tipo de Investigación .....	146

8.4.2. Técnicas de Investigación.....	147
8.4.3. Método de Investigación de Mercados .....	147
8.4.4. Ficha Técnica.....	147
8.4.5. Marco Muestral .....	148
8.4.6. Tipo de Muestreo.....	148
8.4.7. Tamaño de la Muestra .....	148
8.5. Análisis de Mercado.....	148
8.5.1. Alcance y Potencial del Mercado .....	148
8.5.2. Monetización del Software .....	149
8.5.3. Estrategia de Marketing.....	149
8.6. Encuesta .....	149
8.7. Resultados .....	152
8.8. Resultados Esperados.....	153
<b>9. Pruebas en el Software .....</b>	<b>155</b>
9.1. Pruebas en el Software .....	156
9.2. Diseño de Pruebas .....	156
2.2.1. Identificación y Estructuración de Procedimiento de Prueba.....	156
9.3. Pruebas Realizadas.....	157
2.3.1. Pruebas de Caja Negra.....	157

9.4.	Pruebas de Rendimiento.....	157
9.5.	Pruebas de Vulnerabilidad .....	158
<b>10.</b>	<b>Marketing .....</b>	<b>160</b>
10.1.	Descripción .....	161
10.2.	Organización .....	161
10.3.	Servicios.....	162
10.4.	Perfiles de clientes.....	162
10.5.	Estrategias de marketing.....	162
10.5.1.	Estrategia de retención.....	162
10.6.	Métodos de Distribución .....	163
10.6.1.	Marketing de contenidos y redes sociales.....	163
10.6.2.	Optimización de motores de búsqueda .....	163
10.6.3.	Advertising .....	163
10.7.	Medida de éxito.....	164
<b>11.</b>	<b>Despliegue .....</b>	<b>165</b>
11.1.	Cloud Computing .....	166
11.2.	Tipos de Licencia .....	166
11.2.1.	Open Source .....	166
11.3.	Términos y condiciones .....	166
11.4	Políticas y privacidad .....	168

Política de Privacidad .....	168
<b>12. SOFTWARE TERMINADO EN CALIDAD DE PRODUCTO.....</b>	<b>171</b>
12.1. Software en Calidad de Producción .....	172
12.1.1 Puntos Clave de Calidad del Software .....	172
<b>ANEXOS .....</b>	<b>175</b>
Ilustración 1 Logo de la empresa.....	15
Ilustración 2 Estructura organica .....	16
Ilustración 3 Google encontrar mi dispositivo.....	19
Ilustración 4 Google encontrar mi dispositivo.....	19
Ilustración 5 Diagrama de Gantt .....	43
Ilustración 6Proceso de SCRUM .....	55
Ilustración 7 Organigrama .....	74
Ilustración 8 https://www.cursor.com/ .....	118
Ilustración 9 https://www.atlassian.com/software/jira .....	120
Ilustración 10 Formulario de Registro de propiedad intelectual.....	129

**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y  
TELECOMUNICACIONES



## **Plan de Administración del Proyecto de Software**

### **1. PAPS**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Integrantes:**

- |                                       |           |
|---------------------------------------|-----------|
| <b>1.</b> Machuca Yavita Miguel Angel | 216063558 |
| <b>2.</b> Torrez Garcia Jorge Luis    | 218137761 |

**Gestion II-2024 Santa Cruz - Bolivia**

## **1.1. Ámbito del proyecto**

### ***1.1.1. Introducción***

#### **Situación problemática**

La empresa “BTL publicidad SRL” cuentan con la política de ofrecer y vender sus servicios de publicidad hacia a las empresas de manera presencial, lamentablemente muchos de los mercaderistas publicitarios se van a otros lugares y no cumplen con las visitas, se excusan de no tener los datos del cliente y de las empresas, provocando así bajas y quejas de los clientes de no ser atendidos llevándolos a optar por la competencia.

### ***1.1.2. Antecedentes***



*Ilustración 1 Logo de la empresa*

La empresa “BTL publicidad SRL” es una entidad dedicada a la publicidad y campaña, nuestro equipo de diseñadores y creativos talentosos trabaja en el desarrollo de elementos visuales, contenido publicitario y materiales de campaña que impacten y conecten con la audiencia de manera efectiva, realizamos un seguimiento meticoloso del rendimiento de las campañas y proporcionamos informes detallados a nuestros clientes. Esto nos permite ajustar y optimizar constantemente las estrategias para lograr los mejores resultados.

<https://btlbolivia.com/>.

## Estructura Orgánica

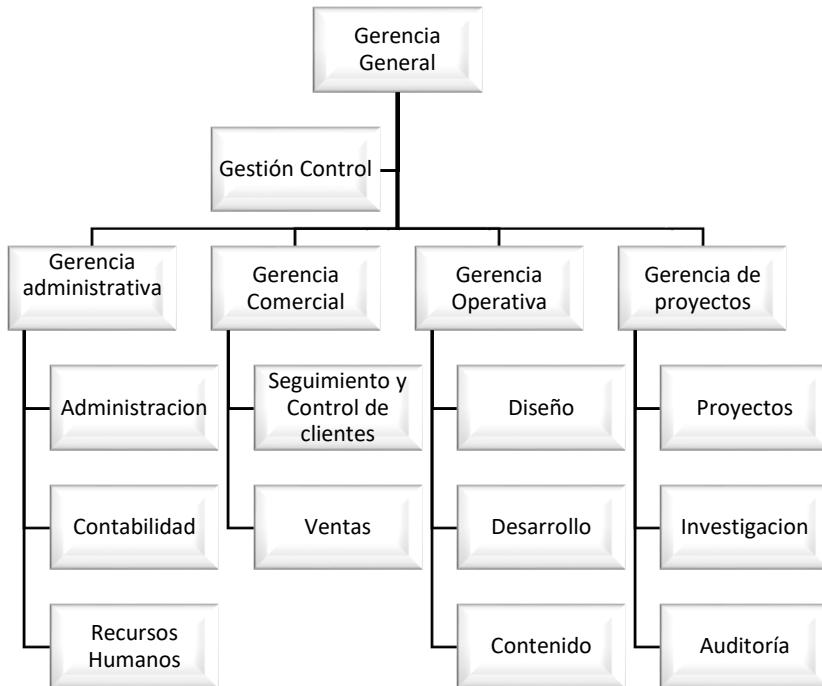


Ilustración 2 Estructura organica

## Misión

En “BTL publicidad SRL”, nuestra misión es brindar soluciones publicitarias excepcionales y estratégicas a nuestros clientes. Nos esforzamos por comprender profundamente sus necesidades y objetivos, y utilizamos nuestra creatividad, experiencia y tecnología avanzada para diseñar campañas publicitarias efectivas que generen resultados tangibles. Estamos comprometidos con la innovación, la integridad y la satisfacción del cliente en cada paso del proceso, buscando ser un socio estratégico de confianza en el éxito de las marcas que servimos

## Visión

Nuestra visión en “BTL publicidad SRL” es ser la agencia de publicidad líder y referente en el mercado, reconocida por nuestra excelencia en la creación y ejecución de campañas

publicitarias innovadoras y efectivas. Buscamos expandir nuestra presencia a nivel nacional e internacional, estableciendo relaciones a largo plazo con nuestros clientes y contribuyendo de manera significativa a su éxito. A medida que avanzamos, mantenemos nuestro compromiso con la creatividad, la responsabilidad social y la mejora continua, para seguir siendo líderes en el mundo de la publicidad y el marketing.

### ***Objetivos***

#### ***Objetivo general***

Diseñar un software para el seguimiento y control de mercaderistas publicitarios con verificación de visitas a clientes de la empresa “BTL publicidad SRL”.

#### ***Objetivos específicos***

- Definir los requisitos funcionales y no funcionales del software a través de los requerimientos.
- Definir los tiempo y tareas.
- Definir las estrategias para el funcionamiento de los módulos software.
- Realizar un análisis de las necesidades y expectativas del usuario.
- Analizar las opciones tecnológicas existentes para la implementación.
- Diseñar la arquitectura del software, de acuerdo con los requerimientos.
- Diseñar la estructura de la base de datos y los modelos necesarios para el funcionamiento del sistema.
- Implementar las funcionalidades principales del software.
- Realizar pruebas de funcionamiento del software.

## **1.2. Métricas**

### ***1.2.1. Google encontrar mi dispositivo***

Google Encontrar mi dispositivo es una función integrada en la gran mayoría de los smartphones y tablets Android por la cual puedes ver dónde están, en caso de que las hayas perdido. Hoy veremos qué es Google Encontrar mi dispositivo, cómo activarlo y qué puedes hacer con él.

#### **Características**

Localización en tiempo real: Muestra la ubicación aproximada de tu dispositivo en un mapa, siempre y cuando tenga conexión a internet.

Bloqueo remoto: Puedes bloquear tu dispositivo de forma remota, solicitando un código PIN para desbloquearlo. Esto evita que alguien más acceda a tus datos personales.

Borrado remoto: Si crees que tu dispositivo ha sido robado, puedes borrar todos los datos de forma remota para proteger tu información sensible.

Reproducir sonido: Incluso si tu dispositivo está en silencio, puedes hacer que suene a todo volumen para encontrarlo más fácilmente si está cerca.

Modo perdido: Esta opción te permite mostrar un mensaje personalizado en la pantalla de bloqueo, con tu número de teléfono o cualquier otra información que quieras compartir.

Historial de ubicaciones: Puedes ver un historial de las últimas ubicaciones conocidas de tu dispositivo.

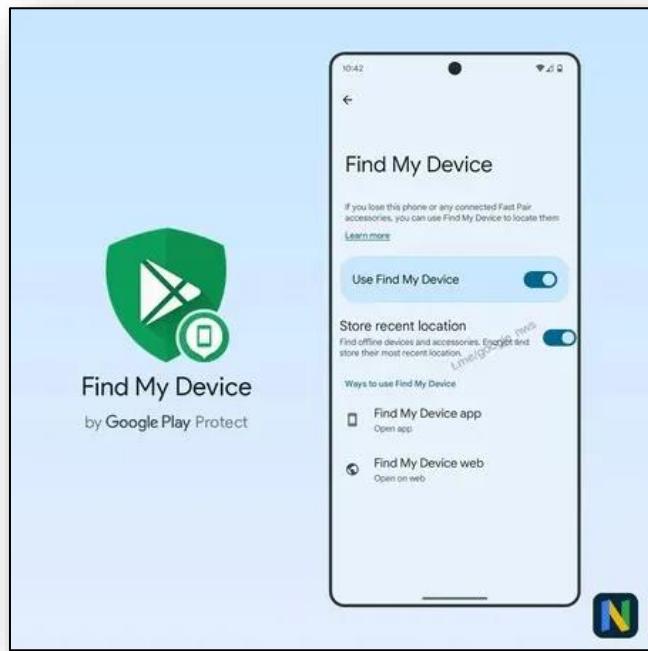


Ilustración 3 Google encontrar mi dispositivo

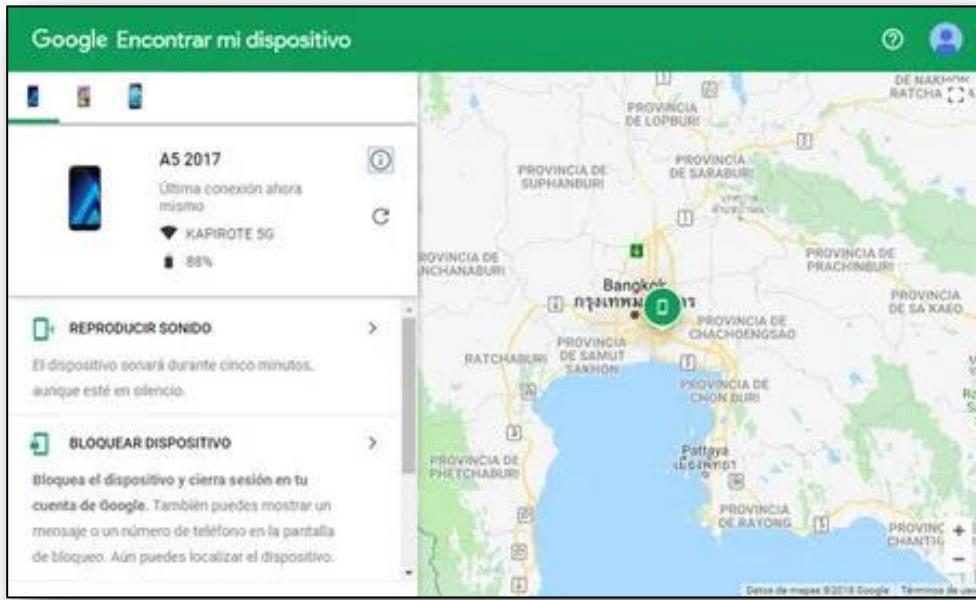


Ilustración 4 Google encontrar mi dispositivo

Esta es la página donde el usuario puede subir su foto y probarse la ropa, como indica anteriormente tiene que seguir las instrucciones al pie de la letra para poder sacar la imagen correctamente.

### *Métricas orientadas a tamaño*

Proyecto	KLDC	Tiempo (m)	Gente	Esfuerzo (p/m)	Errores	Defectos
DRESSX	10	6	5	30	25	10

Productividad= (KLDC/per. Mes)\*1000

Productividad=10/30 \* 1000 = 333

Calidad=Error + Defectos/KLDC

Calidad= (25 + 10)/10 = 3,5

### *Métricas orientadas a la función*

	FACTORES	No Influye	Incidental	Moderado	Medio	Significativo	Esencial	f
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?				X			3
2	¿Se requiere comunicación de datos en línea?					X		4
3	¿Existen funciones de procesamiento distribuido?			X				2
4	¿Es crítico el rendimiento?					X		4
5	¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?				X			3
6	¿Requiere el sistema entrada de datos interactiva?						X	5
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre	X						0

	múltiples pantallas u operaciones?						
8	¿Se actualizan los archivos maestros de forma interactiva?					X	5
9	¿Son complejas las entradas, las salidas, los archivos o las peticiones?		X				2
10	¿Es complejo el procesamiento interno?					X	4
11	¿Se ha diseñado el código para ser reutilizable?			X			2
12	¿Están incluidas en el diseño la conversión y la instalación?	X					0
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?				X		3
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?				X		3
							40

Tabla 1 Métricas de software - Google encontrar mi dispositivo

		Factor de peso			
Parámetros de medición	Cuenta	Simple	Medio	Complejo	Total
Nº de entradas de usuario	1	3	4	6	3
Nº de salidas de usuario	0	4	5	7	0
Nº de peticiones	20	3	4	6	80
Nº de archivos	5	7	10	15	50
Nº de interfaces externas	0	5	0	0	0
				Cta. Total	133

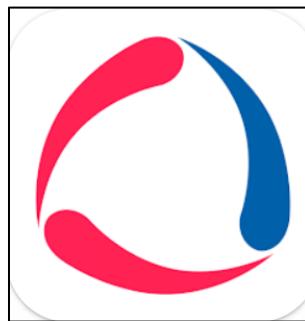
Tabla 2 Factor de peso - Google encontrar mi dispositivo

$$Pf = conteo\ total * [0.65 + 0.01 * \sum_{i=1}^{133} \lceil(F_i)\rceil]$$

$$Pf = 133 * [0.65 + 0.01 * 40]$$

$$\boxed{Pf = 140}$$

## 1.2.2. GURTAM



Gurtam es una empresa de desarrollo de software que ofrece soluciones de IoT centradas en el seguimiento de flotas. Nuestra cartera incluye productos de nicho y de última generación, plataformas basadas en la nube y en servidores, sistemas de backend telemáticos y aplicaciones móviles para uso personal. Elija la plataforma de seguimiento de vehículos que mejor se adapte a las necesidades de su negocio.

Link de la página: <https://gurtam.com/>

The screenshot shows the Gurtam software interface. On the left, there is a sidebar with various tabs: Dashboard, Seguimiento, Informes, Geocercas, Rutas, Conductores, Remolques, Notificaciones, Video, and Unidades. The main area features a map of Germany with a green route line plotted from Hannover to Bielefeld. A blue dashed line indicates an alternative route or a planned path. The map includes labels for many German cities and towns. On the right side, there is a timeline for the route, showing the start time as "Hoy" (Today) at 20:00 and the end time as "26 Mar 00:00". Below the timeline, there are buttons for "días", "horas", "minutes", and a plus sign. At the bottom of the interface, there is a footer with the Gurtam logo and the text "20:03:41 (+03)".

### *Métricas orientadas a tamaño*

Proyecto	KLDC	Tiempo (m)	Gente	Esfuerzo (p/m)	Errores	Defectos
DRESSX	15,2	12	4	48	10	5

$$\text{Productividad} = (\text{KLDC}/\text{per. Mes}) * 1000$$

$$\text{Productividad} = (15,2/48) * 1000 = 316$$

$$\text{Calidad} = \text{Error} + \text{Defectos}/\text{KLDC}$$

$$\text{Calidad} = (10 + 5)/15,2 = 0.986$$

### *Métricas orientadas a la función*

	FACTORES	No Influye	Incidental	Moderado	Medio	Significativo	Esencial	f
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?		X					1
2	¿Se requiere comunicación de datos?				X			3
3	¿Existen funciones de procesamiento distribuido?					X		5
4	¿Es crítico el rendimiento?					X		5
5	¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?					X		5
6	¿Requiere el sistema entrada de datos interactiva?					X		5
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?		X					1
8	¿Se actualizan los archivos maestros de forma					X		5

	interactiva?						
9	¿Son complejos las entradas, las salidas, los archivos o las peticiones?		X.				2
10	¿Es complejo el procesamiento interno?				X		4
11	¿Se ha diseñado el código para ser reutilizable?					X	5
12	¿Están incluidas en el diseño la conversión y la instalación?	X					0
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?			X			3
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?					X	5
							49

Tabla 3 Métricas de software - Gurtam

Factor de Peso					
Parámetros de Medición	Cuenta	Simple	Medio	Complejo	Total
Nº de entradas de usuario	3	3	4	6	9
Nº de salidas de usuario	1	4	5	7	4
Nº de peticiones de usuario	15	3	4	6	60
Nº de archivos	4	7	10	15	40
Nº de interfaces externas	0	5	7	10	0
					113
					Cta. Total 58

Tabla 4 Factor peso - Gurtam

$$Pf = conteo\ total * [0.65 + 0.01 * \sum \lceil (Fi) \rceil]$$

$$Pf = 113 * [0.65 + 0.01 * 49]$$

$$Pf = 129$$

*Driv.in*



Somos una empresa joven, dinámica y flexible, que está cambiando la forma en que se planifica y ejecuta la logística. El partner tecnológico de cientos de empresas alrededor de todo el mundo.

Como una empresa multicultural, valoramos y respetamos la diversidad. No discriminamos a ninguno de nuestros colaboradores por raza, género, orientación sexual, religión, afiliación política ni por ninguna otra condición.

Fomentamos un entorno inclusivo donde todas las voces son escuchadas y apreciadas, promoviendo la igualdad y el respeto en cada aspecto de nuestro trabajo

#### **Link de la pagina:**

<https://driv.in/es/nosotros>

#### *métricas orientadas a tamaño*

Proyecto	KLDC	Tiempo (m)	Gente	Esfuerzo (p/m)	Errores	Defectos
DRESSX	7	3	3	9	2	3

Productividad= (KLDC/per. Mes)\*1000

Productividad=7/9 \* 1000 = 777

$$\text{Calidad} = \text{Error} + \text{Defectos}/\text{KLDC}$$

$$\text{Calidad} = (2 + 3)/7 = 0.714$$

### *Métricas orientadas a la función*

	FACTORES	No Influye	Incidental	Moderado	Medio	Significativo	Esencial	f
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?		x					2
2	¿Se requiere comunicación de datos?			x				3
3	¿Existen funciones de procesamiento distribuido?					x		5
4	¿Es crítico el rendimiento?					x		5
5	¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?					x		4
6	¿Requiere el sistema entrada de datos interactiva?					x		5
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	x						1
8	¿Se actualizan los archivos maestros de forma interactiva?			x				5
9	¿Son complejos las entradas, las salidas, los archivos o las peticiones?		x.					2
10	¿Es complejo el procesamiento interno?					x		4
11	¿Se ha diseñado el código para ser reutilizable?		x					5
12	¿Están incluidas en el diseño la conversión y la instalación?	x						0
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?			x				3
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?		x					2

Tabla 5 Métricas de software – Driv.in

Factor de Peso					
Parámetros de Medición	Cuenta	Simple	Medio	Complejo	Total
Nº de entradas de usuario	2	3	4	6	6
Nº de salidas de usuario	1	4	5	7	4
Nº de peticiones de usuario	10	3	4	6	40
Nº de archivos	5	7	10	15	50
Nº de interfaces externas	0	5	7	10	0
					100

Tabla 6 Factor peso – Driv.in

$$Pf = conteo\ total * [0.65 + 0.01 * \sum_{i=1}^n f_i]$$

$$Pf = 100 * [0.65 + 0.01 * 46]$$

$$Pf = 111$$

Tabla de Métricas de los tres proyectos medidas

Proyecto	KLDC	Gente	Esfuerzo(PM)	Tiempo(en meses)	Errores	Defectos
Google encontrar mi dispositivo	10	5	30	6	25	10
GURTAM	15,2	4	48	12	10	5
Driv.in	7	3	9	3	2	3

### **1.3. Estimaciones del proyecto**

#### **Tamaño del proyecto**

##### **Alcance**

El alcance del proyecto abarca los siguientes requerimientos funcionales:

#### **Módulo de Gestión de Usuarios**

En este módulo gestionara a los usuarios así también como la asignación de sus roles y permisos que tendrán dentro del sistema.

#### **Módulo Gestión de mercaderista**

El módulo contendrá las funcionalidades que tenga que ver con el mercaderista como, el lector de qr, registrar mercaderista.

#### **Módulo de Gestión de orden**

El módulo de gestión de orden permite gestionar las órdenes que se solicitaron y las órdenes que se atendieron exitosamente.

#### **Lista de Casos de Uso**

- 1) Orden de trabajo.
- 2) Generar qr
- 3) Leer qr
- 4) Gestionar mercaderista.
- 5) Gestionar cliente.
- 6) Visualizar seguimiento.

### Factores a considerar:

- Número de módulos: 3
- Número de casos de uso: 6

### Estimación de KLDC:

#### Módulos:

Cada módulo puede tener entre 2-10 KLDC dependiendo de su complejidad

#### Casos de uso:

Cada caso de uso puede agregar entre 0.5-2 KLDC.

#### Usamos los siguientes valores para la estimación:

Módulos: 3 módulos \* 4 KLDC (promedio) = 12 KLDC

Casos de uso: 6 casos de uso \* 1 KLDC (promedio) = 6 KLDC

Total estimado: 18 KLDC

### Clasificación del Tamaño del Proyecto

- Pequeño: Menos de 10 KLDC
- **Mediano: 10 - 50 KLDC ✓**
- Grande: Más de 50 KLDC

Por tanto, concluimos que el tamaño del proyecto es mediano

Factor	Cantidad	KLDC(unidad)	KLDC TOTA
<b>Módulos</b>	3	4	12
<b>Casos de Uso</b>	6	1	6
<b>Total Estimado(KLDC)</b>			39
<b>Calificación</b>			<b>MEDIANO</b>

Tabla 7 Clasificación del tamaño del proyecto

## **Complejidad**

El equipo está conformado por desarrolladores juniors en el desarrollo web, algunas de los frameworks más utilizados por el equipo es Laravel y solo algunos están familiarizados con el desarrollo web con JavaScript, respecto al uso de APIS con inteligencia artificial es nuevo para muchos. Por lo tanto, la complejidad para el desarrollo de este software es alta para este equipo.

## **Conocimiento del equipo**

- Experiencia previa en desarrollo de aplicaciones móviles y web.
- Familiaridad con tecnologías como Laravel y PostgreSQL.

## **Desconocimiento del equipo**

- tecnologías de geolocalización o georreferenciación.
- Integración de mapas con geolocalización.
- Ubicación en tiempo real.

## **Tecnología utilizada**

- React para el backend y frontend.
- PostgreSQL para la base de datos.

## **Estructuración del cliente**

Los usuarios de este software serán un administrador, que pueda controlar todo el sistema y los mercaderistas publicitarios que accederán al sistema, pero de manera parcial, también serán parte del sistema los clientes, pero solo para almacenar su información, ellos no podrán acceder a ningún modulo del sistema.

### **1.4. Objetivos**

#### ***Objetivo general***

Diseñar un software para el seguimiento y control de mercaderistas publicitarios con verificación de visitas a clientes de la empresa “BTL publicidad SRL”.

## *Objetivos específicos*

- Definir los requisitos funcionales y no funcionales del software a través de los requerimientos.
- Definir los tiempo y tareas.
- Definir las estrategias para el funcionamiento de los módulos software.
- Realizar un análisis de las necesidades y expectativas del usuario.
- Analizar las opciones tecnológicas existentes para la implementación.
- Diseñar la arquitectura del software, de acuerdo con los requerimientos.
- Diseñar la estructura de la base de datos y los modelos necesarios para el funcionamiento del sistema.
- Implementar las funcionalidades principales del software.
- Realizar pruebas de funcionamiento del software.

## *Requerimientos funcionales*

**RF1:** Registrar usuarios, tanto clientes como empleados de la empresa.

**RF2:** Crear ordenes de trabajo, donde se asigna un mercaderista publicitario y un cliente para que se haga la visita de negocio, además de otros detalles del trabajo como la información personal del cliente y ver el estado de la orden.

**RF3:** Generar y hacer lectura código qr para la verificación de la visita de negocios del empleado.

**RF4:** Visualizar en un mapa donde están los mercaderistas publicitarios registrados en el sistema, esto se podrá realizar a través del GPS de los teléfonos corporativos que se le asigna a cada mercaderista publicitario, la ubicación que se reciba será actualizada “en tiempo real” y de esta manera se podrá saber la ubicación exacta de cada mercaderista publicitario.

### **No Funcionales:**

**RNF1** Seguridad de datos de usuarios.

**RNF2** Alta disponibilidad y escalabilidad.

**RNF3** Rendimiento óptimo en dispositivos.

### **Rendimiento**

**Base de datos:** se usará una base de datos normalizada, pero con un pequeño grado de redundancia.

**Arquitectura:** se hará uso de la arquitectura MVC.

**Interfaz:** se interactuará a través de vistas y formularios.

### **Fiabilidad**

El software que desarrollaremos es crítico, ya que gestionará información sensible y transacciones financieras de nuestros usuarios. Por lo tanto, hemos identificado varias consideraciones clave para garantizar la fiabilidad del sistema:

#### **Seguridad de Datos:**

- La aplicación debe implementar medidas robustas de seguridad para proteger los datos personales y financieros de los usuarios. Esto incluye la encriptación de datos tanto en tránsito como en reposo, y la utilización de protocolos de seguridad avanzados para proteger la información confidencial.
- Se debe asegurar el almacenamiento seguro de toda la información, utilizando servicios de almacenamiento en la nube con certificaciones de seguridad reconocidas.

#### **Estabilidad del Sistema:**

- La aplicación debe ser estable y confiable, evitando fallos o caídas del sistema. Se deben realizar pruebas exhaustivas de carga y estrés para asegurar que la plataforma puede manejar picos de tráfico sin afectar su rendimiento.

- Un plan de mantenimiento regular debe ser implementado para actualizar el software y realizar tareas de optimización, asegurando que la aplicación funcione de manera óptima en todo momento.

## 1.5. Restricciones

Para el proyecto se han identificado varias restricciones que podrían influir en el desarrollo y funcionamiento del software. Estas restricciones deben ser consideradas para asegurar que el proyecto cumpla con los requisitos y expectativas establecidos.

### 1.5.1. *Restricciones Tecnológicas*

**Compatibilidad de Dispositivos:** El software debe ser compatible con una amplia gama de dispositivos, incluidos smartphones, tabletas y ordenadores de escritorio, funcionando adecuadamente en diferentes sistemas operativos (iOS, Android, Windows, macOS).

### 1.5.2. *Restricciones Presupuestarias*

**Costos de Desarrollo:** El desarrollo de funcionalidades avanzadas como la visualización de mapas en tiempo real puede ser costoso, requiriendo la asignación de un presupuesto adecuado para cubrir los costos de licencias, herramientas de desarrollo y contratación de expertos en estas tecnologías.

**Mantenimiento y Actualizaciones:** Los costos asociados al mantenimiento continuo y las actualizaciones periódicas del software deben ser considerados dentro del presupuesto para asegurar la sostenibilidad del proyecto a largo plazo.

### 1.5.3. *Restricciones de Tiempo*

**Plazos de Entrega:** El proyecto debe ser completado dentro de un plazo determinado, lo que puede limitar el tiempo disponible para desarrollar y probar todas las funcionalidades. Se

deben establecer hitos y entregables claros para asegurar que el proyecto se mantenga en el cronograma.

#### ***1.5.4. Restricciones de Infraestructura***

**Escalabilidad:** La infraestructura subyacente debe ser capaz de escalar para manejar picos en el tráfico y la carga, especialmente durante eventos de alta demanda como promociones y ventas estacionales.

**Disponibilidad de Servicios:** La disponibilidad de servicios externos, como las pasarelas de pago y los servicios de envío, puede afectar la operatividad del software. Se deben establecer redundancias y planes de contingencia para mitigar posibles interrupciones.

## 1.6. Interfaces Externas

### 1.6.1. Interacciones con Software

El software hará el uso de Google maps, para poder ver los mapas y realizar la geolocalización de los vendedores.

### 1.6.2. Interacciones con Personas

La aplicación principalmente interactúa con el ser humano

### 1.6.3. Interacciones con Hardware

El software no cuenta con ninguna interacción con algún hardware externo.

## 1.7. Estimación del proyecto

### 1.7.1. Valor esperado

Proyecto	Optimista	Más probable	Pesimista	Esperadas
	6 meses	8 meses	12 meses	8,33 meses

Fórmula:

$$\text{Valor Esperado} = \frac{(O + 4M + P)}{6}$$

donde:

- $O$  = Estimación Optimista
- $M$  = Estimación Más Probable
- $P$  = Estimación Pesimista

$$VE = (\text{Optimista} + (4 * \text{Mas probable}) + \text{Pesimista}) / 6$$

$$VE = (6 + (4 * 8) + 12) / 6$$

$$VE = 8.33 \text{ meses}$$

### 1.7.2. COCOMO II

Tabla de Puntos de Objeto (P.O.)					
Tipo de Objeto	Cuenta	Básico	Intermedio	Avanzada	Total
Pantalla	12	1	2	3	24
Reporte	2	2	5	8	10
Componente 3GL	2	-	-	10	20
<b>Total</b>					<b>54</b>

Tabla 8 Puntos de Objeto (P.O.)

$$NOP = (Puntos\ de\ Objeto) \times [(100 - Porcentaje\ rehuso)/100]$$

$$NOP = (54) \times [(100 - 20)/100] = 43$$

Proporción de Productividad	Muy baja	Baja	Normal	Alta	Muy alta
Experiencia / Capacidad del desarrollador		x			
Madurez / Capacidad del entorno			x		
PROD	4	7	15	25	50

Tabla 9 Proporción de productividad

$$\text{PROD} = 7 + 15 = 22$$

$$\text{Esfuerzo Estimado} = NOP / PROD$$

$$\text{Esfuerzo Estimado} = 43/22 = 1.95 = 2 \text{ persona-meses}$$

### 1.7.3. Historias de usuario (*Planning Poker*)

Gestionar cliente	Prioridad	PHU	1er Ronda	2da Ronda
El cliente se podrá registrar, el adm podrá gestionar a los usuarios registrados.	Alta	PHU: 8	PHU= 8 PHU= 5 PHU= 3 PHU= 8 PHU= 5	PHU= 8 PHU= 8 PHU= 8 PHU= 8 PHU= 8
<b>Detalles:</b>				
<p><b>1.7.4. 1. El cliente navega hacia el sitio web donde se podrá registrar.</b></p>				PHU = 8
<p>2. El administrador gestionara a los usuarios registrados mediante un panel.</p>				

Tabla 10 HU – Gestionar cliente

Gestionar mercaderista	Prioridad	PHU	1er Ronda	2da Ronda
El mercaderista será registrado en el sistema.	Alta	PHU: 8	PHU= 8 PHU= 5 PHU= 3 PHU= 8 PHU= 5	PHU= 8 PHU= 8 PHU= 8 PHU= 8 PHU= 8
<b>Detalles:</b>				
<p>1. El mercaderista será registrado por el administrador a través del software.</p>				PHU = 8
<p>2. El administrador será quien gestione a los mercaderistas mediante un panel del software.</p>				

Tabla 11 HU – Gestionar mercaderista

Orden de trabajo	Prioridad	PHU			
Orden de trabajo	Alta	PHU: 8			
<b>Detalles:</b>					
1. Se podrá registrar las ordenes de trabajo en el software por parte del administrador mediante un panel del software.					
2. El administrador podrá gestionar las ordenes de trabajo con un panel del software.					<b>PHU = 8</b>
3. El administrador podrá elegir a un mercaderista para una orden de trabajo mediante el software.					

Tabla 12 HU- Orden de trabajo

Generar Qr	Prioridad	PHU			
Generar Qr	Alta	PHU: 5			
<b>Detalles:</b>					
1. Cada Orden de Trabajo tendrá su propio Qr generado al momento de ser creado la orden de trabajo.					
2. Se podrá descargar el Qr. generado del orden de trabajo.					<b>PHU = 5</b>
3. Los Qr. serán enviado al cliente para que el mercaderista lo escanee.					

Tabla 13 HU – Generar Qr

Leer Qr.	Prioridad	PHU	1er Ronda	2da Ronda
Leer Qr.	Alta	PHU: 5	PHU= 5	PHU= 5
<b>Detalles:</b>			PHU= 8	PHU= 5
			PHU= 5	PHU= 5
			PHU= 3	PHU= 5
			PHU= 5	PHU= 5

1. El software tendrá un lector de Qr. el cual utilizará el mercaderista.

2. El mercaderista utilizará el lector de Qr. y podrá registrar su presencia gracias al Qr. que le otorgaron al cliente

Tabla 14 HU – Leer Qr

Visualizar seguimiento	Prioridad	PHU	1er Ronda	2da Ronda
Visualizar seguimiento	Alta	PHU: 5	PHU= 5	PHU= 5
<b>Detalles:</b>			PHU= 8	PHU= 5
			PHU= 5	PHU= 5
			PHU= 3	PHU= 5
			PHU= 5	PHU= 5

1. El administrador podrá ver la ubicación de las ordenes de trabajo en un mapa a través del software.

2. El administrador podrá ver la ubicación en tiempo real de los mercaderistas mediante panel de la aplicación.

Tabla 15 HU – Visualizar seguimiento

## 1.8. Gestión de riesgos

Riesgo	Prob.	Impacto	Plan de aversión	
			%	Reducir presencia
Programador abandona el equipo	55%	Significativo	Salario acorde Capacitación constante Vacaciones Comunicación asertiva	Promover el uso de sistemas de control, para ver los cambios realizados y procesos  Tener una buena documentación, para que los requerimientos y el flujo del software este detallado  Capacitación de parte del programador que se irá, a su respectivo reemplazo
Retraso en la entrega de sprint	50%	Significativo	Los sprints deben ser alcanzables y dadas según la experiencia del programador  Mejor control de parte del product owner para hacer seguimiento  Usar correctamente la metodología ágil	Dar prioridad al sprint atrasado en el siguiente sprint

Problemas de compatibilidad	30%	significativo	<p>Realizar los testings necesarios para evitar problemas de compatibilidad</p> <p><b>Uso de herramientas de testing</b></p>	Poner como tarea en los Sprint los problemas de compatibilidad en el software
Baja calidad del código	40%	critico	<p>Seguir practicas solidas en el desarrollo para código limpio</p> <p>Estructurar de manera óptima el proyecto</p> <p>Seguir patrones de diseños adecuados al proyecto</p>	Tener un equipo que se encarga de revisar que se esté usando el estándar adecuado para codificación para solucionarlos.
Cambios en los requisitos del proyecto	60%	critico	<p>Tener bien claro de que trata el proyecto</p> <p>Tener en cuenta los cambios que pueden surgir en el proyecto, y estructurar el proyecto teniendo esto en mente.</p> <p>Usar metodología ágil</p>	Hacer una pre-investigación, sobre el proyecto, haciendo un análisis de mercado para tener claros todos los requisitos que se solventarán en el proyecto

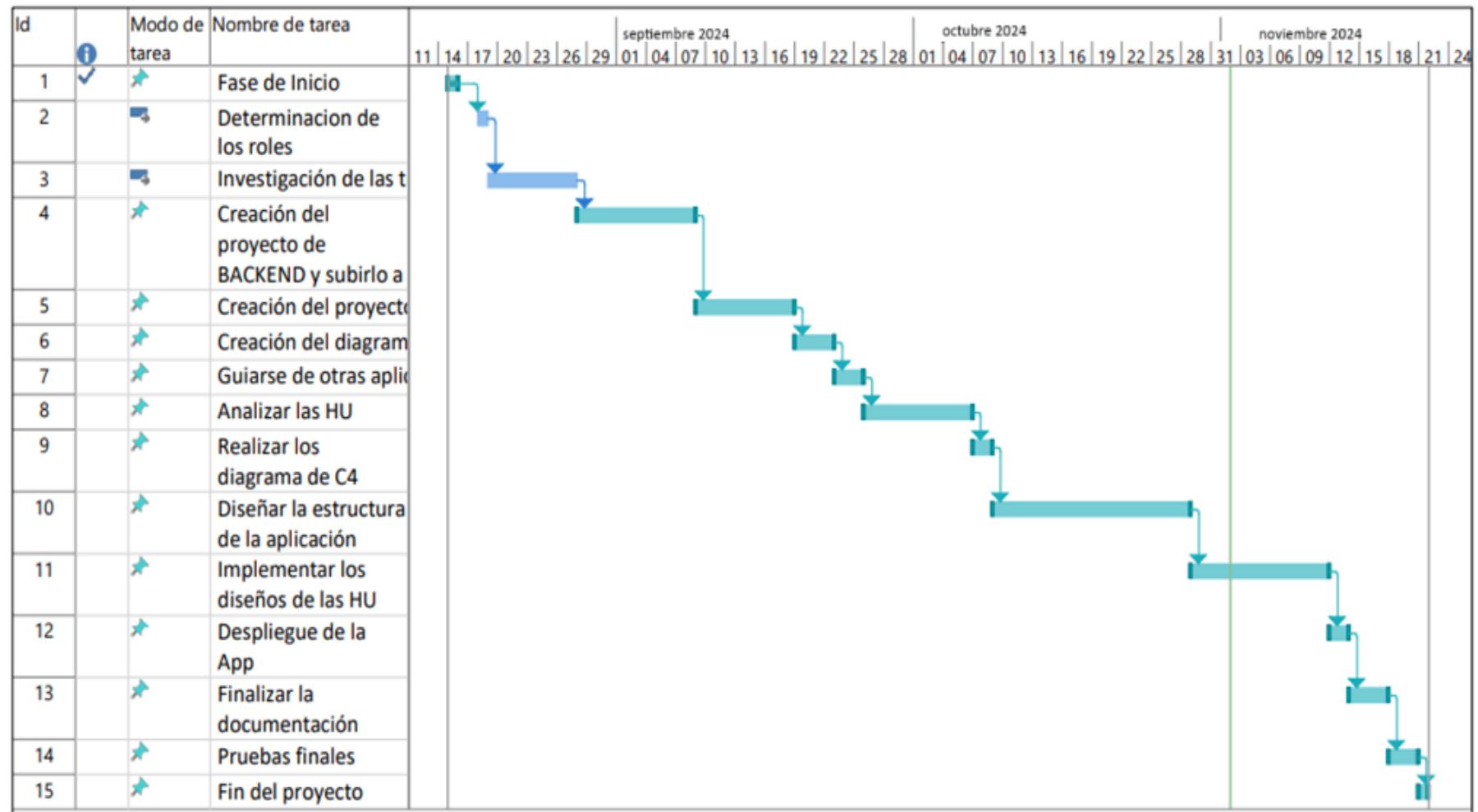
Tabla 16 Gestión de riesgos

## 1.9. Planificación del tiempo

Tabla 17 Actividades para la planificación del tiempo

Código	Actividad	Duración	Fecha de Inicio	Fecha de Fin
A1	Fase de Inicio	1 día	mar 27/08/24	mar 27/08/24
A2	Determinación de los roles	1 día	mié 28/08/24	mié 28/08/24
A3	Investigación de las tecnologías de IA	7 días	jue 29/08/24	lun 09/09/24
A4	Creación del proyecto de BACKEND y subirlo a GITHUB	8 días	mar 10/09/24	jue 19/09/24
A5	Creación del proyecto FRONTEND y subirlo a GITHUB	8 días	vie 20/09/24	mar 01/10/24
A6	Creación del diagrama de clases	2 días	mié 02/10/24	jue 03/10/24
A7	Guíarse de otras aplicaciones similares	3 días	vie 04/10/24	mar 08/10/24
A8	Analizar las HU	7 días	mié 09/10/24	jue 17/10/24
A9	Realizar los diagrama de C4	2 días	vie 18/10/24	lun 21/10/24
A10	Diseñar la estructura de la aplicación	14 días	mar 22/10/24	vie 08/11/24
A11	Implementar los diseños de las HU	10 días	lun 11/11/24	vie 22/11/24
A12	Despliegue de la App	2 días	lun 25/11/24	mar 26/11/24
A13	Finalizar la documentación	2 días	mié 27/11/24	jue 28/11/24
A14	Pruebas finales	3 días	vie 29/11/24	mar 03/12/24
A15	Fin del proyecto	1 día	mié 04/12/24	mié 04/12/24

### 1.9.1. Diagrama de Gantt



### 1.9.2. Diagrama de PERT

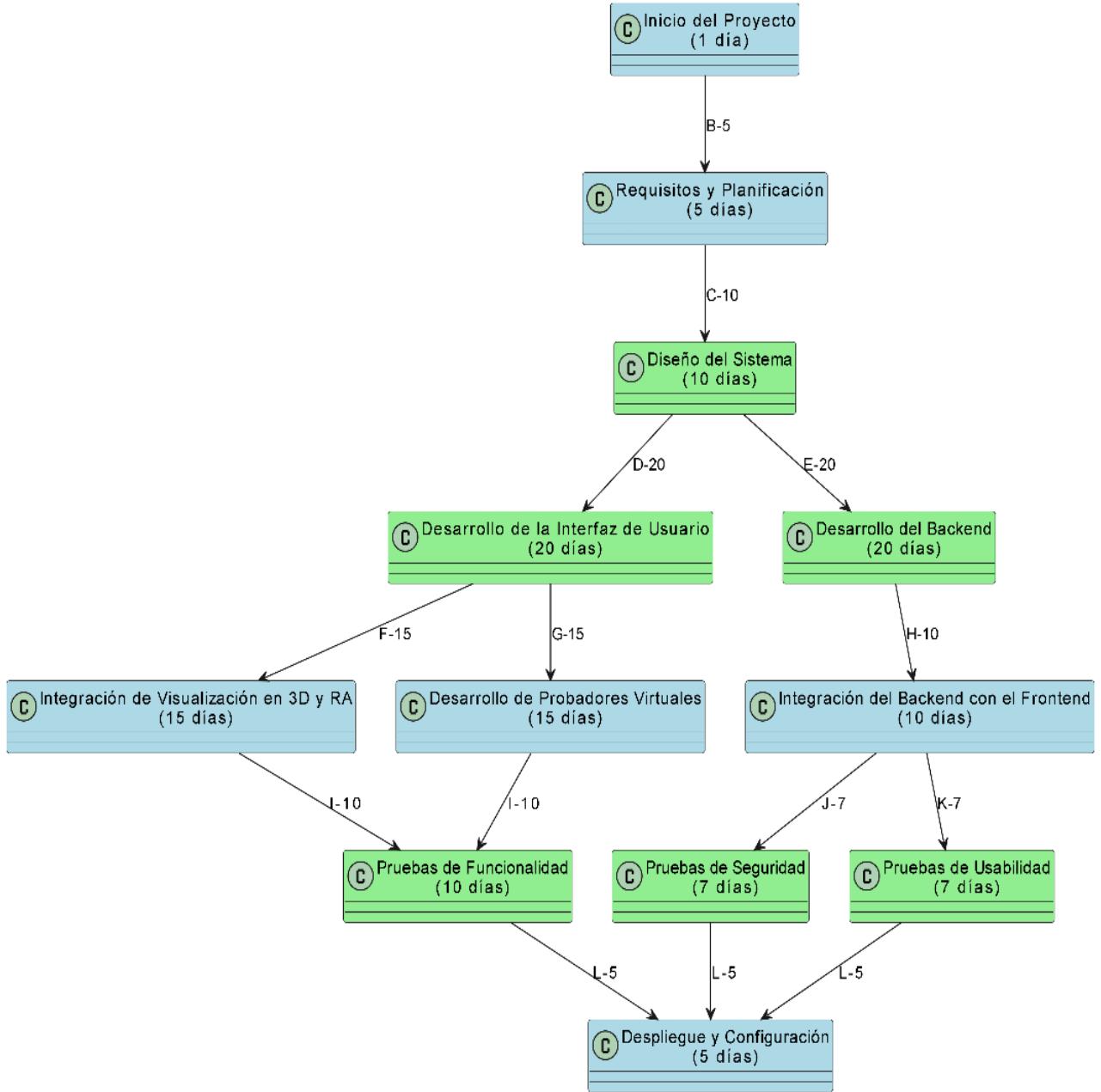


Tabla 18 Diagrama de PERT

## 1.10. Tabla de recursos

Recurso	Desde	Hasta	Tiempo (año)	Cantidad	Precio Unitario	Depreciación	Precio Neto	Total
<b>Hardware</b>	05/08/2024	05/12/2024						0,0
PC(A2)	05/08/2024	05/12/2024	0,33		25850,00	0,25		11700,0
Impresora(A3)	05/08/2024	05/12/2024	0,33	1	1144,00	0,25		1144,0
Celulares(A4)	05/08/2024	05/12/2024	0,33	4	1500,00	0,25		6000,0
<b>Software</b>	05/08/2024	05/12/2024						0,0
SO(A5)	05/08/2024	05/12/2024	0,33		1 1800,00	0,33		1800,0
Hosting(A6)	05/08/2024	05/12/2024	0,33		1 140,00	0,33		140,0
<b>Personal</b>	05/08/2024	05/12/2024						0,0
Analista	05/08/2024	05/12/2024	0,33	1			7000,0	7000,0
Diseñador	05/08/2024	05/12/2024	0,33	1			5000,0	5000,0
Desarrollador	05/08/2024	05/12/2024	0,33	2			6500,0	13000,0
Limpieza	05/08/2024	05/12/2024	0,33	1			1000,0	1000,0
<b>Infraestructura</b>	05/08/2024	05/12/2024						0,0
Alquiler Oficina	05/08/2024	05/12/2024	0,33	1	2000,00			2000,0
Escritorios(A8)	05/08/2024	05/12/2024	0,33	2	650,00	0,2		1300,0
Sillas(A9)	05/08/2024	05/12/2024	0,33	6	380,00	0,2		2280,0
<b>Logística</b>	05/08/2024	05/12/2024						0,0
Material de escritorio	05/08/2024	05/12/2024	0,33	1	200,00			200,0
Cursos(A1)	05/08/2024	05/12/2024	0,33	3	140,00			420,0
Alimentos	05/08/2024	05/12/2024	0,33	1	1800,00			1800,0
<b>TOTAL BS.-</b>								<b>54784,00</b>

Tabla 19 Tabla de Recursos

## 1.11. Organización interna

PERSONA	ROL	CARACTERÍSTICA
Jorge Luis Torrez Garcia	Product Owner	<b>Responsabilidad:</b> Será responsable de la coordinación y la comunicación efectiva en el equipo de desarrollo. Actuará como mediador para comprender las necesidades y prioridades del cliente y definir los requisitos de la app <b>Asistente de cocina.</b> <b>Características:</b> Debe tener una comprensión profunda de los objetivos del asistente de cocina y estar en sintonía con las expectativas de los clientes.
Machuca Yavita Miguel Angel	Scrum Master	<b>Responsabilidad:</b> El Scrum Master garantizará que el equipo de desarrollo siga las prácticas y principios de Scrum. Su enfoque principal será facilitar el proceso Scrum y eliminar obstáculos para el equipo. <b>Características:</b> Debe ser un experto en Scrum y estar capacitado para guiar al equipo en la implementación de esta metodología. Su papel es fundamental para mantener un desarrollo ágil y eficiente del proyecto.
Jorge Luis Torrez Garcia Machuca Yavita Miguel Angel	Developer	<b>Responsabilidad:</b> El equipo de desarrollo será responsable de la creación y desarrollo de los artefactos y funcionalidades requeridos para avanzar en el proyecto. Esto incluye la codificación del software y la construcción de soluciones tecnológicas. <b>Características:</b> Deben ser profesionales competentes en desarrollo de software y trabajar de manera colaborativa para cumplir con los objetivos del proyecto.

Tabla 20 Organización interna - roles

### 1.11.1. Descentralizada Democrática

Una organización Descentralizada Democrática (DD) se caracteriza por la toma de decisiones distribuida entre todos los miembros del equipo, fomentando la participación y el consenso.

#### Justificación:

##### Fomento de la Innovación y Creatividad:

La estructura DD nos permite que todos los miembros del equipo aporten ideas y soluciones, aprovechando las distintas perspectivas y habilidades de cada uno de los integrantes

de nuestro grupo. Esto es crucial ya que en nuestro proyecto involucra la inteligencia artificial y experiencia de usuario, donde la creatividad puede marcar una gran diferencia.

#### **Mayor Motivación y Compromiso:**

Al tener una voz activa en las decisiones, cada uno de nosotros que conformamos el equipo nos sentimos más valorados y motivados. Esto aumenta el compromiso con el proyecto y puede mejorar la calidad del trabajo y la eficiencia del equipo.

#### **Flexibilidad y Adaptabilidad:**

La estructura DD nos permite una adaptación rápida a los cambios y nuevos desafíos, en un entorno de desarrollo ágil, esta flexibilidad es esencial para el éxito del proyecto.

#### **Distribución Equitativa de la Responsabilidad:**

En este caso la toma de decisiones compartida distribuye la responsabilidad entre todos nosotros que conformamos el equipo, reduciendo la carga sobre un solo individuo y fomentando un sentido de propiedad colectiva sobre el proyecto.

#### **Mejora de la Comunicación:**

La estructura DD promueve una comunicación abierta y transparente, esencial para identificar y resolver problemas rápidamente. Esto es especialmente importante en un equipo multidisciplinario que trabaja en diferentes módulos del proyecto.

#### **Comparación con Otras Estructuras:**

##### ***1.11.2. Descentralizada Controlada (DC):***

En una estructura DC, aunque la toma de decisiones está distribuida, existen controles y supervisión más estrictos. Esto puede limitar la creatividad y la innovación, aspectos clave para el desarrollo del asistente de cocina.

### **1.11.3. Centralizada Controlada (CC):**

Una estructura CC centraliza la toma de decisiones en una o pocas personas, lo que puede llevar a una menor participación del equipo y menos innovación. Además, puede crear cuellos de botella en la toma de decisiones, retrasando el progreso del proyecto.

## **1.12. Mecanismos de seguimiento y control**

### **Reunión Técnica Formal (RTF) y Diseño de Formulario (DF)**

Se realizará reuniones periódicamente cumpliendo un enfoque de aplicación de scrum como marco de trabajo para el desarrollo de software. Nuestras reuniones fueron programadas para cualquier día disponible a las 20:00, con una duración mínima de 30 minutos, para mostrar los avances y progresos que se ha tenido en grupo. Durante las reuniones a cada integrante el scrum master hace las siguientes preguntas:

<b>¿Cuál es el avance que se ha tenido con respecto a la tarea asignada?</b>	<b>Esto para saber en lo que se ha estado trabajando en la semana, y el tiempo que le ha dedicado.</b>
<b>¿Se tuvo algún inconveniente con respecto a la tarea que se le asignó?</b>	Para saber los inconvenientes, o dudas que se ha tenido al momento de desarrollar la tarea que se le asignó, para así afinar detalles.
<b>En caso de que no haya terminado la tarea asignada ¿Para cuándo crees poder culminar con la tarea que le fue asignada?</b>	Para tener un tiempo estimado de culminación de la tarea asignada, y poder asignarle otra tarea o funcionalidad.

Pregunta Grupal ¿Tienen recomendaciones o sugerencias para el desarrollo de software?	Para mejorar el desarrollo de software o sus funcionalidades.
---	---

Tabla 21 preguntas del scrum master

- **Dayli SCRUM**

DÍAS	LISTA DE ACTIVIDADES PARA EJECUTAR EL SPRINT 1	ACTIVIDADES PARA SU FUNCIONAMIENTO	DIFICULTAD EN EJECUCIÓN	RESPONSABLES	ROL
Agosto 29/08/2024	Reunión grupal de manera presencial	Aclaración de actividades para su desarrollo Aclaración de observaciones de cada punto	Falta de coordinación Comprensión no inmediata	Machuca Yavita Miguel Angel Jorge Luis Torrez García	Product Owner Scrum Master Desarrollador Desarrollador Desarrollador
septiembre 05/09/2024	Reunión grupal por Zoom para la documentación	Ánalisis y corrección	Poco entendimiento en la parte del alcance y perfil	Machuca Yavita Miguel Angel Jorge Luis Torrez García	Product Owner Scrum Master Desarrollador Desarrollador Desarrollador
Septiembre 10/09/2024	Reunión grupal por Zoom para la documentación y diseño	Diseño de la Base de Datos	Ninguna	Machuca Yavita Miguel Angel Jorge Luis Torrez García	Product Owner Scrum Master Desarrollador Desarrollador Desarrollador

Tabla 22 Dayli SCRUM inicial

- ***Sprint Review***

N. TAREAS	LISTA DE ACTIVIDADES PARA EJECUTAR EL SPRINT 1	RESPONSABLE	ROL	OBSERVACIONES	ESTADO
1	Diseño de la BD	Jorge Luis Torrez García	Desarrollador	Ninguna	Completo
2	Implementación de la Base de Datos	Jorge Luis Torrez García	Desarrollador	Ninguna	Completo
3	Creación del Endpoint de Registro de Usuario	Jorge Luis Torrez García	Product Owner	Incompleto	Incompleto
4	Creación del Endpoint de Inicio de Sesión	Jorge Luis Torrez García	Product Owner	Ninguna	Completo
5	Diseño de la Interfaz de Registro de Usuario	Machuca Yavita Miguel Angel	Desarrollador	Ninguna	Completo
6	Diseño de la Interfaz de Inicio de Sesión	Machuca Yavita Miguel Angel	Desarrollador	Ninguna	Completo
7	Implementación del Backend de Registro de Usuario	Machuca Yavita Miguel Angel	Scrum Master	Ninguna	Completo
8	Implementación del Backend de Inicio de Sesión	Machuca Yavita Miguel Angel	Scrum Master	Ninguna	Completo

9	Creación de la Página de Listado de Mercaderista	Jorge Luis Torrez García	Desarrollador	Ninguna	Completo
10	Creación de la Página de Detalles del Producto	Machuca Yavita Miguel Angel	Desarrollador	Ninguna	Completo
11	Implementación del Endpoint para Obtener qr	Machuca Yavita Miguel Angel	Scrum Master	Ninguna	Completo
12	Implementación del Endpoint para Obtener Detalles de las ordenes	Jorge Luis Torrez García (JLT)	Product Owner	Ninguna	Incompleto
13	Diseño de la Funcionalidad de Carrito de Compras	Jorge Luis Torrez García (JLT)	Desarrollador	Ninguna	Completo
14	Implementación del Backend	Machuca Yavita Miguel Angel	Scrum Master	Incompleto	Incompleto
15	Pruebas de Usabilidad y Funcionalidad del Registro y Login de Usuarios	Jorge Luis Torrez García (JLT)	Desarrollador	Incompleto	Incompleto

Tabla 23 Sprint Review de las tareas iniciales

### 1.13. Tabla de recursos

NOMBRE	DESCRIPCIÓN	PRECIO	IMAGEN
A1: <b>Servidor DELL POWEREDGE T440</b>	Marca DELL, Modelo Poweredge t440, t440q3fy22pev1, tower server, intel xeon silver 4208, 16gb, 2tb sata hot-plug hd, perc h330/idrac9 basic, procesador intel xeon modelo silver 4208 velocidad 2.1 ghz.	2560 \$US	 <p>Imagen referencial</p>
A2: <b>Lenovo ThinkPad E16</b>	Lenovo ThinkPad E16 Home & Business Laptop (AMD Ryzen 5 7530U 6-Core, 16GB RAM, 512GB PCIe SSD, AMD Radeon, 15.6" FHD(1920x1080), WiFi 6, BT 5.2, Backlit, FP, Webcam, Win 10 Pro) W/Dockztorm Hub	800 \$US	

<b>A3: Smartphone POCO X6 PRO</b>	Xiaomi Poco X6 PRO 5G + 4G LTE Global Unlocked (512GB + 12GB) GSM 6.67" 64MP Triple Camera <b>PRECIO:</b> 400 \$us	400 \$us	
<b>A4: Windows 10 Pro</b>	Licencia de Windows 10 PRO 64 bits	219 \$us	
<b>A5: Windows Project</b>	Licencia de Windows Project 64 bits	10 \$us	
<b>A6: Oficina</b>	1 ambientes · 1 baño · 1 garajes · 42.00 m2 alquila tu oficina en el centro de la ciudad (torre cainco) superficie: 4167 mts2	500 \$us mensuales alquiler	

**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y  
TELECOMUNICACIONES



## **2. Modelo de Desarrollo de Software**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Integrantes:**

- |                                |           |
|--------------------------------|-----------|
| 1. Machuca Yavita Miguel Angel | 216063558 |
| 2. Torrez Garcia Jorge Luis    | 218137761 |

## 2.1. PROCESO DE DESARROLLO DE SOFTWARE “SCRUM”

Utilizaremos el patrón de desarrollo Scrum, aplicaremos reuniones diarias de seguimiento, retrospectivas al final de cada Sprint y revisiones con el cliente al final de cada Sprint.

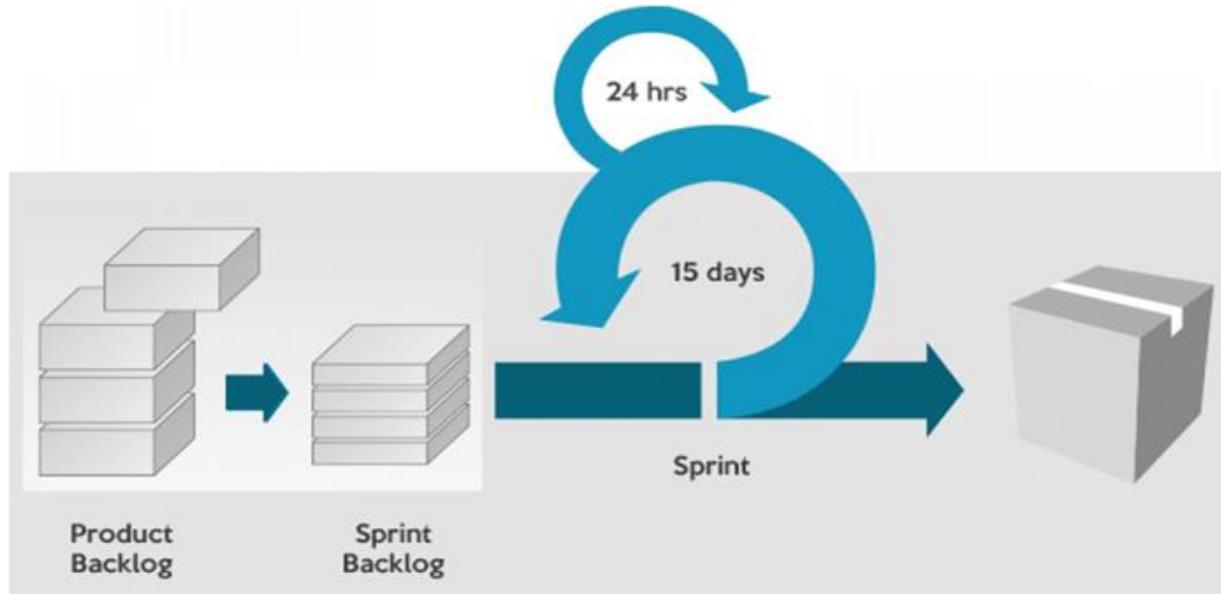


Ilustración 6Proceso de SCRUM

### 2.1.1. Requisitos Iniciales

- **Usuarios:** Se define los usuarios que existirán en el sistema: Mercaderista, Cliente, Personal RRHH
  - **Mercaderista:** Tendrá acceso a las órdenes de trabajo que se le asignen así también como a un listado de cliente. No necesita conocimientos técnicos
  - **Cliente:** Este usuario tendrá acceso al QR para que el mercaderista registre su visita. No necesita conocimientos técnicos.
  - **Personal RRHH:** Este usuario es el encargado de administrar el sistema, deberá tener mínimo conocimientos informáticos. Conocimientos de administración de personal y conocimientos básicos de computación.

### **2.1.2. *Equipos de Desarrollo***

<b>Persona</b>	<b>Rol</b>	<b>Característica</b>
Machuca Yavita Miguel Angel	Scrum Master	Hacerse cargo de la realización de todo el Sprint 0
Torrez Garcia Jorge Luis	Product Owner	Especificar sobre todos los aspectos que requiere el software

### **2.1.3. *Definición del Patrón de Desarrollo***

El patrón de desarrollo seleccionado para este proyecto es el Modelo-Vista-Controlador (MVC). MVC es un patrón arquitectónico ampliamente utilizado que separa la lógica de negocio, la presentación y la interacción del usuario en tres componentes distintos. Esta separación de responsabilidades mejora el modularidad y facilita el mantenimiento y la evolución del software a lo largo del tiempo.

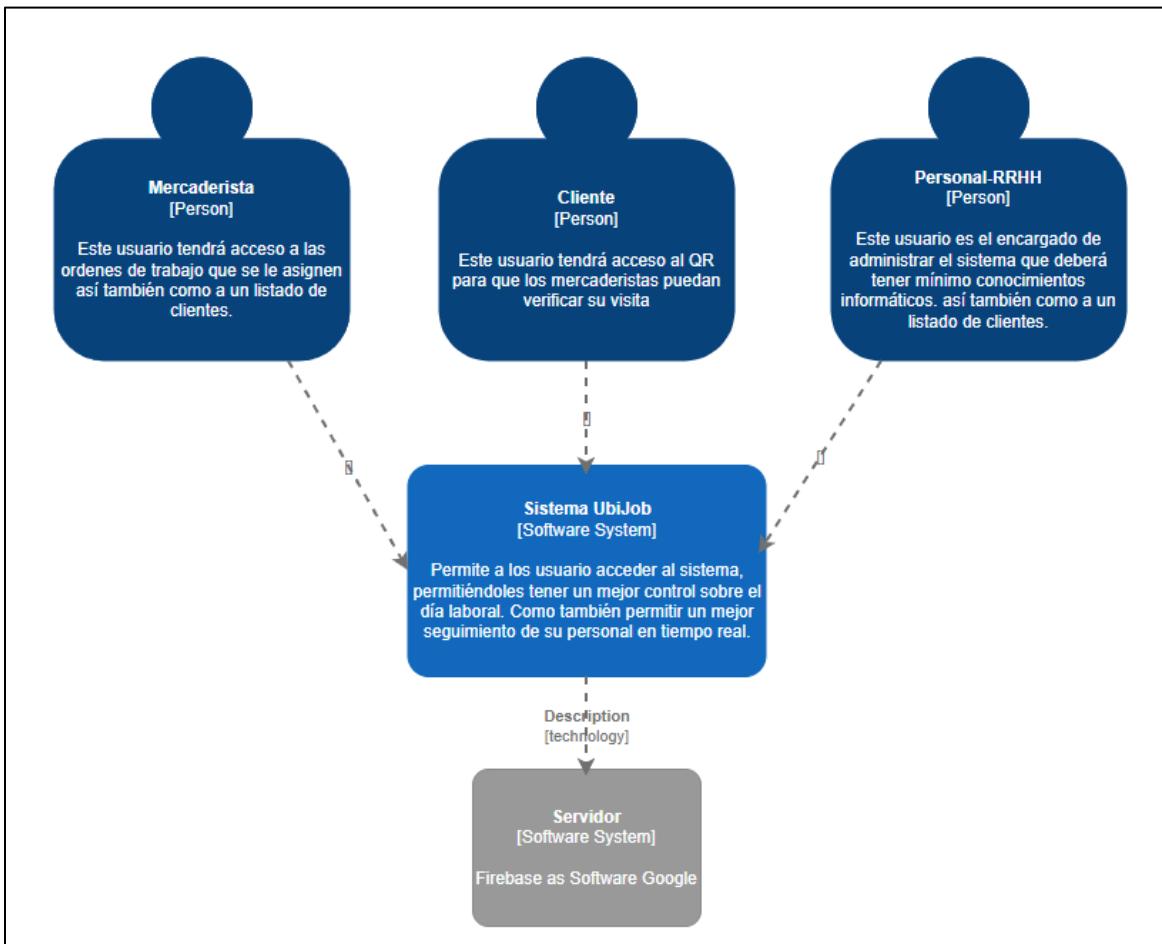
### **2.1.4. *Duración del Sprint***

Este proyecto tendrá 3 sprint con una duración de 3 semanas aproximadamente.

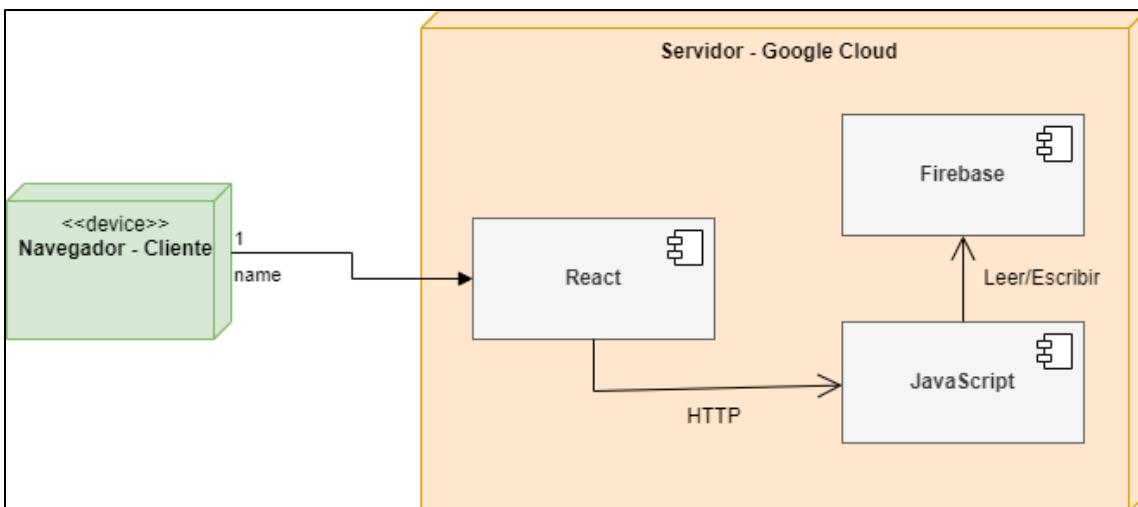
### **2.1.5. *Herramientas de Gestión de Proyecto***

Para una gestión efectiva del proyecto, hemos seleccionado Jira como nuestra herramienta de gestión de proyectos. Jira es una plataforma líder en la industria que nos permitirá planificar, asignar y hacer un seguimiento detallado del progreso de las tareas en cada Sprint. Con Jira, esperamos optimizar la colaboración y alcanzar los objetivos del proyecto de manera eficiente y organizada.

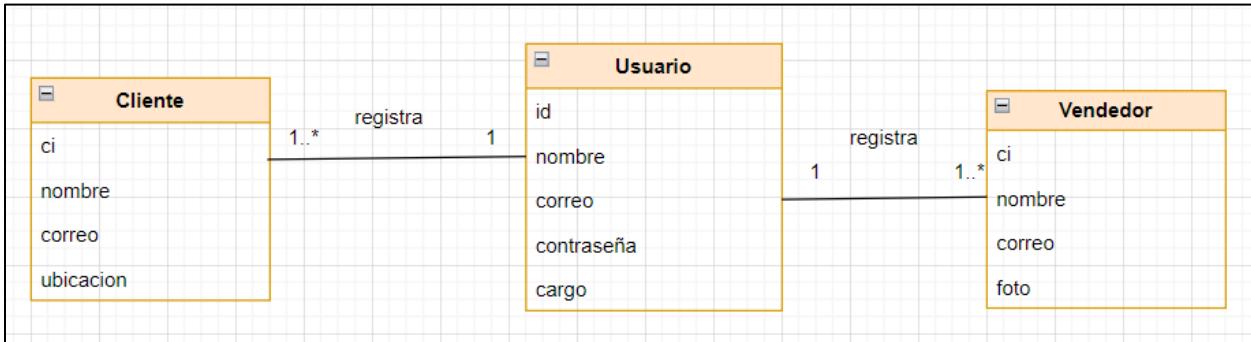
### 2.1.6. Modelo de contexto inicial



### 2.1.7. Modelo de Arquitectura Inicial



## 2.1.8. Base de datos inicial



## 2.1.9. Producto Backlog Inicial

PRODUCT BACKLOG				
Proyecto		Software Para El Control Y Seguimiento De Los Reponedores De Una Empresa		
Product Owner		Jorge Luis Torrez Garcia		
Versión		v1.0.0		
Id	Rol	Funcionalidad	Razón/Resultado	Prioridad
1	Análisis	Elaborar historia de usuario para usuario	Definir los requisitos de la huusuario	Alta
2	Diseño	Diseñar interfaz de usuario para usuario	Diseño de interfaz	Alta
3	Implementación/ Desarrollo	Implementar la historia de usuario de usuario	Desarrollo e implementación de los requerimientos.	Alta
4	Pruebas	Realizar pruebas a la implementación	Observaciones sobre 1 implementado o	Alta
5	Análisis	Elaborar historia de usuario para Gestionar Vendedor	Definir los requisitos de la hu gestionar vendedor	Alta
6	Diseño	Diseñar interfaz de usuario para Gestionar Vendedor	Diseño de interfaz	Alta
7	Implementación/ Desarrollo	Implementar la historia de usuario de Gestionar Vendedor	Desarrollo e implementación de los requerimientos.	Alta
8	Pruebas	Realizar pruebas a la implementación	Observaciones sobre 1 implementado o	Alta
9	Diseño	Diseñar Interfaz de Cliente	Diseño de interfaz	Alta

<b>10</b>	Desarrollo	Implementar Interfaz de Cliente	Desarrollo implementación de los requerimientos.	e Alta
<b>11</b>	Desarrollo	Implementar CRUD de cliente	Desarrollo implementación de los requerimientos.	e Alta
<b>12</b>	Diseño	diseñar interfaz para generar orden de trabajo	Diseño de interfaz	Alta
<b>13</b>	Desarrollo	Integrar librería para generarQR	Integración de librerías en el desarrollo del proyecto para generar qr	Alta
<b>14</b>	Desarrollo	Integrar API Maps para visualizar seguimiento	Integración de API en el desarrollo del proyecto para visualizar seguimiento	Alta
<b>15</b>	Diseño	Desplegar Mapa en interfaz web	Poder visualizar el mapa en el sitio web.	Alta

## 2.2. Sprint 3

### 2.2.1. Objetivos del Sprint

Diseñar Interfaz, implementar CRUD de usuario

### 2.2.2. Patrón de desarrollo por Historia de Usuario

#### Historias de Usuario

							
HU-Nro.	Nombre Corto del HU	Modulo	Estimación	Desarrollador			
1	Gestionar Usuarios	Inicio	2 día	Torrez Garcia Jorge Luis			
<b>Como:</b>	Análisis						
<b>Quiero:</b>	Poder iniciar sesión con mis credenciales de usuarios						
<b>Para:</b>	Acceder a las funciones personalizadas de la aplicación						
<b>Descripción:</b>	Se debe implementar un sistema de inicio de sesión que permita a los usuarios autenticarse con sus credenciales.						
Proceso/Lógica							
1) El usuario abre la aplicación e ingresa a la pantalla de inicio de sesión.		2) La aplicación verifica las credenciales ingresadas con los datos almacenados en la base de datos.					
Criterios de Aceptación							
1) El usuario puede ingresar su nombre de usuario y contraseña. 2) La autenticación es exitosa si las credenciales son correctas. 3) Se muestra un mensaje de error si las credenciales son incorrectas.							
Prioridad	Alta	Estimación PHU	1				

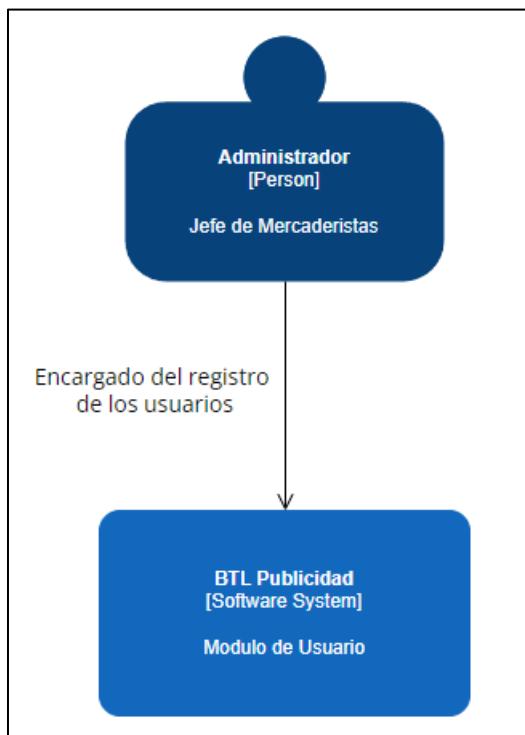
**Excepciones:** 1) Si el nombre de usuario o la contraseña son incorrectos, mostrar un mensaje de error.

### Prototipo / Mockup

The mockup shows a login interface for the 'UbiJob' platform. At the top, there is a header bar with icons for back, forward, search, and refresh, followed by the 'UbiJob' logo. Below the header, the title 'Iniciar Sesión' is centered. There are two input fields: one for 'Correo electrónico' containing '.....' and another for 'Contraseña'. A 'Ingresar' button is positioned below the password field. To the right of the password field is a 'Contactanos' link. The bottom of the screen features a footer bar with a double arrow icon.

Fecha Publicada	Estado	Fecha Terminada	Product Owner
15/09/2024	Terminada	26/09/2024	Torrez Garcia Jorge Luis

## Contexto del Sistema



## Sprint backlog

SPRINT BACKLOG					
Número de Sprint:			Tiempo Programado:		
Objetivo: Análisis, diseño, implementación y pruebas para la HU1.Usuarios					
Fecha de inicio:			Fecha de finalización:		
Id	Tarea	Tipo	Estimación	Responsable	Estado
HU1-1	Integrar librería para generar QR	Desarrollo	1	Torrez Garcia Jorge Luis	Completado
HU1-2	Integrar API Maps para visualizar seguimiento	Desarrollo	1	Torrez Garcia Jorge Luis	Completado
HU1-3	Desplegar Mapa en interfaz web	Diseño	1	machuca yavita miguel angel	Completado
HU1-4	Realizar pruebas finales de integración (de todo lo desarrollado en los sprints previos)	Pruebas	1	Torrez Garcia Jorge Luis	Completado

Jira Tu trabajo Proyectos Filtros Paneles Equipos Planes Aplicaciones Crear Quedan 30 días Buscar Compartir Filtro Grupo Formato Más

Mi proyecto de Kanban  
Proyecto de software

PLANIFICACIÓN Cronograma Tablero Lista Formularios NOVEDAD Objetivos + Añadir vista DESARROLLO Código Páginas del proy... Configuración del pro... Indicadores archivadas...

Proyectos / Mi proyecto de Kanban

### Lista

Buscar en la lista

	Tipo	# Clave	Resumen	Estado	Persona asignada	Etiquetas
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-1	SPRINT 1	FINALIZADA	TORREZ GARCIA JO...	TOR
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-3	Análisis: Elaborar historia de usuario para usuario	TAREAS POR HA...	TORREZ GARCIA JO...	ANÁLISIS
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-4	Diseño: Diseñar interfaz de usuario para usuario	TAREAS POR HA...	TORREZ GARCIA JO...	DISEÑO
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-5	IMPLEMENTACIÓN/DESARROLLO: Implementar la historia de usu...	TAREAS POR HA...	TORREZ GARCIA JO...	IMPLEMENTACIÓN
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-6	Pruebas: Realizar pruebas a la implementación	TAREAS POR HA...	TORREZ GARCIA JO...	PRUEBAS
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-7	Análisis: Elaborar historia de usuario para Gestionar Vended...	TAREAS POR HA...	TORREZ GARCIA JO...	ANÁLISIS
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-8	Diseño: Diseñar interfaz de usuario para Gestionar Vendedor	TAREAS POR HA...	TORREZ GARCIA JO...	
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-9	SPRINT 2	EN CURSO		TOR
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-16	SPRINT 3	TAREAS POR HA...	TORREZ GARCIA JO...	TOR

+ Crear Quickstart

Proyectos / Mi proyecto de Kanban

### Lista

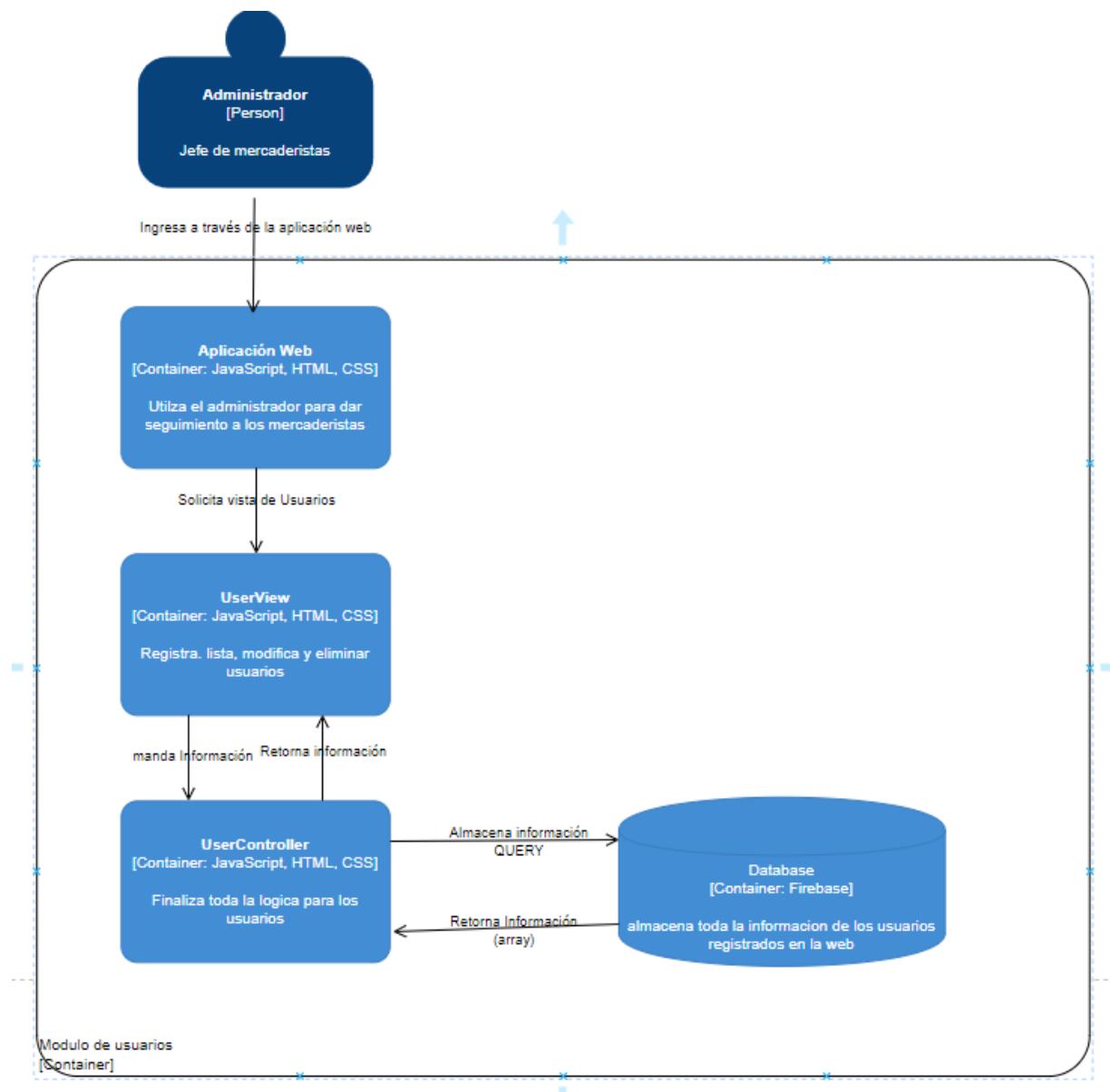
Enviar comentarios

Buscar en la lista

	Tipo	# Clave	Resumen	Estado	Persona asignada	Etiquetas
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-1	SPRINT 1	FINALIZADA	TORREZ GARCIA JO...	TOR
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-9	SPRINT 2	EN CURSO		TOR
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-16	SPRINT 3	TAREAS POR HA...		TOR
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-20	Pruebas: Realizar pruebas finales de integración (de todo lo...	TAREAS POR HA...	TORREZ GARCIA JO...	TOR
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-19	Diseño: Desplegar Mapa en interfaz web	TAREAS POR HA...	TORREZ GARCIA JO...	TOR
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-18	Desarrollo: Integrar API Maps para visualizar seguimiento	TAREAS POR HA...	TORREZ GARCIA JO...	TOR
<input type="checkbox"/>	> <input checked="" type="checkbox"/>	KAN-17	Desarrollo: Integrar librería para generar QR	TAREAS POR HA...	TORREZ GARCIA JO...	TOR

## Diseño

### Diagrama Contendor



## Diseño de datos



### Mapeo

PK

idUsuario

Nombre

Correo

Contraseña

Cargo

## Tabla de volumen

Atributos	Tipo de dato	Amplitud	Llave	Nulo	Descripción
idUsuario	Int	11	PK	No	Identificador de tabla
Nombre	Varchar	100		No	Nombre de usuario
Correo	Varchar	100		No	E-mail de usuario
Contraseña	Varchar	100		No	Contraseña de la cuenta de usuario
Cargo	Varchar	100		No	Cargo del usuario

## *Sprint Retrospective*

REUNIÓN	DETALLE
Review	Se ha concluido exitosamente la pila del Product Backlog asignado para este Sprint
Retrospective	se cumplió con lo planeado para este Sprint con algunas irregularidades de tiempo, pero se cumplió
Tarea	Todas las tareas para este sprint fueron completadas
Tiempo	El tiempo establecido para concluir el Sprint fue aproximadamente según lo planeado
Roles	Los roles fueron llevados a cabo sin problemas

**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y  
TELECOMUNICACIONES



### **3. Manual de Calidad (SQA)**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Integrantes:**

- |                                       |           |
|---------------------------------------|-----------|
| <b>1.</b> Machuca Yavita Miguel Angel | 216063558 |
| <b>2.</b> Torrez Garcia Jorge Luis    | 218137761 |

### **3.1. Introducción**

Una de las fases más importantes en la creación de cualquier proyecto de software es el Aseguramiento de la Calidad del Software (SQA), ya que garantiza que los productos desarrollados cumplan con los requerimientos técnicos establecidos y proporcionen la confianza necesaria en cuanto a su funcionamiento y calidad. El SQA es un proceso sistemático y planificado que engloba una serie de acciones cuyo objetivo principal es asegurar que cada ítem y componente del software se desarrolle bajo los más altos estándares de calidad, minimizando los errores y asegurando la conformidad con los requisitos del cliente.

El aseguramiento de la calidad del software, también denominado aseguramiento del producto de software, implica la verificación de que cada fase del ciclo de vida del proyecto cumple con los estándares y procedimientos previamente definidos. Para ello, se desarrolla un plan de aseguramiento de la calidad del software (SQAP), que contiene todas las directrices que deben seguirse para monitorear la calidad del producto. Dicho plan establece actividades específicas que deberán ser evaluadas y supervisadas, asignando responsabilidades claras al equipo encargado del SQA.

Este documento detalla los planes, roles y actividades que cada miembro de la organización deberá desempeñar para cumplir con los objetivos de calidad establecidos, permitiendo que el software final sea confiable, eficiente y cumpla con las expectativas de los usuarios y clientes.

### **3.2. Antecedente**

La empresa "BTL Publicidad SRL" es una entidad dedicada al ámbito de la publicidad y las campañas, fundada con la visión de crear soluciones innovadoras que impacten de manera efectiva en la audiencia. Desde su creación, ha consolidado un equipo de diseñadores y creativos altamente capacitados, especializados en el desarrollo de elementos visuales, contenido

publicitario y materiales de campaña. La misión de la empresa es proporcionar servicios publicitarios excepcionales, adaptados a las necesidades específicas de cada cliente, utilizando estrategias que optimicen el rendimiento y la visibilidad de las marcas que confían en nosotros.

El equipo de "BTL Publicidad SRL" se caracteriza por realizar un seguimiento minucioso de las campañas, generando informes detallados que permiten ajustar las estrategias y optimizar constantemente los resultados. Además, la organización cuenta con personal especializado en las áreas de diseño, análisis de campañas y gestión de proyectos, garantizando un enfoque estratégico y colaborativo en cada uno de los servicios que ofrece.

Con una visión orientada a ser líderes en el mercado publicitario a nivel nacional, "BTL Publicidad SRL" está comprometida con la innovación, la integridad y la satisfacción de sus clientes, trabajando de la mano con ellos para lograr campañas publicitarias efectivas que generen resultados tangibles y medibles.

### **3.3. Objetivos**

#### ***3.3.1. objetivo general***

Elaborar un plan de aseguramiento de la calidad del software (SQA) para la empresa "BTL Publicidad SRL" que garantice la eficiencia y efectividad en el desarrollo y ejecución de sus sistemas de gestión y control de campañas publicitarias.

#### ***3.3.2. objetivos específicos***

Se tienen los siguientes objetivos:

- Definir los estándares de calidad que deberán ser verificados en cada fase del ciclo de vida del software utilizado para la gestión de campañas publicitarias.
- Establecer los roles y responsabilidades del equipo encargado del SQA, incluyendo a los diseñadores, creativos y personal técnico involucrado en el desarrollo y mantenimiento del software.

- Determinar las etapas del ciclo de vida del software que serán abarcadas por el plan SQA, asegurando que todas las áreas de trabajo estén debidamente contempladas y monitoreadas.
- Describir las tareas específicas de aseguramiento de calidad que se llevarán a cabo para garantizar que el software cumpla con los requisitos técnicos y funcionales.
- Especificar los documentos clave que serán utilizados en el desarrollo y la implementación del plan, así como los reportes y registros necesarios para asegurar la trazabilidad y el control de calidad del proceso.

### **3.3.3. *Misión***

Estamos comprometidos con la creación de productos de software de la más alta calidad, empleando estándares reconocidos y metodologías avanzadas para asegurar que nuestras soluciones satisfagan plenamente las expectativas de nuestros clientes. Nuestro objetivo es ofrecer productos que no solo cumplan con los requerimientos técnicos, sino que también impulsen la eficiencia y el éxito de las empresas que confían en nosotros. A través del uso de tecnologías de vanguardia y un enfoque riguroso en el control de calidad, nos esforzamos por ser el socio estratégico preferido para el desarrollo de software, contribuyendo al crecimiento de sectores clave en el país.

### **3.3.4. *Visión***

Nuestra visión es ser una empresa líder en el desarrollo de software, reconocida a nivel nacional e internacional por la excelencia de nuestros productos y servicios. Aspiramos a ofrecer soluciones de software personalizadas, especialmente en el ámbito de los sistemas ERP, con una gestión eficiente y un enfoque en la mejora continua. Buscamos que nuestros productos sean sinónimo de innovación, confiabilidad y satisfacción del cliente, estableciéndonos como un referente en el sector tecnológico y consolidándonos como socios de confianza en la transformación digital de nuestros clientes.

### **3.4. Política de calidad**

Asumimos el compromiso de desarrollar nuestras actividades bajo los más altos estándares de calidad, implementando un Sistema de Gestión de la Calidad basado en normas reconocidas internacionalmente, como la ISO 9001:2008, con el objetivo de aumentar nuestra competitividad, cumplir con las expectativas de nuestros clientes y expandir nuestra participación tanto en el mercado nacional como internacional. Para asegurar el cumplimiento de esta política, la dirección se compromete a difundirla, garantizar su comprensión y supervisar su cumplimiento a través de los siguientes lineamientos:

- Proporcionar productos de software con una calidad orientada al cero defecto, asegurando su conformidad con los requerimientos establecidos.
- Cumplir estrictamente con los plazos de entrega acordados con nuestros clientes, respetando siempre los tiempos y requerimientos del proyecto.
- Mejorar de forma continua nuestros procesos y el sistema de gestión de calidad, implementando soluciones innovadoras y eficientes.
- Capacitar permanentemente a nuestro personal según sus necesidades y las nuevas tecnologías, promoviendo su integración y desarrollo profesional.
- Interpretar correctamente los requerimientos y expectativas de nuestros clientes, generando soluciones que no solo los satisfagan, sino que aporten valor adicional.

### **3.5. SLOGAN**

“La satisfacción de nuestros clientes es nuestra mayor prioridad”.

### **3.6. Plan de Aseguramiento de Calidad de Software (SQAP)**

#### **Propósito**

El propósito de este plan es especificar cómo se va asegurar la calidad del software que va a ser implementado durante el proceso. El objetivo del Aseguramiento de la Calidad (Software Quality Assurance) es entregar a la administración una vista adecuada del proceso utilizado y los productos construidos mediante acciones planificadas y sistemáticas que aseguren la calidad de dichos procesos y productos.

Este plan describe las actividades a realizar por el SQA y define un conjunto de estándares a seguir para lograr el objetivo descrito anteriormente. Cabe aclarar que, según el modelo de calidad a utilizar en el curso, el criterio decisivo para determinar la calidad de un producto de software es la calidad desde el punto de vista del usuario, utilizado en un ambiente y en un contexto de uso determinados: “calidad en el uso”.

Este Plan solo cubre la parte del ciclo de vida correspondiente al desarrollo del software, pero no cubre la parte del ciclo de vida correspondiente al mantenimiento. En este caso particular el software a desarrollar consiste en un sistema web para el control o seguimiento de obras en construcción. Se desarrollará en PHP.

#### **Descripción**

La calidad del software se entiende como el cumplimiento de los requisitos explícitamente definidos y documentados, la alineación con los estándares de desarrollo aplicables y la incorporación de requisitos implícitos que todo producto de calidad debe cumplir. A través de la implementación del Plan de Aseguramiento de la Calidad de Software (SQAP), se busca cumplir con los siguientes elementos clave de calidad del software:

- Correctitud: Asegurarse de que el software funcione conforme a los requerimientos definidos.
- Eficiencia: Optimizar el uso de recursos y ofrecer un rendimiento adecuado.
- Fiabilidad: Garantizar que el software funcione de manera consistente bajo diversas condiciones.
- Facilidad de uso: Proporcionar una experiencia de usuario intuitiva y accesible.
- Facilidad de mantenimiento: Hacer que el software sea fácilmente modificable y actualizable.
- Seguridad e integridad: Proteger los datos y garantizar la estabilidad del sistema frente a posibles amenazas o fallos.
- Portabilidad: Asegurar que el software pueda ser trasladado y utilizado en distintos entornos y plataformas.

Este plan se aplicará en todos los proyectos de software desarrollados por Jair Soft con el fin de garantizar que cada producto cumpla con los más altos estándares de calidad y satisfacción del cliente.

### **3.7. Estándares**

Para alcanzar la calidad total de los productos y la mejora continua, se utilizan los siguientes estándares:

- ❖ **IEEE STD-730:** "El aseguramiento de la calidad de software es un modelo planificado y sistemático de todas las acciones necesarias a fin de asegurar que el producto cumpla con los requerimientos técnicos establecidos".
- ❖ **IEEE STD-729:** Análisis de requerimientos de software.
- ❖ **IEEE STD-828:** Estándar para planes del manejo de las configuraciones de software.
- ❖ **IEEE STD-829:** Estándar para documentación de pruebas de software.
- ❖ **IEEE STD-830:** Estándar para especificación de requerimientos de software.
- ❖ **IEEE STD-1012:** Estándar para la planificación de verificación y validación de software.

- ❖ **IEEE STD-1063:** Estándar para los manuales de usuarios de software.

## Valores

- Responsabilidad
  - Cumplir a tiempo con los compromisos adquiridos con nuestros clientes.
- Calidad
  - Hacer las cosas bien y a la primera para entregar a nuestros clientes productos o servicios bajo los mejores terminos de calidad, eficiencia y costo.
- Nuestros clientes
  - Nuestra razón de ser.
- Nuestros Colaboradores
  - Es la fortaleza de nuestra empresa para lograr nuestra misión.

## Organigrama

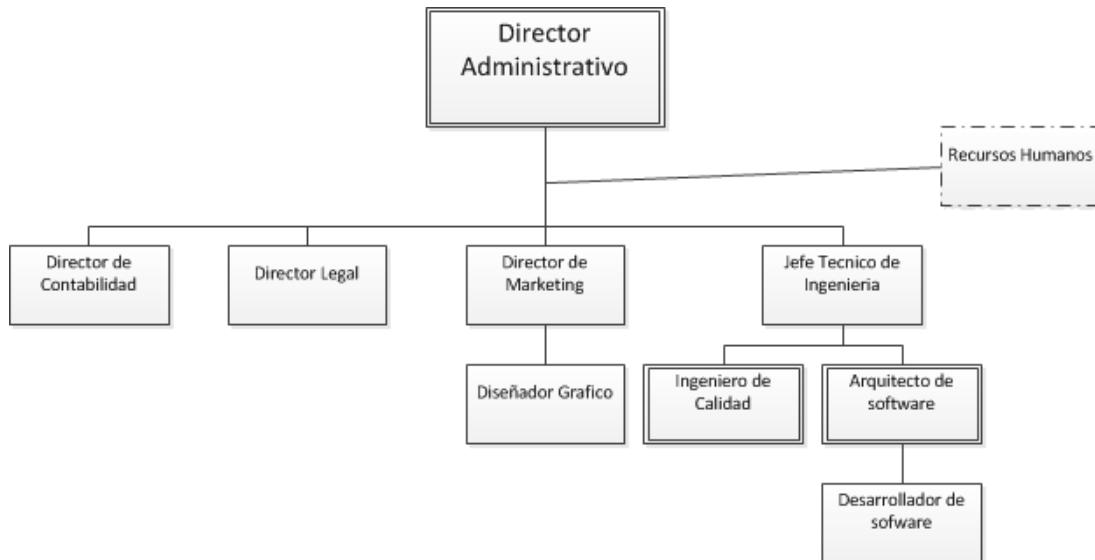


Ilustración 7 Organigrama

## 3.8. Gestión

### 3.8.1. Organización

Las líneas de trabajo dentro de la organización que están más relacionadas con la calidad del software son: Verificación y Gestión de Proyecto.

### **3.8.2. Tareas**

La relación de tareas asociadas con el ciclo de vida de desarrollo de software y las actividades de la SQA son las siguientes, las cuales se ejecutarán durante el desarrollo del producto de software:

CICLO DE VIDA DEL SOFTWARE (PUDS)	TAREAS Y ACTIVIDADES ASOCIADAS AL CICLO DE VIDA DEL SOFTWARE
Requerimientos	<b>Análisis de requisitos de software</b> <b>Generación de especificaciones</b> <b>Revisión de especificaciones</b> <b>Revisión de las especificaciones de software</b>
Diseño	<b>Diseño preliminar</b> <b>Generación de especificaciones de diseño preliminar</b> <b>Revisión del diseño preliminar</b> <b>Diseño detallado</b> <b>Generación de especificaciones de diseño detallado</b> <b>Revisión del diseño detallado</b> <b>Revisión del diseño preliminar</b> <b>Revisión del diseño detallado del software</b> <b>(Ambos versus las especificaciones)</b>
Implementación	<b>1. Codificación</b> <b>2. Generación de código</b> <b>3. Revisión de código</b> <b>Revisión de código versus la documentación generada</b>

<b>Pruebas</b>	<b>Elaboración de pruebas de unidad y generación de resultados</b> <b>Revisión de resultados</b> <b>Elaboración de las pruebas de unidad y de integración del software</b> <b>Revisión de los resultados de las pruebas</b> <b>Revisión de las pruebas funcionales y evaluación de los resultados</b>
<b>Instalación y Prueba Final</b>	<b>Instalación del producto software</b> <b>Prueba final bajo ambiente real</b> <b>Generación de resultados de prueba</b> <b>Revisión de resultados</b> <b>Revisión de la instalación del software y evaluación de Resultados</b>

Tabla 24 Relación de Tareas Asociadas

Las actividades de SQA definidas en el modelo de proceso son:

### 3.8.3. Actividades

#### Ciclo de vida del software cubierto por el Plan

Las etapas más importantes del ciclo de vida del software que cubre el Plan son la etapa del relevamiento de requerimientos y el principio de la etapa de diseño dado una buena especificación de requerimientos y un diseño adecuado constituyen una base sólida para el proyecto; y errores detectados en forma tardía de estas etapas son muy costosos e incluso podrían hacer fracasar al proyecto.

Los productos de proyecto que tendrán revisiones de calidad son todos los entregables que

requiere el modelo de proceso seguido en el proyecto.

Se hace especial énfasis en los entregables que incluyen:

- Especificación de Requerimientos
- Descripción de la Arquitectura y Alcance del Software
- Plan de Proyecto

### **Actividades de calidad a realizarse**

Las tareas a ser llevadas a cabo deberán reflejar las evaluaciones a realizar, los estándares a seguir, los productos a revisar, los procedimientos a seguir en la elaboración de los distintos productos y los procedimientos para informar de los defectos detectados a sus responsables y realizar el seguimiento de los mismos hasta su corrección.

Las actividades que se realizarán son:

- Revisar cada producto
- Revisar el ajuste al proceso
- Realizar Revisión Técnica Formal (RTF)
- Asegurar que las desviaciones son documentadas

### **Revisar cada producto**

En esta actividad se revisan los productos que se definieron como claves para verificar en el Plan de calidad.

Se debe verificar que no queden correcciones sin resolver en los informes de revisión previos, si se encuentra alguna no resuelta, debe ser incluida en la siguiente revisión. Se revisan los productos contra los estándares, utilizando la checklist definida para el producto.

Se debe identificar, documentar y seguir la pista a las desviaciones encontradas y verificar que se hayan realizado las correcciones.

Como salida se obtiene el Informe de revisión de SQA, este informe debe ser distribuido a los responsables del producto y se debe asegurar de que son conscientes de desviaciones o discrepancias encontradas.

### **Revisar el ajuste al proceso**

En esta actividad se revisan los productos que se definieron como claves para verificar el cumplimiento de las actividades definidas en el proceso. Con el fin de asegurar la calidad en el producto final del desarrollo, se deben llevar a cabo revisiones sobre los productos durante todo el ciclo de vida del software.

Se debe recoger la información necesaria de cada producto, buscando hacia atrás los productos previos que deberían haberse generado, para poder establecer los criterios de revisión y evaluar si el producto cumple con las especificaciones.

Esta información se obtiene de los siguientes documentos:

Plan del Proyecto, Plan de la iteración, Plan de Verificación. Antes de comenzar, se debe verificar en los informes de revisión previos que todas las desviaciones fueron corregidas, si no es así, las faltantes se incluyen para ser evaluadas.

Como salida se obtiene el Informe de revisión de SQA correspondiente a la evaluación de ajuste al Proceso, este informe debe ser distribuido a los responsables de las actividades y se debe asegurar de que son conscientes de desviaciones o discrepancias encontradas.

### **Realizar Revisión Técnica Formal (RTF)**

El objetivo de la RTF es descubrir errores en la función, la lógica o la implementación de cualquier producto del software, verificar que satisface sus especificaciones, que se ajusta a los estándares establecidos, señalando las posibles desviaciones detectadas. Es un proceso de revisión riguroso, su objetivo es llegar a detectar lo antes posible, los posibles defectos o desviaciones en

los productos que se van generando a lo largo del desarrollo. Por esta característica se adopta esta práctica para productos que son de especial importancia.

En la reunión participan el responsable de SQA e integrantes del equipo de desarrollo.

Se debe convocar a la reunión formalmente a los involucrados, informar del material que ellos deben preparar por adelantado, llevar una lista de preguntas y dudas que surgen del estudio del producto a ser revisado.

La duración de la reunión no debe ser mayor a dos horas.

Como salida se obtiene el Informe de RTF.

### **Asegurar que las desviaciones son documentadas**

Las desviaciones encontradas en las actividades y en los productos deben ser documentadas y ser manejadas de acuerdo a un procedimiento establecido.

Se debe chequear que los responsables de cada plan los modifiquen cada vez que sea necesario, basados en las desviaciones encontradas.

## **Relaciones entre las actividades de SQA y la planificación**

Actividad	Semana cuando se realiza
Planificar la Calidad	Semana 2
Evaluar y Ajustar el Plan de SQA	Semanas 3 y 4
Revisión Técnica Formal (RTF)	Semanas 5, 6, 7, 8, 10, 11 y 12
Revisar las entregas	Todas las semanas
Revisar el Ajuste al Proceso	Semanas 3 a 12 (inclusive)
Evaluar la calidad de los productos	Semanas 3 a 12 (inclusive)
Realizar el informe final de SQA	Semana 14
Describir la Versión	Semanas 5, 7, 9, 11 y 12
Escribir las notas de la versión	Semanas 6, 8, 10 y 12

*Tabla 25 Relaciones entre las actividades de SQA y la planificación*

### **3.8.4. Responsables**

#### **Responsabilidades del grupo de Desarrollo**

Esta organización es responsable de:

- ❖ Desarrollar un producto de software en base a lo definido en el SQAP y los contratos establecidos con el cliente.
- ❖ Generar la debida documentación definida en la SQAP acerca de cada una de sus actividades con el fin de llevar un control de las mismas.
- ❖ Entregar la documentación de desarrollo que se exige en el plan (SQAP).

#### **Responsabilidades del Cliente**

Esta organización es responsable de:

- ❖ Proveer la información necesaria para el desarrollo del software con el fin de satisfacer sus necesidades.
- ❖ Brindar los recursos y condiciones necesarias para elaborar el software.
- ❖ Participar activamente en la organización del SQA para obtener óptimos resultados.

## **Responsabilidades de la SQA**

Actividades del grupo SQA:

- ❖ Establecimiento del plan SQA para el proyecto.
- ❖ Participar en el desarrollo de la descripción del proceso de software.
- ❖ Revisión de las actividades de ingeniería del software para verificar su ajuste al proceso del software.
- ❖ Auditoria de los productos de software designados para verificar el ajuste con los definidos como parte del proceso de software.
- ❖ Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.
- ❖ Registrar lo que no se ajuste a los requisitos e informar a sus superiores.
- ❖ Coordinar el control y la gestión de cambios.
- ❖ Analizar las métricas del software.

## **Esta organización es responsable de:**

Garantizar la calidad del producto de software desarrollado.

- ❖ Implantar normas y actividades para el desarrollo del software.
- ❖ Realizar reuniones para resolver los posibles conflictos durante el desarrollo del software.
- ❖ Aprobar y publicar el SQAP.
- ❖ Observar las deficiencias en el SQAP.
- ❖ Mejorar el SQAP, recomendando modificaciones o correcciones con el fin de obtener resultados óptimos.

- ❖ Autorizar la implantación del software.
- ❖ Enfoque de gestión de calidad.
- ❖ Tecnologías (métodos y herramientas).
- ❖ Revisiones Técnicas Formales.
- ❖ Estrategia de pruebas.
- ❖ Control de la documentación y de cambios.
- ❖ Procedimientos que aseguren ajustes a los estándares.
- ❖ Mecanismos de medición y generación de informes.

Como parte de las actividades del responsable de SQA se revisarán los productos que se consideren relevantes para la calidad del producto y del proceso.

A continuación, se identifican esos productos y el responsable de las acciones correctivas para eliminar los defectos de cada producto.

<b>Producto</b>	<b>Responsable</b>
Documento de Requerimientos	Jefe de técnico de Ing.
Modelo de Casos de Uso	Arquitecto de software
Alcance del Sistema	Jefe técnico del sistema
Descripción de la Arquitectura	Arquitecto de Software
Modelo de Diseño	Arquitecto de Software
Modelo de Datos	Arquitecto de Software
Estándar de Implementación	Ing. De Calidad
Estándar de documentación técnica	Ing. De Calidad
Documento de Estimaciones	Ing. De Calidad
Documento de Riesgos	Jefe Técnico de Ing

Plan del Proyecto	Jefe Técnico de Ing.
Plan de Verificación y Validación	Ing. De Calidad
Reporte de pruebas unitarias, de integración y del Sistema	Jefe Técnico de Ing.
Plan de Implementación	Jefe Técnico de Ing.
Estándar de Documentación de Usuario	Ing. De Calidad
Documentación de Usuario	Ing. De Calidad
Plan de Gestión de Configuración	Jefe Técnico de Ing.

Tabla 26 Productos y Responsables de Acciones Correctivas

### 3.9. Evolución del SQAP

La organización responsable del SQAP, es la organización SQA. El SQAP, será dado a conocer a todas las organizaciones, cualquier modificación durante la implementación del SQAP debido a circunstancias o necesidades, se ejecutará a través del siguiente plan para la implantación y evolución del SQAP.

#### 3.9.1. *Control de cambio*

El control de cambio del SQAP, será realizado utilizando versiones evolutivas, así por ejemplo al presente SQAP, se denomina como la versión 1.0, los siguientes puntos describen el criterio a seguir para la asignación de versión a las modificaciones del SQAP.

1. Si los cambios solo afectan a secciones específicas del SQAP y las mismas no modifican la línea base del SQAP, la versión asignada será la suma de más 0.1, a la versión actual, los cambios se realizarán insertando y /o eliminando hojas en el SQAP vigente.

2. Si los cambios son significativos la versión asignada será la suma de cualquier valor entre [0.1,0.2,...,9.8,9.9,1] para alcanzar el próximo número entero, a la versión actual del SQAP y se imprimirá completamente el nuevo SQAP.
3. Si se repite con frecuencia el punto 1, y no se imprimió completamente el SQAP modificado ya varias veces, entonces para contar con un SQAP consistente y mantenible se debe imprimir completamente el Plan, contadas las modificaciones realizadas previamente, para la asignación de versión se utilizará el criterio descrito en el punto 2.

### ***3.9.2. Cuando hacer modificaciones***

Las modificaciones del SQAP serán realizadas cuando se den las siguientes situaciones:

- Surjan modificaciones en el Plan
- Cambios en el modelo de desarrollo como ser requerimientos, diseño, etc.
- Incumplimiento de alguna sección del SQAP

### ***3.9.3. Implantación de modificaciones***

Las modificaciones del SQAP, se harán en función al problema presentado de la siguiente manera:

- Detectar uno de los puntos especificados en 3.4.2.
- Realizar un análisis del problema.
- Realizar reuniones técnicas con las áreas involucradas en la modificación del SQAP, estas áreas además de exponer los problemas, darán recomendaciones para la modificación correspondiente.
- Hacer la modificación.
- Implementar la modificación correspondiente
  - Una vez finalizada la implementación se evaluará si fue correcta la modificación.

- Promulgar la nueva versión del SQAP.

Las modificaciones al SQAP, solo ser realizará en las secciones del SQAP que sean afectadas por la observación, este trabajo estará a cargo de la organización de la SQA.

#### **3.9.4. *Publicación del SQAP***

La publicación de las versiones del SQAP las hará el consultor a todos los miembros del SQAP si se presenta el punto 3.4.1 se imprimirá todo el documento SQAP actualizado y publicado a través de los representantes de la organización.

### **3.10. Documentación**

#### **Propósito**

Identificación de la documentación relativa a desarrollo, Verificación & Validación, uso y mantenimiento del software.

Establecer como los documentos van a ser revisados para chequear consistencia: se confirman criterio e identificación de las revisiones.

#### **Documentación mínima requerida**

La documentación mínima es la requerida para asegurar que la implementación logrará satisfacer los requerimientos.

#### **Especificación de requerimientos del software**

El documento de especificación de requerimientos deberá describir, de forma clara y precisa, cada uno de los requerimientos esenciales del software además de las interfaces externas.

El cliente deberá obtener como resultado del proyecto una especificación adecuada a sus necesidades en el área de alcance del proyecto, de acuerdo al compromiso inicial del trabajo y a los cambios que este haya sufrido a lo largo del proyecto, que cubra aquellos aspectos que se haya acordado detallar con el cliente.

La especificación debe:

- Ser completa:
  - Externa, respecto al alcance acordado.
  - Internamente, no deben existir elementos sin especificar.
- Ser consistente, no puede haber elementos contradictorios.
- Ser no ambigua, todo término referido al área de aplicación debe estar definido en un glosario.
- Ser verificable, debe ser posible verificar siguiendo un método definido, si el producto final cumple o no con cada requerimiento.
- Estar acompañada de un detalle de los procedimientos adecuados para verificar si el producto cumple o no con los requerimientos.
- Incluir requerimientos de calidad del producto a construir.
- Los requerimientos de calidad del producto a construir son considerados dentro de atributos específicos del software que tienen incidencia sobre la calidad en el uso' y se detallan a continuación:

### **Funcionalidad**

- a) adecuación a las necesidades
- b) precisión de los resultados
- c) interoperabilidad
- d) seguridad de los datos

### **Confiabilidad**

- a) madurez
- b) tolerancia a faltas

c) recuperabilidad (Ver si aplica)

### **Usabilidad**

- a) comprensible
- b) aprendible
- c) operable
- d) atractivo

### **Eficiencia**

- a) comportamiento respecto al tiempo (Ver si aplica)
- b) utilización de recursos

### **Mantenibilidad**

- a) analizable
- b) modificable
- c) estable, no se producen efectos inesperados luego de modificaciones
- d) verificable

### **Portabilidad**

- a) adaptable (Ver si aplica)
- b) instalable
- c) co-existencia
- d) reemplazante (Ver si aplica)

Cada uno de estos atributos debe cumplir con las normas y regulaciones aplicables a cada uno.

### **Descripción del diseño del software**

El documento de diseño especifica como el software será construido para satisfacer los requerimientos.

Deberá describir los componentes y subcomponentes del diseño del software, incluyendo interfaces internas. Este documento deberá ser elaborado primero como Preliminar y luego será gradualmente extendido hasta llegar a obtener el Detallado.

El cliente deberá obtener como resultado del proyecto el diseño de un producto de software que cubra aquellos aspectos que se haya acordado con el cliente incorporar al diseño, en función de la importancia que estos presenten y de sus conexiones lógicas.

El diseño debe:

- Corresponder a los requerimientos a incorporar:
  - Todo elemento del diseño debe contribuir a algún requerimiento
  - La implementación de todo requerimiento a incorporar debe estar contemplada en por lo menos un elemento del diseño.
- Ser consistente con la calidad del producto

### **3.11. Plan de Verificación y Validación**

La generación y documentación de la descripción del Plan de Verificación y Validación (SWP) es la siguiente:

#### **MODELO A USAR PARA EL CONTENIDO DEL SWP**

1. OBJETIVO
  2. ALCANCE
  3. DEFINICIONES, ACRONIMOS Y ABREVIACIONES
  4. ORGANIZACIÓN RESPONSABLES
  5. CICLO DE VIDA DE VERIFICACION Y VALIDACION
- APENDICE
- INDICE

La organización responsable por las tareas de verificación y validación del software es la organización de SQA comandada por la organización del consultor, la cual interactúa con la

organización de desarrollo para alcanzar los objetivos del plan. En casos necesarios de conflictos extremos entre el consultor y el desarrollador se recurrirá al cliente.

### **3.11.1. Ciclo de Vida de Verificación y Validación**

El plan se basa en el siguiente ciclo de vida del software:

- ✖ Fase de concepto
- ✖ Fase de requerimientos
- ✖ Fase de diseño
- ✖ Fase de implementación
- ✖ Fase de prueba
- ✖ Fase de instalación y prueba
- ✖ Fase de operación y mantenimiento

<b>1 FASE DE CONCEPTO V&amp;V</b>		
<b>TAREAS MÍNIMAS DE V&amp;V</b>	<b>ENTRADAS REQUERIDAS</b>	<b>SALIDAS REQUERIDAS</b>
Evaluación de concepto de documentación Evaluar el concepto de documentación para determinar si el concepto propuesto satisface las necesidades del usuario y el objetivo del proyecto. Identificar las restricciones principales de interfaces del sistema y limitaciones de objetivos propuestos. Fijar critibilidad de cada elemento de software	Concepto de documentación: Planificación del proyecto, Memo de iniciación del proyecto	Reporte de tareas Reporte de anomalías

Tabla 27 Tareas, entradas y salidas de V&V (Fase 1)

2 FASE DE DISEÑO V&V		
TAREAS MÍNIMAS DE V&V	ENTRADAS REQUERIDAS	SALIDAS REQUERIDAS
<p><b>Análisis de seguimiento a los requerimientos de software.</b></p> <p>Seguir los requerimientos SRS hacia los requerimientos del sistema dentro del concepto de documentación. Analizar identificando relaciones para correcciones, consistencia, completitud y optimización.</p>	Concepto de documentación SRS. Documentación de requerimientos de interfaces.	Reporte de tareas. Reporte de anomalías.
<p><b>Evaluación de requerimientos de software.</b></p> <p>Evaluar los requerimientos SRS para: precisión, cumplimiento, confiabilidad, y capacidad de ser probado. Fijar la critibilidad de requerimientos para identificar rendimiento o área críticas de software.</p>	Concepto de documentación SRS. Documentación de requerimientos de interfaces.	Reporte de tareas. Reporte de anomalías.
	Concepto de documentación SRS. Documentación de requerimientos de interfaces.	Reporte de tareas. Reporte de anomalías.
	Concepto de documentación SRS. Documentación de requerimientos de interfaces.	Plan de pruebas del sistema. Reporte de anomalías.

	Documentación de usuario.	
	Concepto de documentación SRS. Documentación de requerimientos de interfaces. Documentación de usuario.	Plan de pruebas de aceptación. Reporte de anomalías.

Tabla 28 Tareas, entradas y salidas de V&V (Fase 2 )

3 Fase de Implementación V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas

Tabla 29 Tareas, entradas y salidas de V&V (Fase 3)

### 3.11.2. Informe de Verificación y Validación

El formato para la documentación de los resultados de la implementación del plan de verificación y validación del software (SWR) es la siguiente:

*Reporte sumario de fase V&V*

<b><u>REPORTE SUMARIO DE FASE V&amp;V</u></b>		Pág.
FASE: .....		
# De Reporte: Lugar:	Fecha: / / Hora:	.....
<p>a) Descripción de las tareas de V&amp;V realizadas:</p> <p>.....</p> <p>.....</p>		
<p>b) Sumario de resultados de tareas:</p> <p>.....</p> <p>.....</p>		
<p>c) Sumario de anomalías y resolución:</p> <p>.....</p> <p>.....</p>		
<p>d) Evaluación de calidad del software:</p> <p>.....</p> <p>.....</p>		
<p>e) Recomendaciones:</p> <p>.....</p> <p>.....</p>		
<b>Equipo de Trabajo:</b> <b>Nombre:</b> ..... .....		<b>Firma</b>

## *Reporte de Anomalías*

<b><u>REPORTE DE ANOMALIAS</u></b>		Pág.   .....
FASE: .....		
# De Reporte: .....	Fecha: .. / .. / ..	Hora: .....
Lugar: .....		
a) Descripción y ubicación: ..... .....		
b) Impacto: ..... .....		
c) Causa: ..... .....		
d) Critibilidad: ..... .....		
e) Recomendaciones: ..... .....		
<b>Equipo de Trabajo:</b> Nombre: ..... ..... ..... .....		
Firma .....		

## *Reporte Final de V&V*

<b>REPORTE FINAL DE V&amp;V</b>		Pág.		
FASE: .....				
# De Reporte: .....	Fecha: .. / .. / .. Hora: .....			
Lugar: .....				
<p>a) Resumen de todas las tareas V&amp;V, durante el ciclo de vida del software: ..... .....</p> <p>b) Resumen de resultados de tareas: ..... .....</p> <p>c) Resumen de anomalías, resolución: ..... .....</p> <p>d) Evaluación total de la calidad del software: ..... .....</p> <p>e) Recomendaciones: ..... .....</p>				
<table border="1"><tr><td><b>Equipo de Trabajo:</b> <b>Nombre:</b> ..... ..... .....</td><td><b>Firma</b></td></tr></table>			<b>Equipo de Trabajo:</b> <b>Nombre:</b> ..... ..... .....	<b>Firma</b>
<b>Equipo de Trabajo:</b> <b>Nombre:</b> ..... ..... .....	<b>Firma</b>			

### **3.12. Documentación del Usuario (UD)**

Esta descripción de documentación de usuario (UD) se basa en el estándar ANSI / IEEE – Std 1036 “STANDARD FOR SOFTWARE USER DOCUMENTATION”.

La información especificada debe ser incluída en la documentación del usuario, esta documentación de usuario comprenderá de un conjunto. En cada documento se debe tomar en cuenta y describir los siguientes puntos.

Los documentos de usuario serán presentados en dos modos: instruccional y de referencia.

Los usuarios del software utilizarán los documentos ya sea para aprender acerca del software (modo instruccional) o para refrescar su memoria acerca del software (modo de referencia).

#### ***3.12.1. Modo Instruccional***

Un modo instruccional de documento debe:

- ✖ Proveer el ambiente y la información necesaria para entender el sistema.
- ✖ Proveer la información necesaria para aprender lo que puede hacer con el software y como lo puede usar.
- ✖ Proveer ejemplos para reforzar el proceso de aprendizaje.

#### ***3.12.2. Modo de Referencia***

Un documento de modo de referencia debe:

- ✖ Organizar y proveer información necesaria.
- ✖ Facilitar accesos aleatorios a la información.

Los documentos de modo de referencia que debe ser incluidos son:

- a) Manual de comandos.
- b) Manual de mensajes de error.
- c) Manual de llamadas de programas.
- d) Guía de referencia rápida.
- e) Manual de Herramientas del software.
- f) Manual de utilitarios.

## **MODELO A USAR PARA EL CONTENIDO DEL UD**

1. TITULO DE LA PAGINA
2. RESTRICCIONES
3. GARANTIAS Y OBLIGACIONES CONTRACTUALES
4. TABLA DE CONTENIDO
5. LISTA DE ILUSTRACIONES
6. INTRODUCCION
  - Descripción de audiencia
  - Declaración de aplicación
  - Declaración de objetivos
  - 6.4 Descripción del uso de documentos
  - 6.5 Documentos relacionados
  - 6.6 Convenciones
    - 6.6.1 Símbolos
    - 6.6.2 Convenciones de estilo
    - 6.6.3 Convenciones de sintaxis de comandos
  - 6.7 Instrucciones de reportes de problemas
7. CUERPO DEL DOCUMENTO
  - 7.1. Cuerpo del documento en modo instruccional
    - 7.1.1 Alcance
    - 7.1.2 Materiales
    - 7.1.3 Preparaciones
    - 7.1.4 Precauciones y prevenciones
    - 7.1.5 Métodos
    - 7.1.6 Información relacionada
  - 7.2. Cuerpo del documento en modo de referencia
    - 7.2.1 Objetivo
    - 7.2.2 Materiales
    - 7.2.3 Preparaciones
    - 7.2.4 Entradas
    - 7.2.5 Precauciones y prevenciones

- 7.2.6 Invocación
  - 7.2.7 Operaciones de suspensión
  - 7.2.8 Operaciones de terminación
  - 7.2.9 Salidas
  - 7.2.10 Condiciones de error
  - 7.2.11 Información relacionada
8. MENSAJES DE ERROR, CONOCIMIENTO DE PROBLEMAS, RECUPERACION DE ERROR
  9. ANEXOS
  10. BIBLIOGRAFIA
  11. GLOSARIO
  12. INDICE

### **3.13. Estándares, prácticas y convenciones**

#### ***3.13.1. Estándar de Codificación***

La codificación del software se realizará con el lenguaje C# de Visual Studio.

Las normas de codificación se definen de la siguiente forma:

- ☒ El software debe ser subdividido en módulos independientes, de acuerdo al diseño establecido.
- ☒ La documentación de un programa debe tener el siguiente formato:

- Nombre del programa

    Objetivo

- Nombre de las entradas:

    Base de Datos

    Archivos

    Registros

    Formatos de pantalla

- Nombre de las salidas:

- Base de Datos
- Archivos
- Registros
- Formatos de pantalla
- Reportes
- Nombre de los archivos de actualización:
  - Base de Datos
  - Archivos
  - Registros
- Nombre del autor
  - Fecha de creación
- Historial de actualizaciones
  - Versión
  - Fecha de cambio
  - Objetivo de cambio
- ✖ Cada módulo debe explicar sus funciones
- ✖ La declaración de cualquier variable debe estar comentada, explicando su función.
- ✖ Debe existir una sola instrucción por cada línea de código.
- ✖ Cada función debe de estar debidamente documentada, explicar la funcionalidad, la función de cada parámetro.
- ✖ Cada mensaje de error o excepciones deben de indicar el lugar donde se originó y la función o procedimiento en el cual se produjo.
- ✖ Para asignar nombres a las variables debe de realizarse de la siguiente forma:
  - xNombreCliente**
  - xSaldo**
  - xFechaInicio**

Donde **x** = indica el tipo de dato, puede representar: enteros, reales, cadenas, etc.

- ✖ Los nombres de las funciones deben de indicar su funcionalidad.
- ✖ Cada clase implementada debe de estar comentada de la siguiente forma:

- Nombre
- Fecha y hora de creación
- Autor
- Nombre del módulo al que pertenece
- Funcionalidad
- ☒ Se usaran las Convenciones de Codificación en Java de **Sun Microsystems, Inc**

### ***3.13.2. Estándar de Comentarios***

- ☒ Un comentario debe explicar porque se realiza alguna acción.
- ☒ Los comentarios dentro de un módulo deben estar separados del código.
- ☒ Utilizar comentarios de más de una línea para realizar descripciones, y comentarios de una línea para realizar especificaciones.

#### **Estándares para documentación**

Entregable	Estándar

### ***3.13.3. Responsables de verificar el cumplimiento***

Los responsables de realizar la verificación del cumplimiento con los estándares definidos son:

- ☒ El jefe del equipo de desarrollo.
- ☒ La organización del SQA.

## **3.14. Revisiones y auditorias**

Los responsables de estas revisiones es la organización del SQA, con la participación de todo elemento de la organización que tengan que ver con los requerimientos, tales como: los diseñadores del software, agentes de pruebas.

Las revisiones y auditorias de los resultados del desarrollo se realizan a medida que se terminan

cada una de las fases del ciclo de vida de desarrollo de software, con el fin de:

- ✖ Conocer el progreso alcanzado en el desarrollo.
- ✖ Evaluar el ajuste a los requerimientos del sistema.
- ✖ Evaluar la eficiencia en el trabajo.

Se deben llevar a cabo, al menos, las siguientes revisiones y auditorias:

#### ***3.14.1. Revisión de los Requerimientos de Software (SRR):***

La SRR se genera para:

- ✖ Evaluar las especificaciones de requerimientos del software (SRS).
- ✖ Asegurar que los requerimientos establecidos en la SRS, sean los correctos y estén completos.
- ✖ Garantizar la calidad, viabilidad e integridad de los requerimientos establecidos.

Los requerimientos de revisiones de SRS en la SRR son los siguientes:

- a)** Fiable
- b)** Completo
- c)** Depurable
- d)** Modificable
- e)** Consistente
- f)** Libre de ambigüedades
- g)** Utilizable durante la fase de operación y mantenimiento.
- h)** Inspeccionar que la relación entre los requerimientos y sus derivados sea la adecuada.

#### ***3.14.2. Revisión del Diseño Preliminar (PDR):***

La PDR es realizada para evaluar la suficiencia técnica del SDD preliminar, antes de comenzar con el diseño detallado, define los siguientes puntos:

- ✖ Evaluar el progreso, consistencia y suficiencia técnica del alcance de diseño con los requerimientos funcionales de la SRS.

- ✖ Verificar la existencia y compatibilidad de las interfaces entre el software, el hardware y los usuarios finales.
- ✖ Determinar un diseño de software que cumpla con los requerimientos.

Para la PDR se toman como requerimientos de revisión los siguientes puntos:

- ✖ Revisar que se detallen todas las interfaces con otro software, sistemas de comunicación, etc. Para una adecuada identificación de interfaces y de un diseño óptimo.
- ✖ Revisar que exista un análisis del diseño para verificar la compatibilidad con los requerimientos críticos.
- ✖ Revisar que se establece los requerimientos del factor humano.

#### **3.14.3. Revisión del Diseño Crítico (CDR):**

La CDR es generada para determinar la aceptabilidad de cómo la SDD cumple con la SRS. Evalúa la suficiencia técnica, integridad del diseño detallado del software, antes de comenzar a codificar para establecer que el diseño detallado satisface los requerimientos de la SRS.

Para la CDR se toman como requerimientos de revisión los siguientes puntos:

- ✖ Evaluar la compatibilidad del diseño detallado con la SRS.
- ✖ Examinar la representación de datos en forma de diagramas lógicos, algoritmos, almacenamiento y representación de datos.
- ✖ Determinar la compatibilidad e integridad de requerimientos de interfaces.
- ✖ Establecer que todas las interfaces internas y externas incluyendo interacciones con la base de datos sean expresadas.

#### **3.14.4. Revisión del Plan de Verificación y Validación (SVVPR):**

La SVVPR es generado para la evaluación de:

- ✖ Los métodos de Verificación y Validación definidos en el SVVP.
- ✖ El cumplimiento durante el desarrollo del software con el SVVP.

Se realizan revisiones incrementales para asegurar que los métodos de verificación y validación del software sean los adecuados para los datos del software que se está desarrollando.

Se toman como criterios de requerimientos de la SVVPR los siguientes puntos:

- ✖ Reportes para una adecuada documentación de resultados de todas las revisiones, verificaciones y pruebas basadas en los requerimientos listados en el SVVP.
- ✖ Descripciones adecuadas de la configuración del software, para ser examinado, incluyendo pruebas de soporte de software y hardware.
- ✖ Planes de pruebas y diseño de pruebas para asegurar que todos los requerimientos son examinados.
- ✖ Procedimientos y situaciones de pruebas para asegurar que las entradas sean las adecuadas para el software.
- ✖ Programación de pruebas identificando que pruebas serán realizadas, cuando y por quien van a ser realizadas.

### **3.14.5. Auditoria Funcional:**

Esta verificación es realizada antes de la entrega del software, para verificar que todos los requerimientos especificados en la SRS fueron alcanzados.

La verificación funcional compara el código con los requerimientos documentados del software, como se estableció en SRS. Su propósito es asegurar que el código hace todo y solo lo que se indica en la documentación establecida por la SRS.

Se definen los siguientes puntos:

- ✖ Nomenclatura
- ✖ Número de identificación de la especificación
- ✖ Número de ítem de configuración
- ✖ La especificación de requerimientos de software
- ✖ Copia de código objeto
- ✖ Listado actualizado de ítems de configuración especificados
- ✖ El reporte de verificación y validación del software
- ✖ Listado del cumplimiento exitoso de pruebas funcionales

- ✖ Listado de todo lo planificado y pruebas que no fueron ejecutadas
- ✖ Actualizaciones para la documentación previamente liberada deberá ser revisada para asegurar su exactitud y consistencia.

#### ***3.14.6. Auditoria Física (PA):***

Esta verificación es realizada para verificar que el software y su documentación son internamente consistentes y están listas para su entrega.

La verificación física compara el código con su documentación de soporte, su propósito es asegurar que la documentación a ser entregada describa correctamente el código.

La documentación necesaria para realizar la verificación física es la siguiente:

- ✖ Descripción del diseño del software SDD
- ✖ Productos de software
- ✖ Documentación asociada

Para proveer evidencia de un adecuado control del contenido del sistema y consistencia del equipo de auditoria se examina lo siguiente:

- ✖ Los documentos de especificación del sistema para formatos y cumplimientos
- ✖ Reportes funcionales para discrepancia y acciones tomadas
- ✖ Descripción del diseño para símbolos, etiquetas, referencias y descripción de datos
- ✖ Los manuales para formatos de completitud y cumplimiento con la descripción de datos.
- ✖ Los elementos de Sw liberados en el medio son óptimos para transferir y transmitir
- ✖ Identificar los cambios en los ítems de configuración de datos

#### ***3.14.7. Auditorías del Proceso (IPA):***

Estas verificaciones serán desarrolladas dentro de los procesos de desarrollo del software, como ser el diseño para verificar la consistencia del diseño incluyendo:

- ✖ Código versus documentación del diseño
- ✖ Especificaciones de interfaces hardware / software.

- ✖ Implementación de diseño versus requerimientos funcionales

El objetivo es verificar la consistencia del producto a través del proceso del desarrollo para determinar que:

- ✖ Las interfaces hardware/software sean consistentes con el diseño de requerimientos en la SRS.
- ✖ Los requerimientos funcionales de la SRS, sean completamente probados por el SVVP.
- ✖ El diseño del producto especificado en la SDD, satisface los requerimientos funcionales del SRS.
- ✖ El código es consistente con la SDD.

### **3.15. Revisiones de gestión:**

Estas revisiones son realizadas periódicamente para evaluar la ejecución del SQAP. Estas revisiones deberán ser realizadas por el elemento organizacional del consultor.

Se podrían planificar otras revisiones, como por ejemplo la revisión de la documentación de usuario.

Para cada tipo de revisión, se debe explicar:

- ✖ Su objetivo.
- ✖ Qué producto es el que se evalúa.
- ✖ Sus propósitos.
- ✖ Cuál es el elemento organizativo responsable de llevar a cabo la revisión.
- ✖ Cuáles son los elementos organizativos que deben tomar parte de la revisión.
- ✖ Cuáles son los requisitos de revisión.
- ✖ Dónde deben documentarse los resultados de la revisión.
- ✖

Para cada tipo de auditoria se debe explicar:

- ✖ Su objetivo.
- ✖ Cuál es el elemento organizativo responsable de llevar a cabo la auditoria.
- ✖ Dónde deben documentarse los resultados de la auditoria.

- ❖ Cuáles son las entradas para la auditoria.

Se definen los tres tipos de revisiones (Evaluación de la calidad de los productos, Revisar el ajuste al proceso y Revisión Técnica Formal – RTF –), sus objetivos y mecanismos.

### **Evaluación de la calidad de los productos:**

#### **Objetivo:**

Revisar los productos que se definieron como claves para asegurar la calidad.

Detectar desviaciones en los objetivos de calidad definidos e informar a los responsables para que sean corregidas.

#### **Mecanismo:**

Se revisan los productos para verificar que cumplan con los estándares y con los objetivos de calidad definidas para el producto.

Se debe verificar que no queden correcciones sin resolver en los informes de revisión previos, si se encuentra alguna no resuelta, debe ser incluida en la siguiente revisión. Se debe identificar, documentar y seguir la pista a las desviaciones encontradas y verificar que se hayan realizado las correcciones.

Como salida se obtiene el Informe de revisión de SQA, que contiene todas las desviaciones o defectos encontrados durante la revisión. Este informe debe ser distribuido a los responsables del producto y se debe asegurar que ellos son conscientes de las desviaciones o discrepancias encontradas y de las acciones correctivas que deben realizar.

#### ***3.15.1. Revisar el ajuste al proceso:***

#### **Objetivo:**

Revisar si los productos se obtuvieron realizando las actividades que se indican en el Modelo de Proceso.

**Mecanismo:**

Se revisan los productos que se definen como claves para verificar el cumplimiento de las actividades definidas en el proceso, durante todo el ciclo de vida del software.

Se debe recoger la información necesaria de cada producto, buscando hacia atrás los productos previos que deberían haberse generado y son entrada para el producto objeto de revisión, para poder establecer los criterios de revisión y evaluar si el producto cumple con las especificaciones.

Esta información se obtiene de los siguientes documentos:

- ✖ Plan del Proyecto
- ✖ Plan de la iteración
- ✖ Plan de Verificación

Se debe verificar si todos los pasos del proceso de desarrollo son seguidos apropiadamente.

Antes de comenzar, se debe verificar en los informes de revisión previos que todas las desviaciones fueron corregidas, si no es así, las faltantes se incluyen para ser evaluadas.

Como salida se obtiene el Informe de revisión de SQA correspondiente a la evaluación de ajuste al Proceso, que contiene todas las desviaciones o defectos encontrados durante la revisión.

Este informe debe ser distribuido a los responsables de las actividades y se debe asegurar que ellos son conscientes de las desviaciones o discrepancias encontradas y de las acciones correctivas que deben realizar.

***3.15.2. Revisión Técnica Formal (RTF):*****Objetivo:**

Descubrir errores en la función, la lógica ó la implementación de cualquier producto del software, verificar que satisface sus especificaciones, que se ajusta a los estándares establecidos, señalando las posibles desviaciones detectadas.

### **Mecanismo:**

Es un proceso de revisión riguroso, su objetivo es llegar a detectar lo antes posible, los posibles defectos o desviaciones en los productos que se van generando a lo largo del desarrollo. Por esta característica se adopta esta práctica para productos que son de especial importancia.

En la reunión participan el responsable de SQA e integrantes del equipo de desarrollo.

Se debe convocar a la reunión formalmente a los involucrados, informar del material que ellos deben preparar por adelantado, llevar una lista de preguntas y dudas que surgen del estudio del producto a ser revisado.

Como salida se obtiene el Informe de RTF.

### **Requerimientos Mínimos:**

Los elementos mínimos que deberán ser revisados son:

- ✖ Especificación de Requerimientos
- ✖ Modelo de Diseño y Descripción de la Arquitectura
- ✖ Plan de Verificación y Validación
- ✖ Plan de Gestión del Proyecto
- ✖ Plan de Gestión de Configuración
- ✖ Diseño vs. Especificación de requerimientos
- ✖ Implementación vs. Diseño
- ✖ Verificación vs. Especificación de requerimientos

### **Agenda**

En esta sección se detallan todas las revisiones de calidad que se realizarán durante todo el proyecto, organizadas por fase e iteración.

#### **Fase I – Inicial**

##### **Iteración I**

Entregable	Realizado	Revisión	Tipo de revisión
Nombre del entregable o producto a revisar	Fase, iteración y semana en que se debe realizar la versión del producto a revisar	Semana, si se quiere también la fecha, en la que se realizará la revisión del entregable o producto	Tipo de revisión que se realizará: Evaluación de la calidad de los productos, Revisar el ajuste al proceso o Revisión Técnica Formal

.....

Iteración N

Fase II – Elaboración

Iteración I

.....

Iteración N

Fase III – Construcción

Iteración I

.....

Iteración N

Fase IV – Transición

Iteración I

.....

Iteración N

Y así sucesivamente para cada una de las fases del ciclo de vida de desarrollo de software.

### ***3.15.3. Gestión de Configuración***

El objetivo del SQA en esta área es asegurar que se realizan las actividades de gestión de configuración establecidas en el Plan de Configuración y que se realizan según lo establecido en el proceso. Se pueden definir las siguientes actividades mínimas que se deberían realizar:

- ✖ Asegurar que se generó la Línea Base del proyecto en el momento establecido en el modelo de proceso.
- ✖ Asegurar que la Línea Base del proyecto generada es correcta.
- ✖ Se verifica periódicamente que el Responsable de SCM mantiene apropiadamente el control de la línea base, así como el registro completo de cambios para requerimientos, diseño, código, verificación y documentación.
- ✖ Se monitorean los procedimientos del Comité de Control de Cambios para verificar que son efectivamente realizados como se especificaron en el Plan de configuración.

### ***3.15.4. Gestión de Problemas y Acciones Correlativas***

En esta sección se describen las prácticas y procedimientos que se van a utilizar para la notificación, seguimiento y resolución de problemas de software, así como las responsabilidades organizativas. El propósito de un sistema de Gestión de Problemas y Acciones Correlativas es:

- ✖ Asegurar que todos los problemas de documentan, se corrigen y no caen en el olvido.
- ✖ Asegurar que se evalúa la validez de los informes de problemas.
- ✖ Realimentar al desarrollador y el usuario sobre el estado de los problemas.
- ✖ Proporcionar datos para medir y predecir la calidad y fiabilidad del software.

Cualquier problema en el producto de software que sea encontrado durante el ciclo de vida de desarrollo de software, debe ser reportado a través de un reporte en el cual se detalla la fecha de cuando fue encontrado el problema, una identificación preliminar del mismo, descripción, etc., este reporte debe ser firmado por los que identificaron el problema, debe ser entregado a la organización responsable de los problemas.

La organización responsable de los problemas del software, es la organización del SQA, comandada por la organización del consultor, estas organizaciones son las encargadas de determinar el cronograma, lugar y temario, para llevar a fijar la acción correctiva del problema.

<b><u>REPORTE FINAL DE PROBLEMAS</u></b>		Pág. ....		
				
# De Reporte: .....	Lugar: .....	Fecha: ..... / ..... / ..... Hora: .....		
<p>a) Identificación del problema: ..... .....</p> <p>b) Descripción: ..... .....</p> <p>c) El evento ejecutado cuando se presentó el problema es: ..... .....</p> <p>d) Posibles orígenes del problema: ..... .....</p>				
<table border="1"><tr><td><b>Equipo de Trabajo:</b> <b>Nombre:</b> ..... ..... .....</td><td><b>Firma</b></td></tr></table>			<b>Equipo de Trabajo:</b> <b>Nombre:</b> ..... ..... .....	<b>Firma</b>
<b>Equipo de Trabajo:</b> <b>Nombre:</b> ..... ..... .....	<b>Firma</b>			

Las acciones a seguir para corregir los problemas presentados se describen de la siguiente manera:

- ☒ Antes de la identificación de la presencia de un problema se debe buscar los posibles orígenes del mismo sin desechar ninguna de las posibilidades, para esto se debe tomar dos rutas para generar el reporte de problema completo y consistente.
  - Registrar las causas sospechosas del origen del problema, esto asegura que no se descarta ninguna situación posible.
  - Registrar causas adicionales
  - Registrar causas plenamente identificadas, esto es, causas verificadas por detectores del problema
  - Registrar otras causas externas o relacionadas a las identificadas anteriormente
- ☒ Generar el reporte del problema, este debe estar completamente detallado y de acuerdo a los puntos que contiene el mismo, este reporte debe ser entregado a la organización responsable por los problemas.
- ☒ La organización responsable por los problemas, convocan a una reunión técnica, en la cual participarán además de la organización responsable, los elementos de las organizaciones afectadas por el problema, quienes describirán el problema y darán recomendaciones necesarias para solucionar el problema.
- ☒ La especificación de acciones correctivas generada en la reunión técnica, será entregada a los elementos organizacionales afectados por el problema para que estos implementen las acciones correctivas respectivas.

### **3.16. Herramientas, técnicas y metodologías**

En esta sección se identifican todas las herramientas, técnicas y metodologías que se van a utilizar en el desarrollo que apoyan el Aseguramiento de Calidad.

Algunas de las herramientas son:

- Utilidades del sistema operativo WINDOWS XP.
- Utilidades del sistema operativo LINUX.
- Documentación de ayuda.
- Instaladores.

Las técnicas que ayudan a la evaluación o mejora de la calidad son:

- ANSI / IEEE – STD 830 Guide for Software Requirements Specifications
- ANSI / IEEE – STD 1016 Recommended Practice for Software Design Descriptions
- ANSI / IEEE – STD 1008 Standard for Software Unit Testing
- ANSI / IEEE – STD 1063 Standard for Software User Documentation
- ANSI / IEEE – STD 1028 Standard for Software Reviews and Audits

Las metodologías de Aseguramiento de Calidad serán conjuntos integrados de técnicas, de entre los anteriores.

### ***3.16.1. Control del Código***

En esta sección se definen los métodos, técnicas y facilidades que se van a utilizar para controlar el almacenamiento y mantenimiento de versiones del código.

Se especifica un procedimiento de control del Código que:

- Defina cuál es el software que se va a controlar.
- Describa un método estándar para identificar, etiquetar y catalogar el software.
- Liste la localización física del software bajo control
- Describa la localización, forma de mantenimiento y de uso de las copias de
- seguridad.
- Describa los procedimientos para distribución de copias.
- Identifique la documentación que se verá afectada por los cambios.
- Describa los procedimientos para la construcción de una nueva versión.

### ***3.16.2. Control de Medios***

En esta sección se definen los métodos y facilidades que se van a utilizar para proteger el medio físico de accesos no autorizados y daños y degradaciones inesperadas, y las organizaciones responsables para realizar este control.

La organización responsable por esta tarea es la organización de desarrollo, con la supervisión de la organización de la SQA.

Se debería asegurar que:

- Está garantizado el almacenamiento y recuperación de software.
- El software está accesible únicamente para aquellos que lo necesitan.
- Se controla el entorno para que no se degrade el medio físico en el que se almacena el software.
- Se almacenan copias del software crítico y del código en línea base fuera de las instalaciones de la organización.

### ***3.16.3. Medio de Almacenamiento***

El medio del programa de computadora se define como aquellos medios sobre los cuales los datos son almacenados.

Se utilizarán los siguientes medios:

- Los discos duros como dispositivos primarios.
- Los CD's como almacenamiento secundario, para guardar las copias de seguridad.
- La documentación respectiva sobre el desarrollo de software (papel).

### ***3.16.4. Proceso de copias de seguridad***

Las copias de seguridad serán realizadas a la finalización de cada sesión de trabajo, registrándose la fecha y hora de copia de seguridad.

### ***3.16.5. Puntos de Control***

Para el acceso no autorizado, se debe asignar cuentas privilegiadas, cada usuario que interactúa con el software tendrá su propia cuenta de acuerdo al cargo que desempeñe.

Se cuenta con la integridad de la Base de Datos para la protección de los datos.

Se realizará una revisión periódica del software con el fin de que funcione óptimamente

### ***3.16.6. Control de Suministradores y Subcontratos***

En esta sección se explica de qué forma se va a asegurar que el software comprado o subcontratado cumple los requisitos técnicos.

### **3.16.7. Recolección, Mantenimiento y Retención de Registros**

Las organizaciones responsables por las tareas de esta sección, es la organización del consultor, en coordinación con la organización de la SQA.

Se identifica aquella documentación que se debe retener, y se especifican los métodos y facilidades que se utilizarán para recolectar, proteger y mantener esta documentación.

También se especificará el período de retención para cada tipo de registro.

Se puede registrar no sólo documentación, sino también los medios físicos que contienen las versiones de los programas y los materiales utilizados en las pruebas, para asegurar la repetición de los tests en el futuro.

Los documentos que son requeridos son los siguientes:

- Plan de garantía de calidad del software.
- Especificación de requerimientos del software.
- Descripción del diseño del software.
- Plan de verificación y validación del software.
- Documentación del usuario.

El mantenimiento de los registros del software, será realizado por versiones sucesivas de actualizaciones de las mismas, para esto se lleva un registro de actualizaciones de la documentación.

Los documentos verificados y validados, deben ser documentados en libros impresos, con tres copias de cada documento y almacenado en lugares diferentes y ambiente adecuados.

La retención de registros se realizará en cada finalización de las fases del ciclo de vida de desarrollo de software y según los puntos de verificación y validación.

**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y  
TELECOMUNICACIONES



## **4.Productividad basada en Herramientas CASE**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Integrantes:**

- |                                       |           |
|---------------------------------------|-----------|
| <b>1.</b> Machuca Yavita Miguel Angel | 216063558 |
| <b>2.</b> Torrez Garcia Jorge Luis    | 218137761 |

**Gestion II-2024 Santa Cruz – Bolivia**

## 4.1. ChatGpt

Esta herramienta ayuda en la gestión de proyectos, la programación, la verificación de calidad, la documentación, entre otras tareas. La integración de modelos de inteligencia artificial como ChatGPT en este ecosistema de desarrollo de software promete optimizar aún más la productividad, brindando asistencia en tiempo real, generación automática de código, análisis de requisitos, y más.



### 4.1.1. *La Integración de ChatGPT en el Desarrollo de Software*

ChatGPT puede funcionar como un asistente virtual que ayuda a los desarrolladores a generar y depurar código de manera más eficiente. Por ejemplo:

**Generación automática de código:** ChatGPT puede generar fragmentos de código o funciones completas a partir de una descripción en lenguaje natural. Esto reduce el tiempo dedicado a tareas repetitivas de codificación, permitiendo a los desarrolladores centrarse en tareas más complejas y de alto nivel.

**Depuración y revisión de código:** ChatGPT puede ayudar a identificar errores comunes en el código y sugerir soluciones, acelerando el proceso de depuración. Además, puede proporcionar recomendaciones para mejorar el rendimiento del código.

**Refactorización de código:** Puede sugerir maneras de optimizar y refactorizar el código, mejorando la calidad y mantenibilidad del software.

## **Beneficios de Integrar ChatGPT**

La combinación de ChatGPT trae consigo varios beneficios clave para los equipos de desarrollo de software:

**Aumento de la Productividad:** La automatización de tareas repetitivas, como la generación de código y documentación, reduce el tiempo que los desarrolladores deben dedicar a estas actividades, permitiéndoles centrarse en tareas más estratégicas y complejas.

**Mejora de la Calidad:** ChatGPT puede ayudar a identificar errores y ofrecer recomendaciones para mejorar la calidad del código y la documentación, lo que reduce los defectos en la fase de producción y mejora el mantenimiento del software.

**Reducción de Costos:** Al mejorar la eficiencia del desarrollo y reducir el tiempo de ciclo, los costos de desarrollo pueden disminuir, lo que permite a las empresas ofrecer productos de software de alta calidad a un menor costo.

**Optimización de la Gestión de Proyectos:** Con la integración de ChatGPT en las herramientas de gestión de proyectos, los equipos pueden obtener informes y recomendaciones más rápidas y precisas, lo que ayuda en la toma de decisiones informadas y mejora la planificación de los recursos.

## 4.2. Cursor



Ilustración 8 <https://www.cursor.com/>

**Cursor** es un editor de código basado en inteligencia artificial que está diseñado para mejorar la productividad de los desarrolladores mediante el uso de modelos avanzados de IA. Es similar a otros editores de código populares, pero con la diferencia de que incorpora potentes funcionalidades impulsadas por IA para asistir en tareas de programación, depuración, documentación, y generación de código.

Características clave de Cursor:

### **Asistencia con IA en Tiempo Real:**

Cursor utiliza modelos de IA para ofrecer sugerencias contextuales mientras escribes código, ayudando a generar funciones, completar líneas de código y proporcionar correcciones de errores.

La IA se integra directamente en el flujo de trabajo del desarrollador, lo que permite una codificación más rápida y eficiente.

### **Generación Automática de Código:**

Permite generar fragmentos de código a partir de descripciones en lenguaje natural. Por ejemplo, puedes escribir una solicitud en lenguaje natural como "Genera una función que ordene una lista de números", y la IA creará el código necesario.

### **Explicaciones y Documentación de Código:**

Ofrece la posibilidad de generar explicaciones detalladas de las líneas o bloques de código, lo cual es útil para la documentación y comprensión del código en proyectos complejos.

También puede generar comentarios automáticamente, lo que facilita la documentación del código mientras se escribe.

#### **Soporte para Varios Lenguajes de Programación:**

Al igual que otros editores de código, Cursor es compatible con múltiples lenguajes de programación, como Python, JavaScript, Java, C++, y otros, aprovechando el poder de la IA para ofrecer sugerencias específicas de acuerdo con el lenguaje que estás utilizando.

#### **Integración con Repositorios y Herramientas de Desarrollo:**

Cursor permite integrarse con repositorios como GitHub y otros sistemas de control de versiones, lo que facilita el flujo de trabajo de desarrollo colaborativo.

También se puede integrar con herramientas de CI/CD y otras plataformas de desarrollo para mejorar la automatización y la calidad del código.

#### **Refactorización de Código:**

Ofrece sugerencias para mejorar la estructura y legibilidad del código, como la refactorización de funciones largas o complejas, sugiriendo formas más eficientes de escribir el código sin cambiar su funcionalidad.

#### **Asistencia en la Depuración:**

Ayuda a identificar errores comunes o de sintaxis en el código y sugiere correcciones. También puede generar pruebas unitarias automáticamente para verificar que el código cumple con los requisitos funcionales.

#### **Aprendizaje Personalizado:**

Con el uso continuado, Cursor se adapta a los estilos de codificación y preferencias del usuario, mejorando la precisión de las sugerencias y las explicaciones.

### 4.3. Jira



Ilustración 9 <https://www.atlassian.com/software/jira>

**Jira** es una de las herramientas de gestión de proyectos y seguimiento de incidencias más populares, especialmente utilizada en entornos ágiles como Scrum y Kanban. Fue desarrollada por **Atlassian** y se utiliza principalmente para la gestión de tareas, la planificación de proyectos y el control de versiones en equipos de desarrollo de software. Jira se destaca por su flexibilidad, adaptabilidad y su amplia gama de integraciones con otras herramientas.

Características clave de Jira:

#### **Gestión de Proyectos Ágiles:**

**Scrum:** Jira permite crear tableros Scrum para gestionar proyectos en ciclos de trabajo iterativos (sprints). Los equipos pueden planificar, realizar seguimientos y analizar el progreso de sus tareas y sprints de forma sencilla.

**Kanban:** También ofrece soporte para el método Kanban, ayudando a los equipos a gestionar el flujo de trabajo mediante tableros visuales. Es útil para proyectos con tareas continuas o con menos definición de plazos.

#### **Gestión de Incidencias y Tareas:**

Jira permite crear y gestionar incidencias (bugs, tareas, mejoras, etc.) asociándolas a proyectos y equipos específicos. Las incidencias pueden tener asignados responsables, fechas de vencimiento, prioridades y etiquetas.

Las incidencias pueden categorizarse en tipos como **Epic** (grandes iniciativas o funcionalidades), **Story** (historias de usuario) y **Task** (tareas más específicas).

### **Seguimiento de Progreso:**

**Tableros de trabajo:** Los tableros Scrum y Kanban permiten visualizar el flujo de trabajo de las tareas y su evolución (por ejemplo, de "To Do" a "In Progress" a "Done").

**Burndown Charts:** Jira genera gráficos como el **Burndown Chart**, que muestra el progreso de un sprint, ayudando a los equipos a visualizar si están alcanzando sus objetivos a tiempo.

**Control de tiempo y estimaciones:** Los equipos pueden hacer estimaciones de tiempo para las tareas, lo que ayuda a gestionar los plazos de manera más eficiente.

### **Planificación y Organización:**

**Roadmaps:** Jira ofrece funcionalidades de roadmaps para planificar y visualizar los objetivos a largo plazo del proyecto. Los roadmaps permiten gestionar dependencias y asignar tareas de manera visual.

**Backlog:** Un backlog (lista priorizada de tareas) permite gestionar las tareas pendientes y priorizarlas según la importancia o urgencia.

**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y  
TELECOMUNICACIONES



## **5. Aspectos Legales**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Integrantes:**

- |                                       |           |
|---------------------------------------|-----------|
| <b>1.</b> Machuca Yavita Miguel Angel | 216063558 |
| <b>2.</b> Torrez Garcia Jorge Luis    | 218137761 |

## 5.1. Apertura de una empresa en Santa Cruz

Como parte de los aspectos legales para la apertura de una empresa de desarrollo de software, se realizó una recopilación de los pasos necesarios, los cuales se describen a continuación:

- **FUNDAEMPRESA:** Registro Comercial.
- **NIT:** Número de Identificación Tributaria.
- **Gobierno Municipal:** Padrón municipal y licencia de funcionamiento.
- **Caja de Salud:** Afiliación de Empresas/Afiliación del trabajador.
- **AFPs:** Registro de Empresas/Registro de personas.
- **Ministerio de Trabajo:** Registro del empleador.



### 5.1.1. FUNDAEMPRESA

**FUNDEMPRESA** es una Fundación sin fines de lucro que apoya al desarrollo empresarial y opera el Registro de Comercio de Bolivia, brindando un servicio eficiente a los empresarios y al Estado, orientada a la mejora continua.

Para registrar una Empresa se necesita lo siguiente:

**Control de Homonimia:** Primero se debe verificar la disponibilidad del nombre que

utilizará la Empresa que deseamos crear.

**Inscripción en el Registro de Comercio:** Inscripción de la empresa en el Registro de Comercio de acuerdo al tipo societario que tendrá:

- Empresa Unipersonal o Comerciantes Individuales
- Sociedad de Responsabilidad Limitada
- Sociedad Anónima

#### **5.1.2. NIT**

En Bolivia, el **NIT** (Número de Identificación Tributaria) es un número único de identificación fiscal que se otorga a las personas físicas y jurídicas que realizan actividades económicas en el país y que están obligadas a cumplir con sus obligaciones fiscales.

El NIT consta de 11 dígitos y es emitido por el Servicio de Impuestos Nacionales (SIN).

#### **Requisitos para sacar NIT**

##### **Si es persona natural:**

- ❖ Presentar original y copia del documento de identificación.
- ❖ Comprobante o recibo de servicio público como (agua, luz, gas, electricidad, entre otros)

##### **Si es persona jurídica:**

- ❖ Escrito o acta de constitución de la sociedad.
- ❖ Factura o comprobante de servicio público del local donde la compañía va a realizar o lleva a cabo sus actividades comerciales.
- ❖ El representante legal de la empresa debe presentar un poder que lo identifique como persona autorizada legalmente para representar a la sociedad y aquellos que la conforman.
- ❖ Documento de identificación del representante legal.
- ❖ Croquis o bosquejo del local, tienda o negocio donde se va a llevar a cabo la actividad

económica.

### **¿Qué impuestos debo pagar?**

#### **Los impuestos a pagar son:**

- ❖ Impuesto sobre las utilidades de las empresas, profesiones u oficios (personas naturales) - (IUE)
- ❖ Impuesto al valor agregado - (IVA)
- ❖ Impuesto a las transacciones - (IT)

#### ***5.1.3. Gobierno Municipal***

Se debe obtener la licencia de funcionamiento para actividad económica que realice la Empresa.

#### **Licencia de funcionamiento para actividades económicas en general**

Es la autorización otorgada por el GAMLP para la apertura de una actividad económica, adjuntando los requisitos establecidos, a efecto de obtener la autorización mediante la licencia de funcionamiento municipal.

#### ***5.1.4. Caja Nacional de Salud***

Las Empresas deben inscribirse a sus empleados a la Caja Nacional de Salud para cumplir con las normas sociales vigentes de acuerdo a la norma legal que posean, de la misma manera un trabajador que desee incorporarse a la caja de forma voluntaria también puede hacer sus consultas.

#### ***5.1.5. AFPs***

Las Administradoras de Fondos de Pensiones son las encargadas de administrar los recursos de los trabajadores para que cuando los mismos cumplan requisitos o a la edad de 58 años puedan acceder a una pensión digna. El principal objetivo es incrementar el nivel de ahorro del

país, de acuerdo a las leyes vigentes, todas las empresas están obligadas a registrarse ante las AFPs.

La Gestora Pública de la Seguridad Social de Largo Plazo (GESTORA), fue creada en el marco de la promulgación de la Ley N° 065 de Pensiones el 10 de diciembre de 2010, como una Empresa Pública Nacional Estratégica, siendo su objeto institucional, la administración y representación de los Fondos del Sistema Integral de Pensiones, gestión de prestaciones, beneficios y otros pagos correspondientes al Sistema Integral de Pensiones.

#### Cómo obtener la credencial de empleador

- ❖ Todos los empleadores deben habilitar su usuario operador y obtener sus credenciales, esto equivale a un número PIN para el ingreso al sistema. Para ello debe ingresar al portal web de la Gestora ([www.gestora.bo](http://www.gestora.bo)) y hacer clic en la pestaña Oficina Virtual.
- ❖ Debe elegir la opción Empleador e ir al botón “Regístrese aquí”. El sistema le pedirá elegir el tipo de empleador entre: privado con NIT, privado SUP, público GOB y cooperativista minero RCM.
- ❖ Cuando haya elegido el tipo, se generará un formulario en el que le pedirán datos de su empresa. También deberá dar datos del representante legal o del usuario operador, dependiendo de si es una empresa unipersonal o jurídica.
- ❖ Luego de este paso, al correo electrónico del representante legal o del usuario operador llegará un código de confirmación. Con el dato pase por única vez a las oficinas de la Gestora, donde le habilitarán las credenciales de su usuario operador.

#### :Cómo registrar a tus trabajadores dependientes?

- ❖ Para el aporte mensual deberá ingresar a la oficina virtual del portal web de la Gestora y llenar el Formulario de Pago de Contribuciones (FPC). Éste le solicitará los datos de los trabajadores

y luego le dará opciones para realizar el pago.

- ❖ Si el empleador no hizo el registro, el trabajador también puede hacerlo y habilitar su cuenta en las oficinas de la Gestora. Sólo necesita presentar su cédula de identidad.

#### ***5.1.6. Ministerio de Trabajo***

Todas las Sociedades Comerciales, Empresas Unipersonales de Bolivia, Sociedades Cooperativas, Sociedades Civiles, Asociaciones Civiles y Empresas Públicas; que cuenten con uno o más trabajadores y/o trabajadoras, deben inscribirse en el Registro Obligatorio de Empleadores a cargo del Ministerio de Trabajo, Empleo y Previsión Social.

#### **Propósito**

Esto permite la autorización y utilización del Libro de Asistencia y/o Sistema Alternativo de Control de Personal, así como la apertura del Libro de Accidentes.

### **5.2. Derecho de Autor**

#### ***5.2.1. SENAPI***

El Servicio Nacional de Propiedad Intelectual es una institución pública descentralizada que depende del Ministerio de Desarrollo Productivo y Economía Plural, con competencia de alcance nacional, tiene autonomía de gestión administrativa, legal y técnica; con la misión de administrar en forma descentralizada e integral el régimen de la Propiedad Intelectual en todos sus componentes, mediante una estricta observancia de los regímenes legales del mismo.

#### **Requisitos para el Registro de Software, Páginas WEB y Social Media**

- ❖ Carta de Solicitud de Registro dirigida a la Dirección de Derechos de Autor y Derechos Conexos.
- ❖ Fotocopia de la Cédula de Identidad del o de los Solicitantes, Autores y Titulares.
- ❖ Formulario debidamente llenado, impreso y firmado.

- ❖ El original y una fotocopia simple de los comprobantes de Depósito Bancario
- ❖ Una Copia ejecutable del programa de computación.
- ❖ Capturas de las pantallas y/o Video demostrativo.
- ❖ Documentación adicional, manuales u otros similares.

<p><b>SENAPI</b> Servicio Nacional de Propiedad Intelectual</p>	Número de Solicitud N° _____ <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> Fecha de presentación <input type="text"/> <input type="text"/> <input type="text"/> <b>REGISTRO DE PROGRAMAS DE COMPUTACIÓN</b>		
<b>DATOS DE AUTOR O AUTORES:</b>			
a) Nombre: _____	Seudónimo: _____		
Doc. de Identificación _____	Nacionalidad: _____		
Fecha nacimiento: _____	Fecha defunción: _____		
Dirección: _____	Ciudad: _____ País: _____		
b) Nombre: _____	Seudónimo: _____		
Doc. de Identificación _____	Nacionalidad: _____		
Fecha nacimiento: _____	Fecha defunción: _____		
Dirección: _____	Ciudad: _____ País: _____		
<b>DATOS TITULAR</b>			
Nombre: _____			
Documento de Identificación _____	Nacionalidad: _____		
Dirección: _____	Ciudad: _____ País: _____		
Representante legal: _____	Testimonio: _____		
<b>DATOS DE LA OBRA</b>			
Título: _____			
Año de creación: _____	Individual: <input type="checkbox"/>	Seudónima: <input type="checkbox"/>	Por encargo: <input type="checkbox"/>
Duración: _____	En colaboración: <input type="checkbox"/>	Inédita: <input type="checkbox"/>	Original: <input type="checkbox"/>
	Colectiva: <input type="checkbox"/>	Publicada: <input type="checkbox"/>	Derivada: <input type="checkbox"/>
	Anónima: <input type="checkbox"/>	Póstuma: <input type="checkbox"/>	En depósito: <input type="checkbox"/>
Otra: _____			
<b>BREVE DESCRIPCION DE FUNCIONES:</b>			
<b>ELEMENTOS DEL SOPORTE LOGICO APORTADOS:</b>			
a. Programa de computador <input type="checkbox"/>	b. Descripción del programa <input type="checkbox"/>	c. Material Auxiliar <input type="checkbox"/>	
Observaciones _____			
<b>DATOS DEL SOLICITANTE</b>			
a) Nombre: _____			
Documento de Identificación _____	Nacionalidad: _____		
Dirección: _____	Ciudad: _____ País: _____		
Transferencias: _____			
<b>DOMICILIO PARA EFECTOS DE NOTIFICACIÓN</b> (Domicilio que se encuentre dentro de la jurisdicción del SENAPI, solo en las ciudades de La Paz, El Alto, Cochabamba y Santa Cruz)			
<input type="text"/>	Departamento: <input type="text"/> ▼ Telf/Cel: <input type="text"/>		
NOTA: En caso de no llenar todos los campos <b>especificados</b> en el formulario será punto de observación a la hora de entregar dicho Documento			
Documentación que se adjunta: <input type="text"/>	Firma del solicitante <input type="text"/>		
Este formulario constituye en una declaración jurada respecto a los datos expresados por el solicitante			

Ilustración 10 Formulario de Registro de propiedad intelectual

**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y  
TELECOMUNICACIONES



## **6. Infraestructura para la Producción del Software**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Integrantes:**

- |                                |           |
|--------------------------------|-----------|
| 1. Machuca Yavita Miguel Angel | 216063558 |
| 2. Torrez Garcia Jorge Luis    | 218137761 |

## **6.1. Gestión de Configuración de Software**

### ***6.1.1. Introducción***

Esta sección es una guía para la correcta gestión de la configuración del proceso de desarrollo. Incluye los lineamientos necesarios para el manejo de versiones de los diferentes componentes, el manejo de las diferentes ramas y responsabilidades y actividades de los distintos actores.

### ***6.1.2. Alcance***

- El plan de configuración está basado en algunos supuestos que se detallarán:
- Tener control sobre cada una de las iteraciones y fases, de los productos generados en estas y de los cambios surgidos, evaluados y aprobados.
- Se deben incluir en control de configuración la mayor cantidad de productos posibles, tomando en cuenta siempre las restricciones dadas por la duración del proyecto y por la capacidad organizativa del equipo.
- Si es necesario, se realizarán las modificaciones que el cliente ha pedido y se actuará con las estrategias pertinentes.
- La elección de los elementos de configuración se realizará en base a los entregables, siendo esta responsabilidad del responsable de SCM, apoyado por los integrantes de cada disciplina.
- Al concluir cada una de las iteraciones, se llevará a cabo una prueba de verificación de la funcionalidad.

### ***6.1.3. Objetivos***

Los cambios dentro del desarrollo del SW pueden ocurrir en cualquier momento por lo tanto debemos estar preparados, las actividades de CGS sirven para:

- **Identificar el cambio de nuestro software.**

- **Controlar ese cambio.**
- **Garantizar que el cambio quede bien implantado.**
- **Informar el cambio.**

#### **6.1.4. Líneas Base**

Una línea base es un concepto de gestión de la configuración que lleva a controlar los cambios sin impedir seriamente los cambios justificados. El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) 610.12/1990 define una línea base como:

Una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambios.

Las líneas base se han definido en documentos donde se recogen los requisitos, desarrolla el programador.

#### **6.1.5. Tareas de la configuración del software**

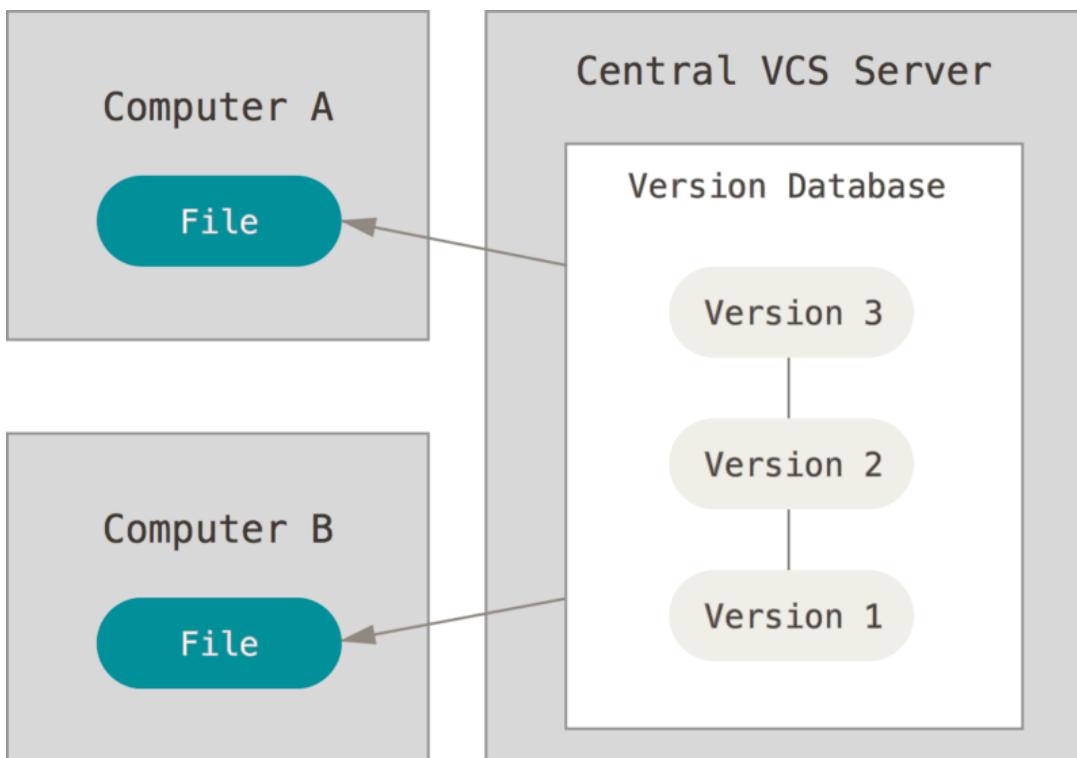
Se identificarán los elementos de la GCS en forma única y se organizarán bajo el enfoque de Orientación a Objetos (OO). Se identifican dos tipos de objetos:

- **Objetos básicos:** es una unidad de texto que pudo ser generada en el análisis, el diseño, la codificación o prueba.
- **Objetos compuestos:** son varios objetos simples, como por ejemplo la especificación de diseño.

Cada objeto poseerá una estructura que lo define en forma única como: nombre, descripción, número de revisión, fecha actual y hora, Información sobre la versión, información del cambio a realizar, quién solicita la modificación, quién la autoriza la modificación, quién se responsabiliza por la modificación, fecha de la última actualización, otros ECS posiblemente afectados.

## 6.2. Herramientas para el Control de Versiones

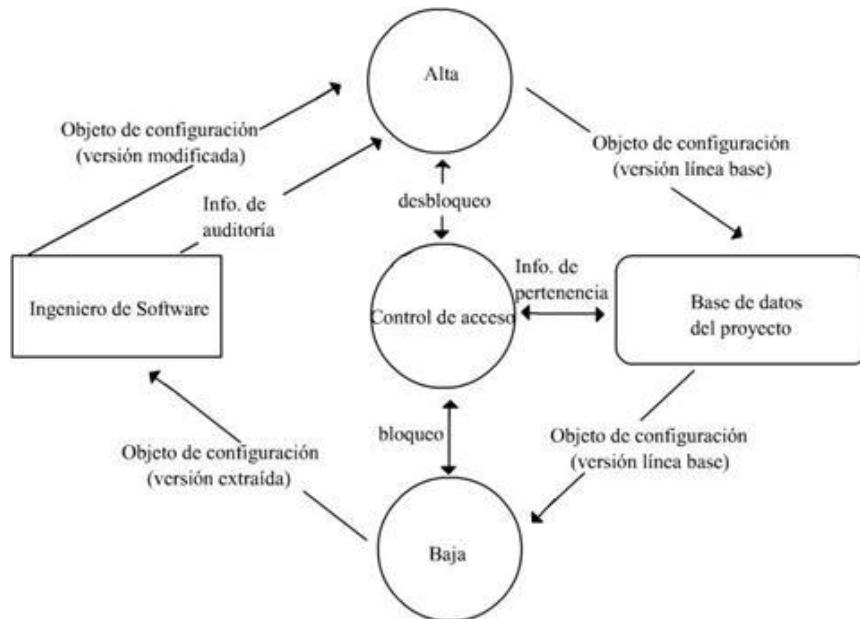
Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedes recuperar versiones específicas más adelante. Aunque en los ejemplos de este libro usarás archivos de código fuente como aquellos cuya versión está siendo controlada, en realidad puedes hacer lo mismo con casi cualquier tipo de archivo que encuentres en una computadora.



Los cambios incontrolados llevan muy rápido al caos, el control del cambio une procedimientos humanos con herramientas automáticas para proporcionar un mecanismo de control, el proceso de control debería contener lo siguiente:

Cuando los desarrolladores hacen un nuevo proyecto, siempre continúan haciéndole modificaciones al código. Incluso después de la puesta en marcha de los proyectos, todavía necesitan actualizar las versiones, corregir errores, agregar nuevas funciones, etc.

El sistema de control de versiones ayuda a registrar los cambios realizados al código. Aún más, registra quién realizó los cambios y puede restaurar el código borrado o modificado.



#### 6.2.1. Herramienta de control de versiones

**Github** permite que los desarrolladores alojen proyectos creando repositorios de forma gratuita. Pero hay que tener una cosa en mente, y es que para poder subir gratis los proyectos deberán ser de código abierto. Y no quieres que tu aplicación sea de código abierto, la plataforma también tiene una versión de pago para alojar proyectos de forma privada.

Como te hemos mencionado más arriba, en Github también puedes entrar a los proyectos de los demás y colaborar para mejorarlo. Esto quiere decir que los usuarios pueden opinar, dejar sus comentarios sobre el código, colaborar y contribuir mejorando el código. También pueden reportar errores para que los desarrolladores lo mejoren.

Github también ofrece una serie de herramientas propias con las que complementar las ventajas que ya tiene el sistema Git de por sí solo. Por ejemplo, puedes crear una Wiki para cada proyecto, de forma que puedas ofrecer toda la información sobre él y anotar todos los cambios de las diferentes versiones.

## **6.3. Herramientas utilizadas en el proyecto**

### **6.3.1. *GitHub***

GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.

Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.

Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original.

Las herramientas que proporciona son:

- Una wiki para el mantenimiento de las distintas versiones de las páginas.
- Un sistema de seguimiento de problemas que permiten a los miembros de tu equipo detallar un problema con tu software o una sugerencia que deseen hacer.
- Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero y debatir sobre determinados cambios realizados en un commit específico.
- Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

Su uso es para:

En nuestra especialidad “Programación”, fuimos aprendiendo cosas y creando programas de código abierto, fomentando el software libre; es por eso que presentamos esta gran herramienta enfocada al crecimiento de proyectos comunitarios y libres.

En esta página podremos crear una cuenta gratuita y comenzar a subir repositorios de código (o crearlos desde 0), para que con la ayuda de todos esos proyectos mejoren; así como también fortalecerlos proyectos de los demás para crecer como grupo.

### ***6.3.2. Visual Studio Code***

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft. Actualmente, como editor de código plano es el más usado y extendido.

#### **Razón de uso:**

Se eligió esta herramienta, ya que a pesar de ser solo un editor de código Fuente, la extensa lista de extensiones lo vuelve potente, hasta llegar al punto de considerarse un IDE.

**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y  
TELECOMUNICACIONES



## **7. Desarrollo del Sitio Web**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Integrantes:**

- |                                       |           |
|---------------------------------------|-----------|
| <b>1.</b> Machuca Yavita Miguel Angel | 216063558 |
| <b>2.</b> Torrez Garcia Jorge Luis    | 218137761 |

## **7.1. Sitio web**

### ***7.1.1. Información Principal***

Nuestro Software poseerá una Landing Page donde se podrá ver las características principales de la empresa, las secciones que se mostraran serán:

- Inicio
- Servicios
- Nosotros
- Contactos
- Chatbot

### ***7.1.2. Logo***



### ***7.1.3. Planificación***

Definir los objetivos del sitio web:

1. Dar a conocer los productos y servicios de **ingversoft**.
2. Informar a los usuarios sobre los productos y servicios ofrecidos.
3. Aumentar las ventas de los productos y servicios.
4. Establecer una presencia en línea y dar a conocer la empresa.
5. Ofrecer planes de precios del software.
6. Implementar un Chatbot para una comunicación constante con los clientes.
7. Implementar un contáctenos.

## 7.1.4. Diseño y Prototipo

### 7.1.4.1. Inicio



The screenshot shows the homepage of IngEverSoft. At the top left is the company logo. The top navigation bar includes links for INICIO, SERVICIOS, NOSOTROS, and CONTACTANOS. On the far right of the header is a "Cotiza aquí" button. The main visual is a photograph of several people in a modern office environment, focused on working at desks with multiple monitors. Overlaid on this image is a large, bold text block: "IngEverSoft Donde la creatividad se encuentra con la funcionalidad". Below this, a smaller text block reads: "¿Buscas llevar tu software a un nivel superior? En nuestra desarrolladora de software, convertimos ideas en soluciones innovadoras. Con un enfoque centrado en la calidad y la excelencia, creamos software que destaca en un mercado digital competitivo. Confía en nosotros para impulsar tu negocio hacia el éxito digital." The overall design is professional and tech-oriented.

### 7.1.4.2. Servicios



The screenshot shows the "Servicios" (Services) page of IngEverSoft. The top navigation bar is identical to the homepage, with links for INICIO, SERVICIOS, NOSOTROS, and CONTACTANOS, and a "Cotiza aquí" button. The main visual is a photograph of a team of people working together on a project, with one person in the foreground looking at a computer screen. Overlaid on this image is a large, bold text block: "Servicios". Below this, a blue call-to-action section contains the question: "¿Cómo podemos ayudarte?". At the bottom of the page, there are three service categories with brief descriptions: "Diseño Creativo", "Investigación y desarrollo", and "Branding & Marketing".

Diseño Creativo	Investigación y desarrollo	Branding & Marketing
Nuestro equipo de diseñadores gráficos y web profesionales lo	Le ayudamos a desarrollar el sitio web adecuado con una buena	¿Necesita ayuda con la marca y el marketing para llegar a una

### 7.1.4.3.Nosotros

Somos un equipo motivado y expertos en desarrollo web con más de 4 años de experiencia en creación de sitios web y marketing para ayudar a las empresas a crecer en línea.

Si estás buscando los mejores servicios de diseño, desarrollo y marketing de sitios web para hacer crecer tu negocio en línea, estás en el lugar correcto para ayudarte. Forma un equipo de empresas comprometidas con tremenda experiencia en diseño, desarrollo y marketing web que lo acompañarán en todo momento.

Nuestros Valores

### 7.1.4.4.Proyectos:

Tienes en mente un proyecto?

Podemos ayudarte a hacer realidad tus ideas. Hablemos de lo que podemos construir y recaudar juntos.

Contactate con nosotros →

**¡Construyamos juntos un negocio exitoso!**

Cuando nos elige a nosotros, no solo está haciendo crecer su negocio. También lo respaldamos y ponemos lo mejor de nosotros para contribuir al crecimiento de todo su equipo y organización. Entonces, si estás buscando el socio adecuado que lo ayude a construir una buena presencia en línea y generar más conversiones e ingresos, ¡Estamos aquí!

#### *7.1.4.5. Contactos*

## Contactanos

¡Estamos aquí para ayudarte!



45 Calle Los Pachios, Radial 26 – 5to anillo – Santa Cruz

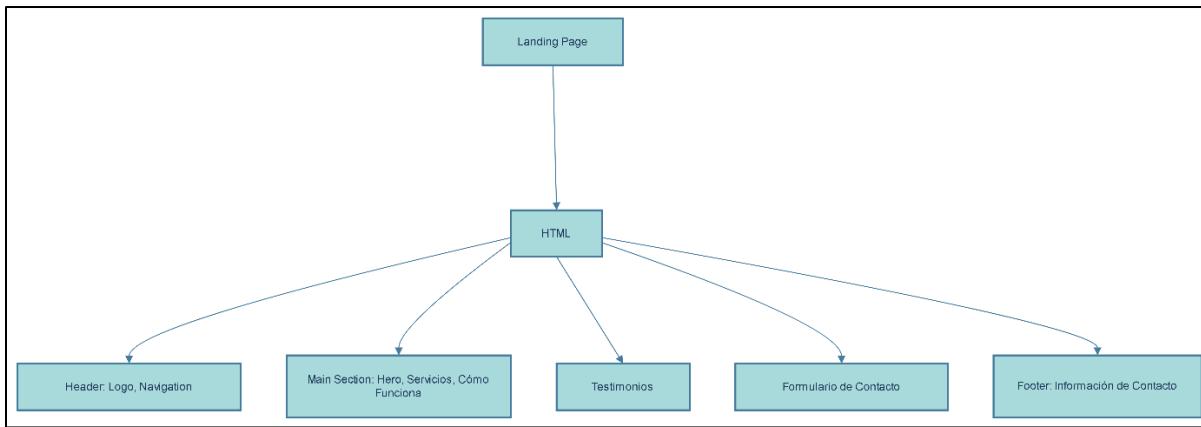


### Envíanos un mensaje

Nombres	Apellidos
dirección de correo electrónico	
Empresa	
Tu mensaje	

[Enviar](#)

### **7.1.5. Arquitectura de la Página Web**



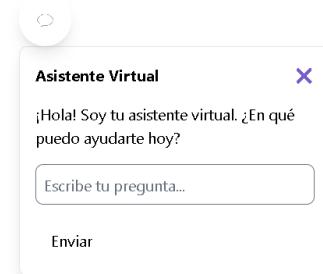
### **7.1.6. Chatbot**

#### **Así Funciona Nuestro Servicio**

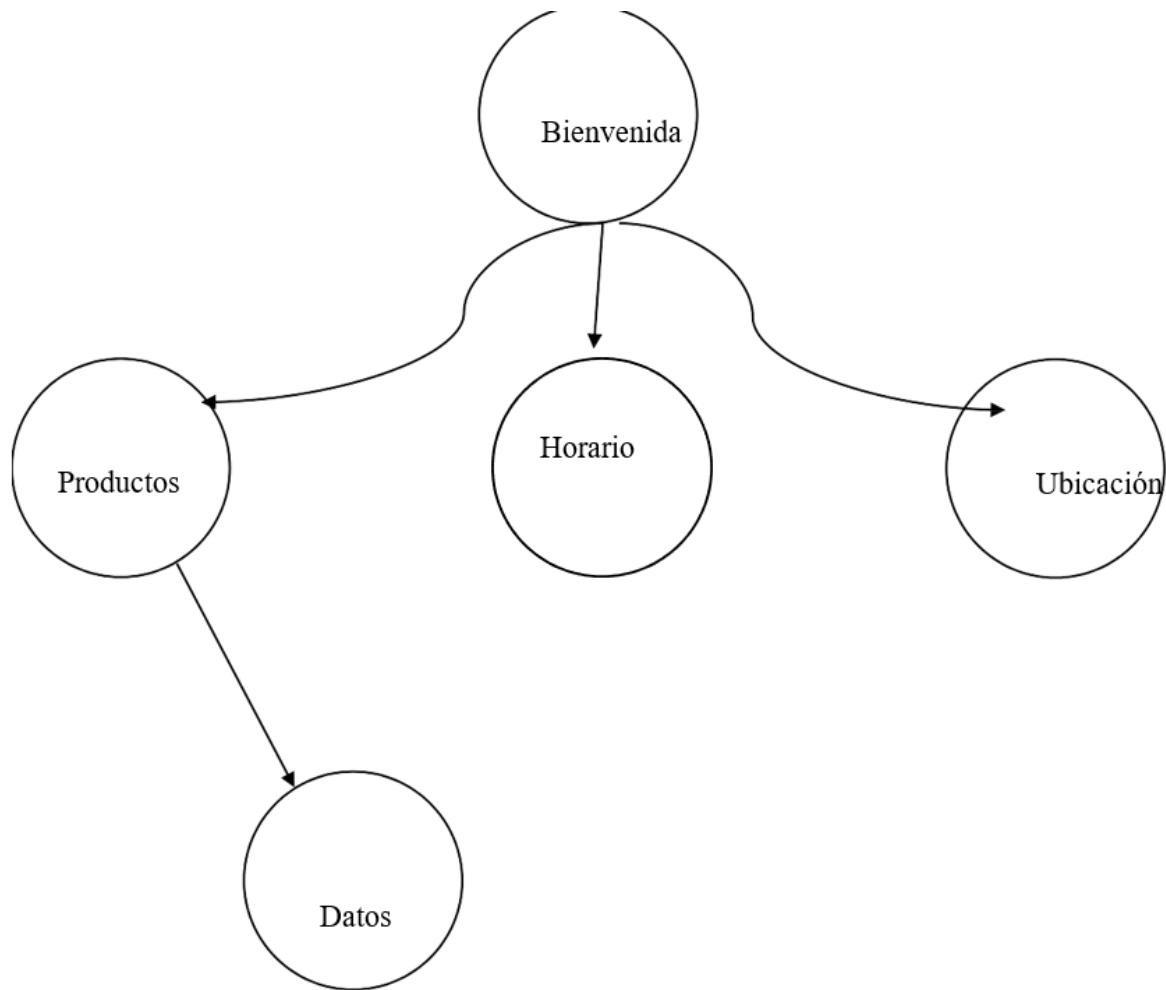
1. **Paso 1:** Define tus objetivos publicitarios con nuestro equipo.
2. **Paso 2:** Asignamos los mercaderistas especializados para tu marca.
3. **Paso 3:** Monitoreamos el rendimiento y ajustamos las estrategias según los resultados.

#### **Lo que Dicen Nuestros Clientes**

"Desde que trabajamos con estos mercaderistas, nuestras ventas han incrementado un 30% en tiendas..."



El chatbot está en la parte inferior derecha, entre sus funcionalidades, el sistema de estados para las preguntas se corresponde de la siguiente manera:



Donde **Producto** se relaciona a la contestación para la asignación de un encargado competente en el área. A esta respuesta se le determinará un envío de correo electrónico relacionado a la empresa de manera automática, para el interesado poder conversar con un empleado asignado a determinar dichos puntos. Este envío de correo en primera instancia se hace

**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y  
TELECOMUNICACIONES



## **8. Estudio de Mercado**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Integrantes:**

- |                                |           |
|--------------------------------|-----------|
| 1. Machuca Yavita Miguel Angel | 216063558 |
| 2. Torrez Garcia Jorge Luis    | 218137761 |

## **8.1. Objetivo del Estudio**

El objetivo de este estudio de mercado es cuantificar el tamaño del mercado y analizar el alcance potencial para un software web que facilite el control y monitoreo de actividades publicitarias para mercaderistas. Este software busca mejorar la eficiencia de los procesos publicitarios, optimizando la planificación, ejecución y análisis de las campañas publicitarias en distintos puntos de venta.

## **8.1. Mercado Objetivo**

El mercado objetivo de Ingeversoft abarca principalmente a mercaderistas y empresas de publicidad que operan en el sector minorista y mayorista. En particular, se dirige a las siguientes categorías:

**Mercaderistas:** Profesionales que gestionan campañas publicitarias en puntos de venta.

**Agencias de publicidad:** Empresas que gestionan campañas publicitarias para terceros.

**Comercios y retailers:** Negocios con múltiples puntos de venta que requieren una gestión centralizada de sus esfuerzos publicitarios.

## **8.2. Demografía:**

**Edad:** 25-45 años, en su mayoría jóvenes profesionales.

**Ubicación:** Empresas ubicadas principalmente en áreas urbanas de Bolivia, con potencial de expansión a mercados internacionales en el futuro.

**Industria:** Retail, productos de consumo masivo, agencias de marketing, y pequeñas y medianas empresas.

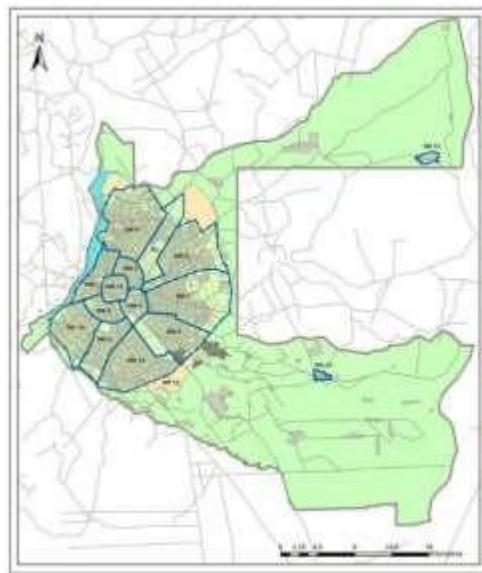
## **8.3. Área del mercado**

### **8.3.1. Extensión Geográfica**

En cuanto a su extensión geográfica el mercado está centrado en la región Metropolitana del Municipio de Santa Cruz de la Sierra.

### **8.3.2. Población**

La población consumidora dentro del área de influencia del mercado considerado en el estudio son empresas que tienen la modalidad de servicio de publicidad de “puerta a puerta” y necesitan realizar uso de un software de control a sus empleados o mercaderistas.



Área demandada para el estudio de mercado.

## **8.4. Investigación de Mercado**

### **8.4.1. Tipo de Investigación**

Se llevará a cabo una investigación **aplicada** y **descriptiva**, con el objetivo de obtener información relevante y detallada sobre el mercado objetivo. La investigación aplicada tiene como fin la implementación práctica de los resultados obtenidos, mientras que la investigación

descriptiva proporcionará detalles tanto generales como específicos sobre las necesidades y comportamientos del público objetivo.

#### **8.4.2. *Técnicas de Investigación***

Para la recopilación de datos, se utilizará una **encuesta electrónica** dirigida a mercaderistas y empresas de publicidad. Esta técnica es adecuada dado que permite llegar a un número considerable de participantes de manera eficiente, sin limitaciones geográficas. La encuesta se enviará por correo electrónico y redes sociales, con el fin de asegurar una cobertura amplia.

#### **8.4.3. *Método de Investigación de Mercados***

Se utilizará un **método cuantitativo** con el objetivo de estimar el tamaño del mercado y las preferencias de los posibles clientes. Las respuestas serán analizadas con técnicas estadísticas para determinar los segmentos más rentables y viables. Además, se aplicará un cuestionario con preguntas de respuesta cerrada para facilitar el análisis de los datos.

#### **8.4.4. *Ficha Técnica***

**Población:** Empresas de publicidad, mercaderistas y comercios que gestionan campañas publicitarias en puntos de venta.

**Elemento:** Personas responsables de la gestión de campañas publicitarias dentro de las empresas objetivo.

**Unidad de Muestreo:** Mercaderistas y responsables de marketing de empresas con presencia en Bolivia.

**Alcance:** Inicialmente en Santa Cruz de la Sierra, con intención de expandirse a nivel nacional.

**Tiempo de Respuesta:** 1 mes para la recopilación de datos.

#### **8.4.5. Marco Muestral**

Las encuestas se realizarán en el mercado de Santa Cruz de la Sierra, con el objetivo de obtener un panorama representativo de la región central de Bolivia.

#### **8.4.6. Tipo de Muestreo**

Se empleará un **muestreo no probabilístico**, utilizando una técnica de muestreo por conveniencia, dado que el acceso a empresas y profesionales de marketing es limitado en algunas áreas.

#### **8.4.7. Tamaño de la Muestra**

Para determinar el tamaño de la muestra, se utilizará la fórmula de muestra para población infinita:

$$n = \frac{Z^2 \times p \times (1 - p)}{E^2}$$

Donde:

- **Z = 95%** (Nivel de confianza).
- **p = 0.5** (Proporción estimada de clientes potenciales).
- **E = 5%** (Margen de error).

Con estos valores, el tamaño mínimo de la muestra es de aproximadamente **384 encuestas**.

### **8.5. Análisis de Mercado**

#### **8.5.1. Alcance y Potencial del Mercado**

Según estudios previos y la proyección del crecimiento del mercado publicitario digital en Bolivia, se estima que el 60% de las empresas de retail y agencias de publicidad utilizarán software de gestión de campañas publicitarias en los próximos 3 años. Esto sugiere un mercado potencial

de aproximadamente **500 empresas en Santa Cruz de la Sierra** que podrían beneficiarse de la solución.

#### **8.5.2. *Monetización del Software***

El modelo de monetización del software se basará en un sistema **SaaS (Software as a Service)** con suscripciones mensuales o anuales. Las opciones de precios son las siguientes:

- **Plan Básico:** \$50 mensuales para pequeñas empresas con hasta 3 puntos de venta.
- **Plan Profesional:** \$150 mensuales para empresas medianas con hasta 10 puntos de venta.
- **Plan Enterprise:** \$500 mensuales para grandes empresas con más de 10 puntos de venta.

Además, se puede incluir un **servicio adicional de capacitación y soporte personalizado** con tarifas de \$100 por sesión de capacitación de 2 horas.

#### **8.5.3. *Estrategia de Marketing***

Para la promoción del software, se llevará a cabo una **campaña de publicidad digital** a través de plataformas como Facebook Ads, Google Ads y LinkedIn, segmentada específicamente para empresas de retail y agencias de publicidad. También se implementará una **estrategia de marketing de contenido**, ofreciendo artículos y casos de estudio sobre cómo mejorar las campañas publicitarias utilizando herramientas digitales.

### **8.6. Encuesta**

#### **Sección 1: Perfil del Respondente**

1. **¿Cuál es el tamaño de su empresa?**
  - Pequeña (1-10 empleados)
  - Mediana (11-50 empleados)
  - Grande (más de 50 empleados)
2. **¿En qué sector se encuentra principalmente su empresa?**
  - Retail
  - Agencia de publicidad

- Tecnología
  - Otros: \_\_\_\_\_
3. ¿Cuántos puntos de venta o sucursales gestiona su empresa?
- 1-3
  - 4-10
  - 11-50
  - Más de 50
4. ¿Cuál es el perfil de su público objetivo principal?
- Jóvenes (18-30 años)
  - Adultos (31-50 años)
  - Adultos mayores (más de 50 años)
  - Otros: \_\_\_\_\_

**Sección 2: Uso Actual de Herramientas Publicitarias 5. ¿Utiliza su empresa alguna herramienta digital para gestionar campañas publicitarias?**

- Sí
  - No
6. ¿Qué tipo de herramientas utiliza actualmente para controlar las campañas publicitarias? (Seleccione todas las que apliquen)
- Software de gestión de campañas publicitarias
  - Hojas de cálculo (Excel, Google Sheets)
  - Herramientas específicas de las redes sociales (Facebook Ads, Google Ads, etc.)
  - Ninguna
  - Otros: \_\_\_\_\_
7. ¿Cuáles son los principales retos que enfrenta su empresa en la gestión de campañas publicitarias? (Seleccione hasta 3 opciones)
- Falta de integración entre plataformas
  - Dificultad para medir el rendimiento de las campañas
  - Falta de visibilidad en tiempo real
  - Gestión ineficiente del presupuesto publicitario
  - Dificultad para coordinar entre múltiples puntos de venta
  - Otros: \_\_\_\_\_

**Sección 3: Interés en el Software de Control Publicitario 8. ¿Estaría interesado en un software que le permita gestionar todas sus campañas publicitarias desde una plataforma centralizada?**

- Sí
- No
- Tal vez

**9. ¿Qué funcionalidades le gustaría que tuviera un software de control publicitario? (Seleccione hasta 3 opciones)**

- Monitoreo en tiempo real de las campañas
- Integración con plataformas de redes sociales (Facebook, Instagram, Google Ads)
- Reportes automatizados y análisis de resultados
- Control de presupuesto publicitario
- Planificación y programación de campañas
- Gestión de promociones y descuentos
- Otros: \_\_\_\_\_

**10. ¿Cuánto estaría dispuesto a pagar por un software mensual que facilite la gestión y monitoreo de sus campañas publicitarias?**

- \$50 - \$100 al mes
- \$101 - \$200 al mes
- \$201 - \$500 al mes
- Más de \$500 al mes

**11. ¿Qué tan importante es para su empresa que el software sea accesible desde dispositivos móviles?**

- Muy importante
- Importante
- Poco importante
- No es importante

**12. ¿Preferiría una solución personalizada para su empresa o una solución estándar?**

- Solución personalizada
- Solución estándar

**Sección 4: Impacto y Valor Agregado del Software 13. ¿Qué tan probable es que recomiende un software de control publicitario a otras empresas o colegas del sector? - [ ] Muy probable - [ ] Probable - [ ] Poco probable - [ ] Nada probable**

14. ¿Cuáles serían los factores clave que le convencerían de invertir en un software de control publicitario? (Seleccione hasta 3 opciones)

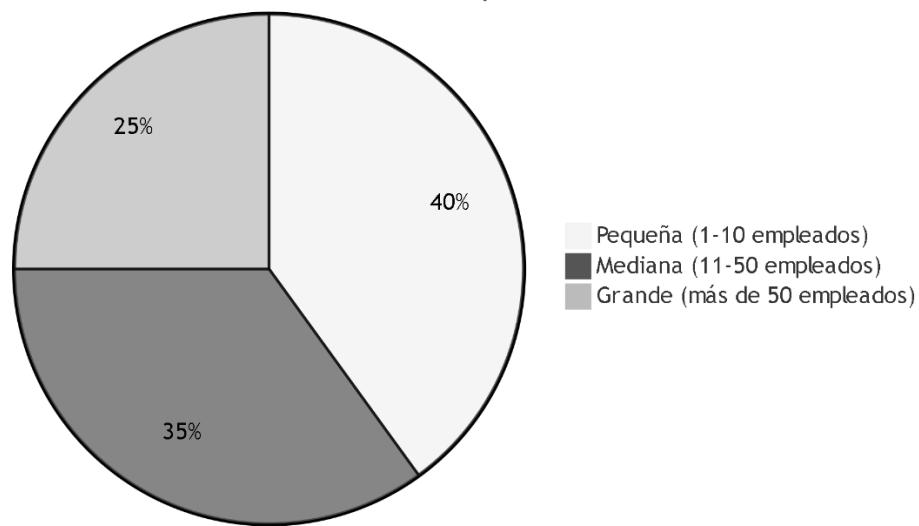
- Precio accesible
- Facilidad de uso
- Funcionalidades completas
- Buen soporte técnico
- Integración con otras herramientas existentes
- Seguridad de los datos

15. ¿Qué otras sugerencias o características le gustaría que tuviera un software de control publicitario?

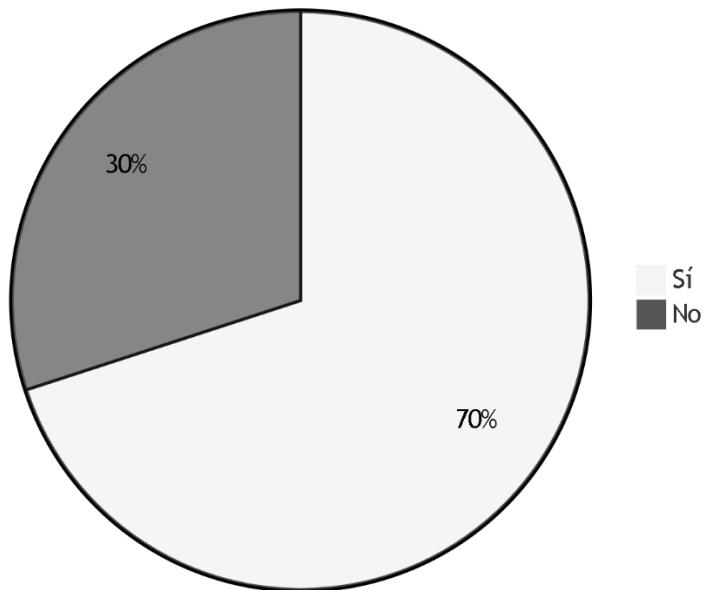
- \_\_\_\_\_

## 8.7. Resultados

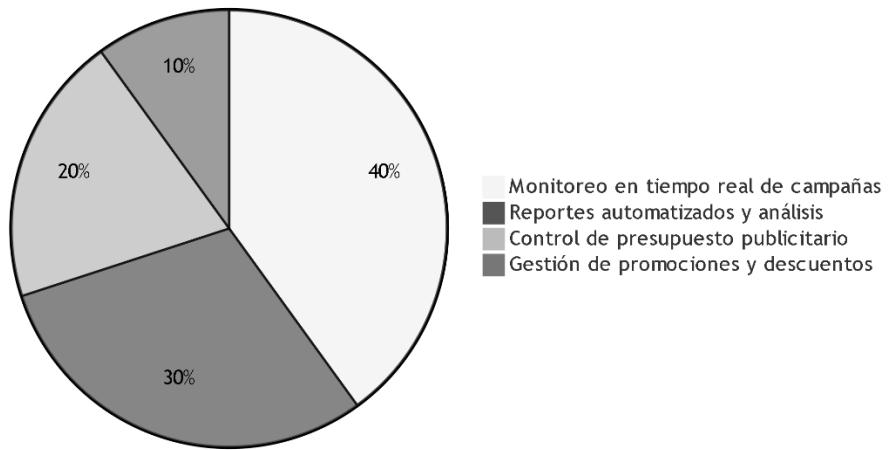
Distribución del Tamaño de las Empresas



### Uso de Herramientas Digitales



### Funcionalidades Deseadas para el Software



## 8.8. Resultados Esperados

Con la implementación de este software y las estrategias de marketing, se espera que en los primeros 12 meses se logren suscripciones de al menos 50 empresas, generando ingresos de aproximadamente **\$30,000** anuales.



**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
**FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y**  
**TELECOMUNICACIONES**



## **9. Pruebas en el Software**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Integrantes:**

- |                                       |           |
|---------------------------------------|-----------|
| <b>1.</b> Machuca Yavita Miguel Angel | 216063558 |
| <b>2.</b> Torrez Garcia Jorge Luis    | 218137761 |

## **9.1. Pruebas en el Software**

### **2.1. Planificación de Pruebas**

Para la realización de las pruebas al software se utilizará el método de pruebas de caja negra. Las pruebas de caja negra, también llamadas pruebas de comportamiento se enfocan en los requerimientos funcionales del software, es decir, las técnicas de prueba de la caja negra permiten derivar conjuntos de condiciones de entrada que revisara por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra intentan encontrar error en la categoría siguiente:

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos o en el acceso a la base de datos externa.
- Errores de comportamiento o rendimiento.
- Errores de inicialización y terminación

## **9.2. Diseño de Pruebas**

### **2.2.1. Identificación y Estructuración de Procedimiento de Prueba**

Se presenta la siguiente estructura para realizar la prueba a los casos de prueba:

- Nombre del caso de prueba: se deberá de especificar el nombre completo del caso de uso a realizar la prueba.

- Entrada: Información que requiera el caso de uso para producir o generar resultados que sean significativos para el usuario.
- Condición de ejecución: Cuando el caso de prueba presente alguna restricción para demostrar su correcto funcionamiento
- Resultado: Transcribir lo que dio como respuesta el software para su validación o corrección si fuese necesario.

### **9.3. Pruebas Realizadas**

#### ***2.3.1. Pruebas de Caja Negra***

##### **Autenticación para la Web:**

###### **Entrada:**

Email: “Correo Electrónico del usuario”

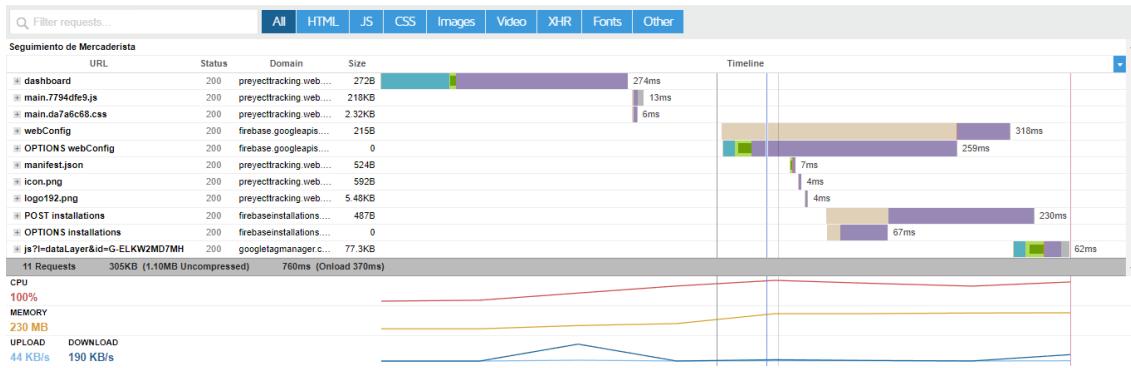
Contraseña: “Contraseña del usuario”

###### **Condiciones:**

- El actor que se autentifica debe ser de tipo Administrador
- Se validará sus credenciales de autenticación

### **9.4. Pruebas de Rendimiento**

En cualquier aplicación web, lograr un rendimiento óptimo y una carga eficiente de recursos es fundamental. Esto se traduce en tiempos de espera más cortos al acceder al software y la capacidad de renderizar la aplicación incluso en situaciones con una señal de internet débil.



Utilizando la herramienta automatizada Gtmetrix, obtenemos un gráfico que resume el consumo de recursos de nuestra aplicación en cada solicitud. Observamos que se requieren al menos 230MB de RAM libre, una cantidad relativamente baja. Además, el uso de ancho de banda de carga y descarga de nuestra aplicación es mínimo, lo que permite su uso incluso en redes 3G debido a su bajo consumo de velocidad de conexión.

## 9.5. Pruebas de Vulnerabilidad

La seguridad es un aspecto crucial al ofrecer un software como producto, ya que la información almacenada es de gran importancia para los clientes. Para evaluar la seguridad, hemos utilizado la herramienta Gtmetrix, la cual realiza un análisis exhaustivo escaneando cada componente, tanto interno como externo, implementado en nuestra aplicación web.

IMPACT	AUDIT	Show Audits Relevant to	All	FCP	LCP	TBT	CLS
Med-Low	Serve static assets with an efficient cache policy	Potential savings of 176KB					
Low	Avoid enormous network payloads <small>LCP</small>	Total size was 306KB					
Low	Reduce JavaScript execution time <small>TBT</small>	213ms spent executing JavaScript					
Low	Reduce initial server response time <small>FCP LCP</small>	Root document took 188ms					
Low	Avoid chaining critical requests <small>FCP LCP</small>	1 chain found					
Low	Reduce unused JavaScript <small>LCP</small>	Potential savings of 184KB					
N/A	Avoid an excessive DOM size <small>TBT</small>	22 elements					
N/A	Largest Contentful Paint element <small>LCP</small>	430 ms					
N/A	Minimize main-thread work <small>TBT</small>	Main-thread busy for 447ms					
N/A	Reduce the impact of third-party code <small>TBT</small>	Total size was 78.5KB					
N/A	Eliminate render-blocking resources <small>FCP LCP</small>						
N/A	Avoid serving legacy JavaScript to modern browsers <small>TBT</small>						
N/A	Avoid large layout shifts <small>CLS</small>						
N/A	User Timing marks and measures						

Gtmetrix nos proporciona soluciones potenciales para ciertos componentes que podrían presentar vulnerabilidades, calificándolos en diferentes niveles de riesgo. En nuestro caso, no se ha identificado ningún riesgo alto que pueda comprometer la integridad de la aplicación.



En resumen, según Gtmetrix, nuestra aplicación web recibe una calificación de A, lo que indica que no presenta problemas graves de vulnerabilidad y su rendimiento está estructurado correctamente en un 99%.

**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
**FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y**  
**TELECOMUNICACIONES**



## **10. Marketing**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Integrantes:**

- |                                       |           |
|---------------------------------------|-----------|
| <b>1.</b> Machuca Yavita Miguel Angel | 216063558 |
| <b>2.</b> Torrez Garcia Jorge Luis    | 218137761 |

**Gestion II-2024 Santa Cruz – Bolivia**

## **10.1. Descripción**

Este plan se presenta para IngEverSoft, un proyecto joven centrado en la gestión, control y seguimiento de los mercaderistas publicitarios, con sede en el Municipio de Santa Cruz de la Sierra, Bolivia. Actualmente varias empresas utilizan personas externas denominadas mercaderistas publicitarios que básicamente se encargan de promocionar los productos al público específico. El objetivo de IngEverSoft es ayudar a las empresas a controlar su personal externo debido a que si no cumplen con su trabajo las ventas serán bajas. Entre nuestros objetivos a largo plazo está el posicionamiento del servicio no solo a nivel municipal o nacional, sino, llegar también a tener un alcance en urbes internacionales.

## **10.2. Organización**

Basado en ciertos elementos de la estructura y diseño organizacional. La estructura de la organización estaría compuesta de la siguiente manera:



### **10.3. Servicios**

El servicio que se pretende ofrecer es el de posicionamiento geográfico de los promotores y además de gestionar los pedidos realizados a través de una aplicación móvil.

### **10.4. Perfiles de clientes**

Actualmente el perfil de cliente que se pretende tener son todas las personas jurídicas del área urbana que utilicen en sus día a día medios electrónicos y estén buscando expandir su producto a zonas alejadas.

### **10.5. Estrategias de marketing**

#### ***10.5.1. Estrategia de retención***

La retención de clientes es vital para hacer crecer la propuesta. No sólo porque atender a un cliente existente es más fácil que visitar a uno nuevo, sino porque los clientes satisfechos es probable que se conviertan en embajadores de la marca, recomendando el/los producto(s) o servicio(s) a familiares y amigos. Los usuarios aumentan su gasto a un ritmo creciente, compran con un margen completo en lugar de con descuento precios y crean eficiencias operativas. Para lograr este objetivo se pretende cumplir las siguientes recomendaciones:

- Continuas actualizaciones y lanzamiento de nuevas versiones: es una forma de que los usuarios sepan que se está trabajando para ayudarlos a alcanzar sus objetivos. Se puede realizar invitaciones (notificaciones) a realizar comentarios constructivos sobre el software.
- Establecimiento de las expectativas del cliente: Esto sucede cuando al usuario se le promete algo que el producto no puede cumplir. Para lograr este objetivo hay que ser muy claro en el mensaje sobre cómo agrega valor para los usuarios.

- Personalización de los mensajes y las ofertas: es necesario realizar una segmentación de los usuarios en función de sus hábitos, esto ayuda a diseñar campañas de marketing de retención en función de cada grupo de usuarios. Esto se puede hacer a través de los datos de sus últimas consultas realizando una categorización.

## **10.6. Métodos de Distribución**

Si bien la calidad del producto juega un papel importante en la aceptación del usuario, la estrategia general de marketing y distribución tiene el mayor impacto al final del día. Por ello se ha optado por las siguientes estrategias:

### ***10.6.1. Marketing de contenidos y redes sociales***

El objetivo del marketing de contenidos es atraer y retener audiencias claramente definidas y clientes rentables. Para ello uno de los planes es la creación del “IngEverSoft”, una extensión del proyecto encargado de ofrecer datos de manera abierta para la comunidad ya que en un futuro se pretende atraer usuarios interesados en el análisis de datos. Por otro lado, la estrategia a tomar por parte de las redes sociales es realizando contenido interesante que pueda atraer a posibles usuarios utilizando canales como Tik Tok, Facebook, Twitter, Instagram y YouTube.

### ***10.6.2. Optimización de motores de búsqueda***

Es el proceso de mejorar la calidad y la cantidad de tráfico del sitio web a un sitio web o una página web desde los motores de búsqueda. Se apunta al tráfico no pagado (“orgánico”) en lugar del tráfico directo o pago. El tráfico no pagado puede provenir de diferentes tipos de búsquedas, incluida la búsqueda de imágenes, la búsqueda de videos, la búsqueda académica y búsquedas de noticias.

### ***10.6.3. Advertising***

Se pretende realizar la utilización de Google Adsense para la publicidad en distintos sitios

web con el fin de entregar mensajes de marketing promocionales a los consumidores. Utilizando como herramientas de recopilación de información demográfica y de comportamiento a Google Analytics y Facebook Insights.

### **10.7. Medida de éxito**

Entre las medidas de Éxito que se tomarán en cuenta se tiene las siguientes:

- Sitio web: Se analiza cuántas personas han hecho clic en la página web, y si este número varió luego de colocar anuncios o la realización de algún tipo de campaña publicitaria.
- La satisfacción del cliente: se tomarán de igual manera las reseñas realizadas por los usuarios antes y después de las estrategias realizadas, obteniendo los datos de redes sociales.
- Aplicación Móvil: Los promotores tendrán un apartado de reclamos, quejas o sugerencias para poder co ayudar en su trabajo que desempeñan ya sea de forma individual o grupal.

**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
**FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y**  
**TELECOMUNICACIONES**



# **11. Despliegue**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

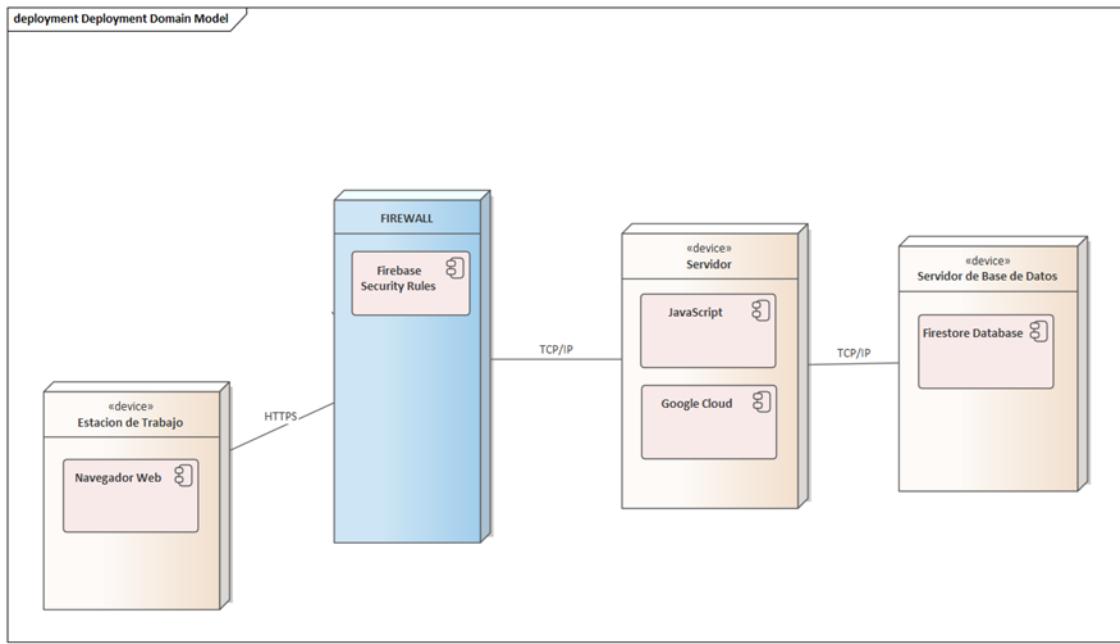
**Integrantes:**

- |                                       |           |
|---------------------------------------|-----------|
| <b>1.</b> Machuca Yavita Miguel Angel | 216063558 |
| <b>2.</b> Torrez Garcia Jorge Luis    | 218137761 |

**Gestion II-2024 Santa Cruz – Bolivia**

## 11.1. Cloud Computing

Para el despliegue, se puede verificar en el siguiente diagrama:



## 11.2. Tipos de Licencia

### 11.2.1. Open Source

Todos los proyectos que estén clasificados dentro de esta categoría utilizarán la Licencia MIT. Para obtener más detalles y conocer a fondo esta licencia, puedes consultar la referencia en el siguiente enlace: [Licencia MIT](#).

## 11.3. Términos y condiciones

### Términos y Condiciones de Uso

Bienvenido a nuestros servicios en línea ("Servicio"), operados por [Nombre de la Compañía] ("Nosotros", "Nuestra Compañía" o "Nosotros"). Al acceder o utilizar nuestro Servicio, usted ("Usuario" o "Usted") acepta estar sujeto a los siguientes términos y condiciones

("Términos"). Lea detenidamente esta sección antes de acceder o utilizar nuestro Servicio. Si no está de acuerdo con alguno de estos términos, absténgase de usar el Servicio de cualquier manera.

## **Aceptación de Términos**

Al acceder o utilizar nuestro Servicio, usted reconoce haber leído, comprendido y aceptado estos Términos, así como nuestro Aviso de Privacidad disponible, y se compromete a cumplir con todas las leyes y regulaciones aplicables. Estos Términos constituyen un contrato legalmente vinculante entre usted y nuestra compañía.

## **Información Veraz y Consentimiento para Comunicaciones**

Usted declara que toda la información proporcionada a través del Servicio es veraz y precisa. La provisión de información falsa o fraudulenta está estrictamente prohibida. Además, acepta recibir nuestras comunicaciones electrónicamente, las cuales pueden incluir correos electrónicos, notificaciones automáticas o mensajes en el Sitio. Puede revocar su consentimiento para recibir comunicaciones específicas en cualquier momento.

## **Funcionalidad del Servicio**

Nuestra compañía proporciona una plataforma web que permite a los usuarios gestionar, controlar y seguir a los usuarios registrados como vendedores. Esto puede incluir información relacionada con clientes, datos personales, ubicaciones exactas y otros detalles.

**Importante:** Reconoce y acepta que nuestra compañía solo es responsable de la información que posee y no será responsable de ningún daño o pérdida derivado del uso de cualquier bien, servicio o contenido disponible a través del Servicio.

## **Restricciones de Uso**

Para garantizar un uso adecuado del Servicio, se prohíben ciertas conductas, como:

- Copiar, modificar, adaptar o realizar ingeniería inversa del Servicio.

- Utilizar el Servicio para spam, correo no deseado o conducta fraudulenta.
- Interferir con la privacidad de otros usuarios o terceros, incluyendo la recolección de información personal sin consentimiento.

## **Derechos de Propiedad Intelectual**

El Servicio y todos los derechos de propiedad intelectual asociados son propiedad exclusiva de nuestra compañía. Esto incluye diseños, logos, algoritmos, derechos de autor y más. Cualquier uso no autorizado está estrictamente prohibido.

## **Cambios y Terminación del Servicio**

Nos reservamos el derecho de modificar, interrumpir o cambiar el Servicio en cualquier momento sin previo aviso. Reconoce que el contenido del Servicio puede ser modificado, ampliado o eliminado sin previo aviso. Nosotros no seremos responsables por dichos cambios o interrupciones.

## **Terminación del Servicio**

Nos reservamos el derecho de bloquear, limitar, suspender o cancelar su acceso al Servicio si consideramos que ha violado estos Términos.

## **Contacto**

Si tiene alguna pregunta sobre estos Términos, contáctenos a través de <https://ingeversoft.com/> . / jorgeluistorrez631@gmail.com

## **11.4 Políticas y privacidad**

### ***Política de Privacidad***

#### **Introducción**

Nuestra compañía se compromete a respetar su privacidad. Esta Política de Privacidad explica cómo recopilamos, usamos, divulgamos, retenemos y protegemos su información personal cuando utiliza nuestros servicios.

## Información Recopilada

Recopilamos la siguiente información personal durante el registro de vendedores y clientes:

- Nombre
- Correo electrónico

Esta información se utiliza exclusivamente para registrar a los usuarios en nuestra base de datos y no se utiliza para ningún otro propósito sin el consentimiento explícito del usuario.

## Uso de su Información

### **Base legal para el uso de la información:**

**Acuerdo:** Recopilamos y procesamos su información personal para proporcionarle nuestros servicios después de su aceptación de estos términos. Esto incluye el registro en nuestra base de datos.

**Consentimiento:** Solicitamos su consentimiento para procesar su información para fines adicionales específicos. Tiene el derecho de retirar su consentimiento en cualquier momento. Por ejemplo, solicitamos su consentimiento para proporcionar contenido personalizado mediante el uso de cookies.

## Seguridad de la Información

Nos preocupamos por la seguridad de la información personal de nuestros usuarios y tomamos medidas para mantenerla segura. Implementamos procedimientos y políticas estándar de la industria para garantizar la seguridad de dicha información y prevenir el acceso no autorizado, la divulgación o la alteración de la misma.

## Cambios en la Política de Privacidad

Nos reservamos el derecho de actualizar nuestra Política de Privacidad en cualquier momento. Cualquier cambio será publicado en esta página. Le recomendamos que revise

periódicamente esta Política para estar informado sobre cómo protegemos la información que recopilamos.

#### Contacto

Si tiene preguntas o inquietudes sobre nuestra Política de Privacidad, no dude en ponerse en contacto con nosotros a través de <https://ingeversoft.com/>.

**UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO**  
FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACION Y  
TELECOMUNICACIONES



## **12. SOFTWARE TERMINADO EN CALIDAD DE PRODUCTO**

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Integrantes:**

- |                                |           |
|--------------------------------|-----------|
| 1. Machuca Yavita Miguel Angel | 216063558 |
| 2. Torrez Garcia Jorge Luis    | 218137761 |

## **12.1. Software en Calidad de Producción**

Nuestro software de gestión empresarial, al igual que la landing page asociada, se encuentra completamente implementado y en calidad de producción. Ambos productos cumplen con los estándares de calidad necesarios para garantizar su rendimiento y eficiencia en un entorno operativo real.

### ***12.1.1 Puntos Clave de Calidad del Software***

A continuación, se detallan los siete aspectos fundamentales que aseguran la calidad del software desarrollado:

#### **Funcionalidad**

**Descripción:** El software de gestión empresarial cumple con todas las funciones necesarias para la administración eficiente de los procesos de negocio, como la gestión de clientes, inventario, facturación y la generación de informes.

**Ejemplo:** Los usuarios pueden agregar, editar y eliminar registros de clientes de manera sencilla. Además, el sistema realiza cálculos automatizados y muestra los resultados de forma clara en el dashboard.

#### **Fiabilidad**

**Descripción:** El software demuestra estabilidad y consistencia operativa en diversos entornos y durante el uso continuo, garantizando una experiencia de usuario sin interrupciones ni fallos.

**Ejemplo:** El sistema ha pasado por rigurosas pruebas de estabilidad, y no se han reportado errores críticos o fallos inesperados en su funcionamiento diario.

## **Usabilidad**

**Descripción:** La interfaz de usuario está diseñada para que los usuarios puedan aprender a utilizar el software de manera eficiente, con una experiencia general satisfactoria.

**Ejemplo:** La interfaz es intuitiva, con menús organizados y una navegación fluida. Además, se ofrece ayuda contextual para las funciones clave, lo que reduce significativamente el tiempo necesario para la capacitación de nuevos usuarios.

## **Eficiencia**

**Descripción:** El software realiza todas sus funciones de manera eficiente, optimizando el uso de recursos y garantizando un alto rendimiento en diversas condiciones.

**Ejemplo:** Las consultas a la base de datos están optimizadas para minimizar los tiempos de respuesta, y el sistema es capaz de manejar grandes volúmenes de datos sin experimentar ralentizaciones.

## **Mantenibilidad**

**Descripción:** El software es fácil de mantener y actualizar, lo que permite realizar mejoras continuas sin afectar negativamente su rendimiento o estabilidad.

**Ejemplo:** El código fuente sigue las mejores prácticas de desarrollo y se han implementado pruebas automáticas que facilitan la identificación y corrección de errores de forma rápida y eficiente.

## **Portabilidad**

**Descripción:** El software puede ser instalado y ejecutado en diferentes sistemas operativos y plataformas, garantizando su accesibilidad en una variedad de dispositivos.

**Ejemplo:** Al ser una aplicación web, el software es accesible desde cualquier computadora con un navegador web, sin necesidad de configuraciones adicionales en el dispositivo.

## **Adaptabilidad**

**Descripción:** El software está diseñado para adaptarse a cambios en el entorno o en los requisitos del negocio, manteniendo su funcionalidad sin necesidad de rediseños costosos.

**Ejemplo:** Se han creado módulos independientes que facilitan la incorporación de nuevas funciones o la modificación de las existentes. Además, la arquitectura del sistema permite una integración sencilla con otras herramientas y sistemas externos.

## **ANEXOS**

 <p>MINISTERIO DE DESARROLLO PRODUCTIVO Y ECONOMÍA PLURAL ESTADO PLURINACIONAL DE BOLIVIA</p> <p><b>SENAPI</b> Servicio Nacional de Propiedad Intelectual</p>	<p>Número de Solicitud N° _____</p> <p>Fecha de presentación 14 07 21</p> <p style="background-color: black; color: white; padding: 5px;"><b>REGISTRO DE PROGRAMAS DE COMPUTACIÓN</b></p>
<p><b>DATOS DE AUTOR O AUTORES:</b></p> <p>a) Nombre: <u>PABLO ANDRES AYAVIRI BONIFACIO</u> Seudónimo: _____</p> <p>Doc. de Identificación <u>8068538</u> Nacionalidad: <u>BOLIVIANO</u></p> <p>Fecha nacimiento: <u>28 DE DICIEMBRE 2021</u> Fecha defunción: <u>--/-/-</u></p> <p>Dirección: <u>C/ SAN MARTIN B/ NOR LIPEZ - CASA #13</u> Ciudad: <u>SANTA CRUZ</u> País: <u>BOLIVIA</u></p> <p>b) Nombre: _____ Seudónimo: _____</p> <p>Doc. de Identificación _____ Nacionalidad: _____</p> <p>Fecha nacimiento: _____ Fecha defunción: _____</p> <p>Dirección: _____ Ciudad: _____ País: _____</p> <p><b>DATOS TITULAR</b></p> <p>Nombre: _____</p> <p>Documento de Identificación _____ Nacionalidad: _____</p> <p>Dirección: _____ Ciudad: _____ País: _____</p> <p>Representante legal: _____ Testimonio: _____</p> <p><b>DATOS DE LA OBRA</b></p> <p>Título: <u>PROGRAMAS DE COMPUTACION</u></p> <p>Año de creación: <u>12/05/2017</u> Individual: <input checked="" type="checkbox"/> Seudónima: <input type="checkbox"/> Por encargo: <input type="checkbox"/>      Duración: <u>--/-/-</u> En colaboración: <input type="checkbox"/> Inédita: <input type="checkbox"/> Original: <input type="checkbox"/>      Colectiva: <input type="checkbox"/> Publicada: <input type="checkbox"/> Derivada: <input type="checkbox"/>      Anónima: <input type="checkbox"/> Póstuma: <input type="checkbox"/> En depósito: <input type="checkbox"/>      Otra: _____</p> <p><b>BREVE DESCRIPCION DE FUNCIONES:</b> _____</p> <p><b>ELEMENTOS DEL SOPORTE LOGICO APORTADOS:</b></p> <p>a. Programa de computador <input checked="" type="checkbox"/> b. Descripción del programa <input type="checkbox"/> c. Material Auxiliar <input type="checkbox"/></p> <p>Observaciones _____</p> <p><b>DATOS DEL SOLICITANTE</b></p> <p>a) Nombre: <u>PABLO ANDRES AYAVIRI BONIFACIO</u></p> <p>Documento de Identificación <u>8068538</u> Nacionalidad: <u>BOLIVIANO</u></p> <p>Dirección: <u>C/ SAN MARTIN B/ NOR LIPEZ - CASA # 13</u> Ciudad: <u>SANTA CRUZ</u> País: <u>BOLIVIA</u></p> <p>Transferencias: _____</p> <p><b>DOMICILIO PARA EFECTOS DE NOTIFICACIÓN</b> (Domicilio que se encuentre dentro de la jurisdicción del SENAPI, solo en las ciudades de La Paz, El Alto, Cochabamba y Santa Cruz.)      Departamento: <u>Santa Cruz</u> Tel/Fel: <u>6633449261</u></p> <p><b>NOTA:</b> En caso de no llenar todos los campos <b>especificados</b> en el formulario será punto de observación a la hora de entregar dicho Documento</p> <p>Documentación que se adjunta: _____</p> <p style="text-align: right; margin-top: 20px;">           Firma del solicitante     </p>	

By Torrez



Facultad de Ingeniería en  
Ciencias de la Computación y Telecomunicaciones  
U.A.G.R.M.

Somos ingeniería!



Software  
Conversación  
CON Taller  
de Sistemas

**"SOFTWARE DE ECOMMERCE PARA OFRECER A LOS  
CLIENTES PRODUCTOS CON REALIDAD AUMENTADA Y PROBADOR  
VIRTUAL DE ROPA MEDIANTE IA"**

Integrantes

Grupo 11

TORREZ GARCIA JORGE LUIS  
MACHUCA YAVITA MIGUEL ANGEL

218137761  
216063558

Materia: INGENIERÍA DE SOFTWARE I

Docente: ING. MARTINEZ CANEDO ROLANDO ANTONI

Gestión I - 2024  
Santa Cruz De La Sierra – Bolivia