UNIVERSIDAD AUTÓNOMA GABRIEL RENE MORENO

FACULTAD DE INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES



**Software de gestión y seguimiento de mercaderías publicitarios**

**GRUPO #11**

**Integrantes:**

**1.** Machuca Yavita Miguel Angel 216063558

**2.** Torrez Garcia Jorge Luis 218137761

**Docente:** Ing. Rolando Martínez

**Materia:** Ingeniería de Software 2

**Fecha:** 07 de diciembre de 2023

Santa Cruz de la Sierra - Bolivia

Contenido

[1. PAPS 7](#_Toc152827510)

[1.1. Ámbito del proyecto 8](#_Toc152827511)

[1.1.1. Introducción 8](#_Toc152827512)

[1.1.2. Antecedentes 8](#_Toc152827513)

[Objetivos 10](#_Toc152827514)

[Objetivo general 10](#_Toc152827515)

[Objetivos específicos 10](#_Toc152827516)

[Requerimientos principales 11](#_Toc152827517)

[Rendimiento 11](#_Toc152827518)

[Fiabilidad 11](#_Toc152827519)

[Interfaces externas 11](#_Toc152827520)

[1.2. Métricas del software 12](#_Toc152827521)

[Métricas orientadas al tamaño 12](#_Toc152827522)

[Métricas orientadas a la función 13](#_Toc152827523)

[Google encontrar mi dispositivo 13](#_Toc152827524)

[GURTAM 15](#_Toc152827525)

[Driv.in 17](#_Toc152827526)

[1.3. Estimaciones del proyecto 18](#_Toc152827527)

[Líneas de código 18](#_Toc152827528)

[Complejidad del proyecto 19](#_Toc152827529)

[Estructuración del cliente 19](#_Toc152827530)

[Interfaces externas 19](#_Toc152827531)

[1.4. Estimaciones del Software 20](#_Toc152827532)

[1.5. Organización de equipos de desarrollo 21](#_Toc152827533)

[Descentralizada Democrática 21](#_Toc152827534)

[Roles 21](#_Toc152827535)

[Responsabilidades 22](#_Toc152827536)

[RTF 23](#_Toc152827537)

[1.6. Gestión de riesgos 23](#_Toc152827538)

[Lista de riesgos 23](#_Toc152827539)

[1.7. Recursos del proyecto 25](#_Toc152827540)

[Tabla de recursos 25](#_Toc152827541)

[1.8. Planificación del tiempo 26](#_Toc152827542)

[Asignación De Tiempo 26](#_Toc152827543)

[Diagrama De Gantt 27](#_Toc152827544)

[2. Modelo de Desarrollo de Software 29](#_Toc152827545)

[2.1. Sprint 3 30](#_Toc152827546)

[2.1.1. Requisitos Iniciales 30](#_Toc152827547)

[2.1.2. Equipos de Desarrollo 30](#_Toc152827548)

[2.1.3. Definición del Patrón de Desarrollo 31](#_Toc152827549)

[2.1.4. Duración del Sprint 31](#_Toc152827550)

[2.1.5. Herramientas de Gestión de Proyecto 31](#_Toc152827551)

[2.1.6. Modelo de contexto inicial 32](#_Toc152827552)

[2.1.7. Modelo de Arquitectura Inicial 32](#_Toc152827553)

[2.1.8. Base de datos inicial 33](#_Toc152827554)

[2.1.9. Producto Backlog Inicial 33](#_Toc152827555)

[Historias de Usuario 35](#_Toc152827556)

[Contexto del Sistema 36](#_Toc152827557)

[Sprint backlog 37](#_Toc152827558)

[Diseño 38](#_Toc152827559)

[3. Manual de Calidad (SQAP) 40](#_Toc152827560)

[3.1. Propósito 41](#_Toc152827561)

[3.2. Misión 41](#_Toc152827562)

[3.3. Visión 41](#_Toc152827563)

[3.4. Estándares 42](#_Toc152827564)

[3.5. Valores 42](#_Toc152827565)

[3.6. Organigrama 43](#_Toc152827566)

[3.7. Gestión 43](#_Toc152827567)

[3.7.1. Organización 43](#_Toc152827568)

[3.7.2. Actividades 43](#_Toc152827569)

[3.7.3. Responsables 46](#_Toc152827570)

[3.7.4. Evolución del SQAP 50](#_Toc152827571)

[3.8. Documentación 51](#_Toc152827572)

[3.8.1. Propósito 51](#_Toc152827573)

[3.8.2. Documentación mínima requerida 52](#_Toc152827574)

[3.9. Estándares, prácticas, convenciones y métricas 56](#_Toc152827575)

[3.9.1. Estándar de documentación 56](#_Toc152827576)

[3.9.2. Estándar de verificación y prácticas 57](#_Toc152827577)

[3.9.3. Estándar de codificación 57](#_Toc152827578)

[3.9.4. Estándares de comentarios 60](#_Toc152827579)

[3.9.5. Responsable de verificar el rendimiento 60](#_Toc152827580)

[3.10. Revisiones y auditorías 60](#_Toc152827581)

[3.10.1. Objetivo 60](#_Toc152827582)

[3.10.2. Requerimientos mínimos 60](#_Toc152827583)

[3.10.3. Auditorias administrativas 63](#_Toc152827584)

[3.11. Verificación 63](#_Toc152827585)

[3.12. Reporte de problemas y acciones correctivas 63](#_Toc152827586)

[3.12.1. Informe de Verificación y Validación 63](#_Toc152827587)

[3.12.2. Reporte sumario de fase V&V 64](#_Toc152827588)

[Reporte de Anomalías 66](#_Toc152827589)

[Reporte Final de V&V 67](#_Toc152827590)

[3.13. Herramientas, técnicas y metodologías 68](#_Toc152827591)

[3.13.1. Herramientas 68](#_Toc152827592)

[3.13.2. Técnicas 68](#_Toc152827593)

[3.13.3. Metodología 68](#_Toc152827594)

[3.14. Gestión de riesgos 68](#_Toc152827595)

[3.15. Referencia Bibliográfica 68](#_Toc152827596)

[4. Productividad basada en Herramientas CASE 70](#_Toc152827597)

[4.1. Enterprise Architect – Herramienta de diseño UML 71](#_Toc152827598)

[4.1.1. Características de Enterprise Architect 73](#_Toc152827599)

[Soporta los 13 diagramas UML 2.1 73](#_Toc152827600)

[Interfaz de usuario intuitiva 73](#_Toc152827601)

[Soporte para Transformaciones MDA 73](#_Toc152827602)

[Documentación flexible y comprensible. 74](#_Toc152827603)

[Salida en formato de texto enriquecido 74](#_Toc152827604)

[Ingeniería de Código Directa e Inversa 74](#_Toc152827605)

[4.2. Balsamiq Mockup 75](#_Toc152827606)

[5. Aspectos Legales 76](#_Toc152827607)

[5.1. Apertura de una empresa en Santa Cruz 77](#_Toc152827608)

[5.1.1. FUNDAEMPRESA 77](#_Toc152827609)

[5.1.2. NIT 78](#_Toc152827610)

[5.1.3. Gobierno Municipal 79](#_Toc152827611)

[5.1.4. Caja Nacional de Salud 79](#_Toc152827612)

[5.1.5. AFPs 79](#_Toc152827613)

[5.1.6. Ministerio de Trabajo 81](#_Toc152827614)

[5.2. Derecho de Autor 81](#_Toc152827615)

[5.2.1. SENAPI 81](#_Toc152827616)

[6. Infraestructura para la Producción del Software 83](#_Toc152827617)

[6.1. Gestión de Configuración de Software 84](#_Toc152827618)

[6.1.1. Introducción 84](#_Toc152827619)

[6.1.2. Alcance 84](#_Toc152827620)

[6.1.3. Objetivos 84](#_Toc152827621)

[6.1.4. Líneas Base 85](#_Toc152827622)

[6.1.5. Tareas de la configuración del software 85](#_Toc152827623)

[6.2. Herramientas para el Control de Versiones 86](#_Toc152827624)

[6.2.1. Herramienta de control de versiones 87](#_Toc152827625)

[6.3. Herramientas utilizadas en el proyecto 88](#_Toc152827626)

[6.3.1. GitHub 88](#_Toc152827627)

[7. Desarrollo del Sitio Web 90](#_Toc152827628)

[7.1. Inicio 91](#_Toc152827629)

[7.2. Servicios 92](#_Toc152827630)

[7.3. Nosotros 93](#_Toc152827631)

[7.4. Contactos 94](#_Toc152827632)

[7.5. Chatbot 95](#_Toc152827633)

[8. Estudio de Mercado 97](#_Toc152827634)

[8.1. Área del mercado 98](#_Toc152827635)

[8.1.2. Extensión Geográfica 98](#_Toc152827636)

[8.1.3. Población 98](#_Toc152827637)

[8.1.3. Ingresos 98](#_Toc152827638)

[8.2. Estudio de la demanda 98](#_Toc152827639)

[8.3. Estudio de la oferta 99](#_Toc152827640)

[8.4. Monetización 99](#_Toc152827641)

[9. Pruebas en el Software 100](#_Toc152827642)

[9.2. Pruebas de Rendimiento 101](#_Toc152827643)

[9.2. Pruebas de Vulnerabilidad 101](#_Toc152827644)

[9.3. Pruebas en el entorno Móvil 102](#_Toc152827645)

[9.3.1. Pruebas de vulnerabilidad 102](#_Toc152827646)

[10. Marketing 104](#_Toc152827647)

[10.1. Descripción 105](#_Toc152827648)

[10.2. Organización 105](#_Toc152827649)

[10.3. Servicios 106](#_Toc152827650)

[10.4. Perfiles de clientes 106](#_Toc152827651)

[10.5. Estrategias de marketing 106](#_Toc152827652)

[10.5.1. Estrategia de retención 106](#_Toc152827653)

[10.6. Métodos de Distribución 107](#_Toc152827654)

[10.6.1. Marketing de contenidos y redes sociales 107](#_Toc152827655)

[10.6.2. Optimización de motores de búsqueda 107](#_Toc152827656)

[10.6.3. Advertising 108](#_Toc152827657)

[10.7. Medida de éxito 108](#_Toc152827658)

[11. Despliegue 109](#_Toc152827659)

[11.1. Cloud Computing 110](#_Toc152827660)

[11.2. Tipos de Licencia 110](#_Toc152827661)

[11.2.1. Open Source 110](#_Toc152827662)

[11.3. Términos y condiciones 110](#_Toc152827663)

[11.4 Políticas y privacidad 112](#_Toc152827664)

[Política de Privacidad 112](#_Toc152827665)

[ANEXOS 115](#_Toc152827666)

[CARATULA SQAP 116](#_Toc152827667)

[Caratula de TALLER DE GRADO 117](#_Toc152827668)

# PAPS

## Ámbito del proyecto

### 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.

### Antecedentes



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 meticuloso 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**

**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

* Analizar la situación del negocio y comprender sus necesidades.
* Analizar los requerimientos del software.
* Definir los requisitos funcionales y no funcionales del software a través de los requerimientos
* Definir las estrategias para el funcionamiento de los módulos software.
* Analizar las opciones tecnológicas existentes para la implementación.
* Diseñar la base de datos.
* Diseñar interfaces intuitivas para el usuario.
* Desarrollo e implementación de los módulos del software
* Realizar pruebas al software.

### Requerimientos principales

**RP1:** Registrar usuarios, tanto clientes como empleados de la empresa.

**RP2:** 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.

**RP3:** Generar y hacer lectura código qr para la verificación de la visita de negocios del empleado.

**RP4:** 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.

### 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

Se realizarán las pruebas necesarias para garantizar que el software funcione de forma correcta siempre.

### Interfaces externas

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

## Métricas del software

### Métricas orientadas al tamaño

Productividad = KLDC/persona-mes

Calidad = errores/KLDC

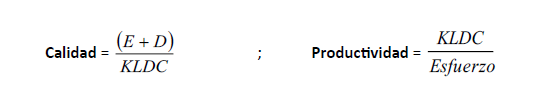
Documentación = págs. Doc. / KLDC

Costo = $/KLDC

Esfuerzo= Gente x Tiempo

**LINEAS DE CODIGO:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |



|  |  |  |
| --- | --- | --- |
| **PROYECTO** | **CALIDAD** | **PRODUCTIVIDAD** |
| **Google encontrar mi**  **dispositivo** | 3.5 | 0.3333 |
| **GURTAM** | 0.986 | 0.316 |
| **Driv.in** | 0.714 | 0.777 |

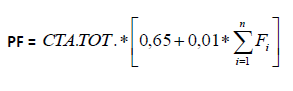
### Métricas orientadas a la función

### Google encontrar mi dispositivo



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Factor de peso** |  |  |  |
| **Parámetros de medición** | **Cuenta** | **Simple** | **Medio** | **Complejo** | **Total** |
| **N° de entradas de usuario** | 1 | 3 | 4 | 6 | 3 |
| **N° de salidas de usuario** | 0 | 4 | 5 | 7 | 0 |
| **N° de peticiones** | 20 | 3 | 4 | 6 | 80 |
| **N° de archivos** | 5 | 7 | 10 | 15 | 50 |
| **N° de interfaces externas** | 0 | 5 | 0 | 0 | 0 |
|  |  |  |  | **Cta. Total** | 133 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **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 múltiples pantallas u operaciones? | X |  |  |  |  |  | 0 |
| 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 |  |  |  | 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 |



**PF =** 133 \* [0,65 + 0,01 \* 40]

PF = 140

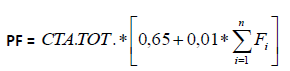
La plataforma “Google encontrar mi dispositivo” cuenta con 140 puntos de función.

### GURTAM



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Factor de Peso** | | | | |  |
| **Parámetros de**  **Medición** | **Cuenta** | **Simple** | **Medio** | **Complejo** | **Total** |
| **N° de entradas de**  **usuario** | 3 | 3 | 4 | 6 | 9 |
| **N° de salidas de**  **usuario** | 1 | 4 | 5 | 7 | 4 |
| **N° de peticiones**  **de usuario** | 15 | 3 | 4 | 6 | 60 |
| **N° de archivos** | 4 | 7 | 10 | 15 | 40 |
| **N° de interfaces**  **externas** | 0 | 5 | 7 | 10 | 0 |
|  | | | | | **113** |  | **Cta.**  **Total** | **58** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 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 | 5 |
|  | | | | | | | | **49** |

****

**PF =** 113 \* [0,65 + 0,01 \* 49]

**PF =** 129

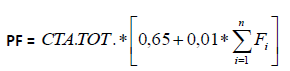
La plataforma “GURTAM” cuenta con 129 puntos de función.

### Driv.in



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Factor de Peso** | | | | |
| **Parámetros de Medición** | **Cuenta** | **Simple** | **Medio** | **Complejo** | **Total** |
| N° de entradas de usuario | 2 | 3 | 4 | 6 | 6 |
| N° de salidas de usuario | 1 | 4 | 5 | 7 | 4 |
| N° de peticiones de usuario | 10 | 3 | 4 | 6 | 40 |
| N° de archivos | 5 | 7 | 10 | 15 | 50 |
| N° de interfaces externas | 0 | 5 | 7 | 10 | 0 |
|  | | | | | 100 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |
|  | | | | | | | | 46 |

****

**PF =** 100 \* [0,65 + 0,01 \* 46]

**PF =** 111

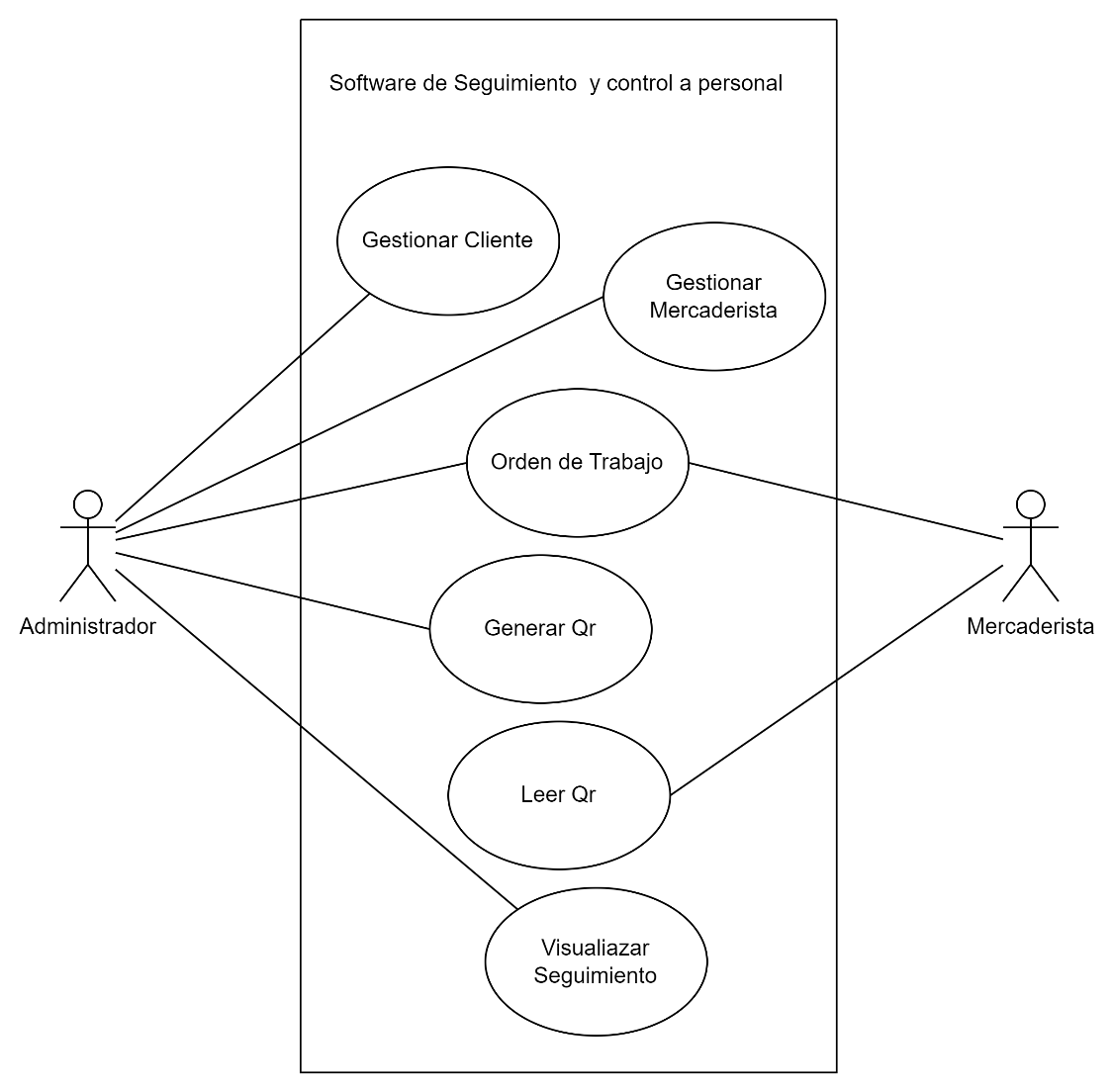
La plataforma “Driv.in” cuenta con 111 puntos de función

## Estimaciones del proyecto

### Líneas de código

Se tienen **12 KLDC**

### Complejidad del proyecto



### 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.

### Interfaces externas

Se hará el uso de Google maps, para poder ver los mapas y realizar la geolocalización de los mercaderistas publicitarios.

## Estimaciones del Software

**KLDC Esperadas** = (Optimista+Mas probable+Pesimista)/3

|  |  |  |  |
| --- | --- | --- | --- |
| **Optimista** | **Más probable** | **Pesimista** | **Esperado** |
| 7 | 10 | 15,2 | 11,73333333 |

**Valor Esperado** = (Optimista+(4\*Mas probable) +Pesimista) /6

8,886

VE

**Estimación de COCOMO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tipo objeto** | **Cuenta** | **Factor** | | | **Total** |
| **Simple** | **Media** | **Avanzado** |
| Pantalla | 15 | 1 | **2** | 3 | 30 |
| Reporte | 3 | **2** | 5 | 8 | 6 |
| Componente 3gl | 8 |  |  | **10** | 80 |
|  | | | | PO | 116 |

**PON** = PO\*[(100-%Reutilización) /100] = 116\*[(100-0.30)/100]

115,652

PON

**Análisis de productividad**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Proporciones de productividad** | **Muy baja** | **Baja** | **Normal** | **Alta** | **Muy alta** |
| Capacidad y experiencia del  desarrollador |  |  | \* |  |  |
| Capacidad y madurez del entorno |  | \* |  |  |  |
| PROD | 4 | 7 | 13 | 25 | 50 |

20

PROD

**ESFUERZO** = PON/PROD = 115.652/20

5.8

E

## Organización de equipos de desarrollo

### Descentralizada Democrática

La estructura de equipo que utilizaremos para el desarrollo del producto será la **Descentralizada Democrática**, ya que la metodología a seguir asigna una tarea a un grupo de trabajo la cual se hace responsable del cumplimiento de esta tarea.

La organización **Descentralizada Democrática** no tiene un jefe permanente, se nombran coordinadores de tareas a corto plazo. La comunicación entre jefe y los miembros es horizontal.

Este tipo de organización se emplea en equipos pequeños y medianos.

### Roles

* **Ever Ochoa Illusco:** Analista, Arquitecto.
* **Miguel Angel Machuca Yavita :** Arquitecto, Desarrollador.

### Responsabilidades

**Analista:**

El analista es alguien que es responsable de entender las necesidades del cliente, y asegurarse de que la solución que está siendo desarrollada se ajusta a esas necesidades.

Las actividades típicas de un analista incluyen la licitación de requisitos, reuniones con clientes y la redacción de especificaciones funcionales.

Incluso si un proyecto es demasiado pequeño para escribir un verdadero documento de especificación, la comprensión de las necesidades del cliente es un trabajo importante, dado que a menudo el éxito de un proyecto de desarrollo depende de qué tan cerca está la solución desarrollada de las expectativas del cliente.

**Arquitecto:**

Al igual que el arquitecto de software, el Arquitecto del Sistema es responsable de pensar el sistema antes de construirlo. Así como el arquitecto de software es responsable para el software, un arquitecto del sistema es responsable del hardware. Muchas aplicaciones ejecutan completamente en un único servidor. Muchos otros sin embargo se ejecutan en grupos de servidores, con servidores dedicados de bases de datos, servidores web y balanceadores de carga. Un arquitecto del sistema tiene en cuenta los requisitos de rendimiento y disponibilidad, el número de usuarios / visitantes, etc. y en base a esto, diseña una infraestructura de servidores y una red.

**Desarrollador**

El desarrollo efectivo de una aplicación es hecho por los desarrolladores del equipo. Pero un desarrollador tiene más responsabilidades que solo escribir código. Él es a menudo responsable de hacer el seguimiento de su propio progreso, e informar al jefe de proyecto de los problemas a los que se enfrenta. Él es también quien implementa las ideas del arquitecto, y como tal, puede tener que discutir las (in)posibilidades de la implementación con el arquitecto.

Una responsabilidad importante es documentar el código. Mientras que muchos desarrolladores piensan que la documentación es algo que será realizado mejor por alguien más, esta es en realidad una responsabilidad importante del desarrollador.

La Documentación de Código tiene como objetivo el explicar a otros desarrolladores aquellas cosas que no resulten evidentes o claras a partir de la lectura del propio código en sí. Se debe dar una idea de por qué un fragmento de código es de la manera que es. El desarrollador es el único que conoce los pensamientos e ideas detrás del código que escribe, lo cual lo convierte en el candidato perfecto para documentarlo.

### RTF

Se realizaron reuniones cada fin de semana para hablar y ver el avance del proyecto y dar opiniones al respecto. Se utilizaron tablas de sprint para tener un seguimiento de las tareas de cada persona y tener una rápida vista de las tareas realizadas y por realizar

## Gestión de riesgos

### Lista de riesgos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Riesgo** | **Probabilidad %** | **Impacto** | **Reducir Probabilidad** | **Reducir Impacto** |
|
| **Riesgos Técnicos** |  | | | |
| Fallas de Hardware | 60 | Crítico | Realizar mantenimiento de las maquinas. Tener máquinas de respuesta. | Utilizar máquinas de compañeros en tiempos de descansos. |
|
|
| Perdida de información | 40 | Medio | Realizar backups cada día de trabajo. Documentar todo lo avanzado. | Revisar la documentación impresa. |
|
| **Riesgos de Gestión** |  | | | |
| Mala estimación de tiempo | 45 | Significativo | Aumentar un porcentaje de error. Tomar en cuenta días festivos en las estimaciones. | Reducir las horas de descanso y aumentar las horas de trabajo. |
|
| Mala estimación de costos | 60 | Medio | Contratar a un experto en costos y presupuestos. | Reutilizar o reciclar instrumentos. |
|
| Baja de un integrante | 65 | Crítico | Dar apoyo moral al grupo de trabajo. Tener buena comunicación | Repartir las tareas del integrante a todo el grupo de trabajo |
|
|
| **Riesgo de negocio** |  | | | |
| Falta de recursos | 40 | Significativo | No utilizar los recursos en gastos innecesarios. Reciclar. | Buscar inversionistas para el desarrollo de este proyecto. |
| Malos contratos de negocios | 40 | Significativo | Definir un buen contrato con ayuda de abogados. Tener buena comunicación con ambas partes para elaborar un contrato claro. | Buscar asesoramiento legal. |
|
|

## Recursos del proyecto

### Tabla de recursos

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Recurso** | **Desde** | **Hasta** | **Tiempo (año)** | **Cantidad** | **Precio Unitario** | **Depreciación** | **Precio Neto** | **Total** |
| **Hardware** | 05/08/2023 | 05/12/2023 |  |  |  |  |  | 0,0 |
| PC(A2) | 05/08/2023 | 05/12/2023 | 0,33 | 2 | 5850,00 | 0,25 |  | 11700,0 |
| Impresora(A3) | 05/08/2023 | 05/12/2023 | 0,33 | 1 | 1144,00 | 0,25 |  | 1144,0 |
| Celulares(A4) | 05/08/2023 | 05/12/2023 | 0,33 | 4 | 1500,00 | 0,25 |  | 6000,0 |
| **Software** | 05/08/2023 | 05/12/2023 |  |  |  |  |  | 0,0 |
| SO(A5) | 05/08/2023 | 05/12/2023 | 0,33 | 1 | 1800,00 | 0,33 |  | 1800,0 |
| Hosting(A6) | 05/08/2023 | 05/12/2023 | 0,33 | 1 | 140,00 | 0,33 |  | 140,0 |
| **Personal** | 05/08/2023 | 05/12/2023 |  |  |  |  |  | 0,0 |
| Analista | 05/08/2023 | 05/12/2023 | 0,33 | 1 |  |  | 7000,0 | 7000,0 |
| Diseñador | 05/08/2023 | 05/12/2023 | 0,33 | 1 |  |  | 5000,0 | 5000,0 |
| Desarrollador | 05/08/2023 | 05/12/2023 | 0,33 | 2 |  |  | 6500,0 | 13000,0 |
| Limpieza | 05/08/2023 | 05/12/2023 | 0,33 | 1 |  |  | 1000,0 | 1000,0 |
| **Infraestructura** | 05/08/2023 | 05/12/2023 |  |  |  |  |  | 0,0 |
| Alquiler Oficina | 05/08/2023 | 05/12/2023 | 0,33 | 1 | 2000,00 |  |  | 2000,0 |
| Escritorios(A8) | 05/08/2023 | 05/12/2023 | 0,33 | 2 | 650,00 | 0,2 |  | 1300,0 |
| Sillas(A9) | 05/08/2023 | 05/12/2023 | 0,33 | 6 | 380,00 | 0,2 |  | 2280,0 |
| **Logística** | 05/08/2023 | 05/12/2023 |  |  |  |  |  | 0,0 |
| Material de escritorio | 05/08/2023 | 05/12/2023 | 0,33 | 1 | 200,00 |  |  | 200,0 |
| Cursos(A1) | 05/08/2023 | 05/12/2023 | 0,33 | 3 | 140,00 |  |  | 420,0 |
| Alimentos | 05/08/2023 | 05/12/2023 | 0,33 | 1 | 1800,00 |  |  | 1800,0 |
| **TOTAL BS.-** | | | | | | | | **54784,00** |

## Planificación del tiempo

### Asignación De Tiempo

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Después de** | **Duración de la Actividad**  **(Días)** |
| A0. Proyecto | - | 50 |
| **A1. Captura de Requisitos** | **-** | **5** |
| A1.1. Identificar Actores y Casos de Uso | - | 1 |
| A1.2. Priorizar Casos de Uso | A1.1. | 1 |
| A1.3. Detallar Casos de Uso | A1.2. | 1 |
| A1.4. Prototipo Interfaz de Usuario | A1.3. | 1 |
| A1.5. Estructurar Modelo de Casos de Uso | A1.4. | 1 |
| **A2. Análisis** | **A1.** | **10** |
| A2.1. Análisis de la Arquitectura | A1. | 2 |
| A2.2. Análisis de Casos de Uso | A2.1. | 3 |
| A2.3. Análisis de Clases | A2.2. | 3 |
| A2.4. Análisis de Paquetes | A2.3. | 2 |
| **A3. Diseño** | **A2.** | **10** |
| A3.1. Diseño de la Arquitectura | A2. | 2 |
| A3.2. Diseño de Casos de Uso | A3.1. | 3 |
| A3.3. Diseño de Clases | A3.2. | 3 |
| A3.4. Diseño del Subsistema | A3.3. | 2 |
| **A4. Implementación** | **A3.** | **15** |
| A4.1. Implementación de la Arquitectura | A3. | 2 |
| A4.2. Integrar el Sistema | A4.1. | 3 |
| A4.3. Implementar un Subsistema | A4.2. | 5 |

|  |  |  |
| --- | --- | --- |
| A4.4. Implementar una Clase | A4.3. | 5 |
| **A5. Prueba** | **A4.** | **10** |
| A5.1. Planificar Prueba | A4. | 2 |
| A5.2. Diseñar Prueba | A5.1. | 2 |
| A5.3. Implementar Prueba | A5.2. | 2 |
| A5.4. Realizar Prueba de Integración | A5.3. | 2 |
| A5.5. Realizar Prueba de Sistema | A5.4. | 2 |

### Diagrama De Gantt

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de tarea** | **Duración** | **Comienzo** | **Fin** |
| **Proyecto** | **50 días** |  |  |
| **1. Captura de requisitos** | **5 días** |  |  |
| 1.1 Actores y CU | 1 día | 10/09/2023 | 10/09/2023 |
| 1.2 Priorizar CU | 1 día | 11/09/2023 | 11/09/2023 |
| 1.3 Detallar CU | 1 día | 12/09/2023 | 12/09/2023 |
| 1.4 Prototipo Interfaz | 1 día | 13/09/2023 | 13/09/2023 |
| 1.5 Estructurar modelo CU | 1 día | 17/09/2023 | 17/09/2023 |
| **2. Análisis** | **10 días** | **18/09/2023** | **01/09/2023** |
| 2.1 Análisis de arquitectura | 2 días | 18/09/2023 | 19/09/2023 |
| 2.2 Análisis de CU | 3 días | 20/09/2023 | 24/09/2023 |
| 2.3 Análisis de clases | 3 días | 25/09/2023 | 27/09/2023 |
| 2.4 Análisis de paquetes | 2 días | 28/09/2023 | 01/09/2023 |
| **3. Diseño** | **10 días** | **02/10/2023** | **15/10/2023** |
| 3.1 Diseño de arquitectura | 2 día | 02/10//2023 | 03/10/2023 |
| 3.2 Diseño de CU | 3 día | 04/10/2023 | 08/10/2023 |
| 3.3 Diseño de clases | 3 día | 09/10/2023 | 11/10/2023 |
| 3.4 Diseño de subsistemas | 2 día | 12/10/2023 | 15/10/2023 |
| **4. Implementación** | **15 días** | **16/10/2023** | **05/10/2023** |
| 4.1 Implementación de Arq. | 2 días | 16/10/2023 | 17/10/2023 |
| 4.2 Integrar el sistema | 3 días | 18/10/2023 | 22/10/2023 |
| 4.3 Implementar un subsistema | 5 días | 23/10/2023 | 29/10/2023 |
| 4.4 Implementar una clase | 5 días | 30/10/2023 | 05/10/2023 |
| **5. Prueba** | **10 días** | **06/11/2023** | **19/11/2023** |
| 5.1 Planificar prueba | 2 días | 06/11/2023 | 07/11/2023 |
| 5.2 Diseñar prueba | 2 días | 08/11/2023 | 09/11/2023 |
| 5.3 Implementar prueba | 2 días | 12/11/2023 | 13/11/2023 |
| 5.4 Realizar prueba de integración | 2 días | 14/11/2023 | 15/11/2023 |
| 5.5 Realizar prueba de sistemas | 2 días | 16/11/2023 | 19/11/2023 |

# Modelo de Desarrollo de Software

## Sprint 3

### 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 registro 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.
* **Modelo Saas:** Permite a nuestros usuarios conectarse a nuestra aplicación en la nube a través de Internet y usarla. Con esto ofrecemos un modelo de pago por uso.

### Equipos de Desarrollo

|  |  |  |
| --- | --- | --- |
| **Persona** | **Rol** | **Característica** |
| Machuca Yavita Miguel Angel | Scrum Master | Hacerse cargo de la realización de todo el Sprint 0 |
| Ochoa Llusco Ever Cristian | Product Owner | Especificar sobre todos los aspectos que requiere el software |

### 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.

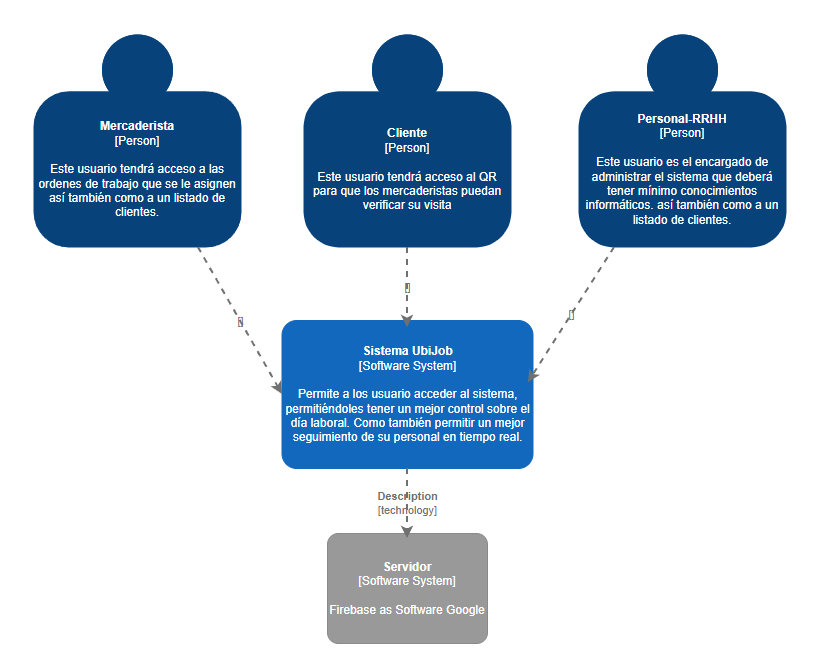
### Duración del Sprint

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

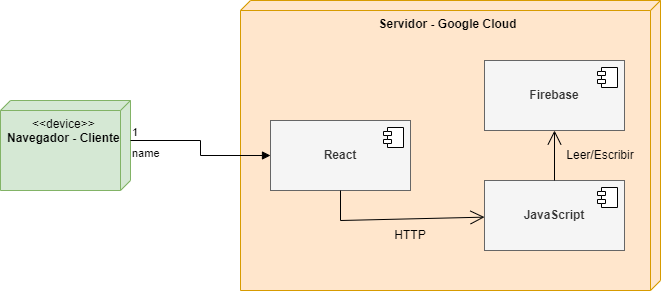
### ***Herramientas de Gestión de Proyecto***

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.

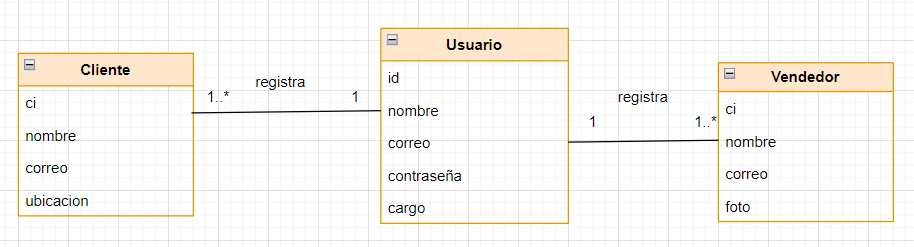
### Modelo de contexto inicial



### Modelo de Arquitectura Inicial



### Base de datos inicial



### Producto Backlog Inicial

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PRODUCT BACKLOG** | | | | | | |
| **Proyecto** | | Software Para El Control Y Seguimiento De Los Reponedores De Una Empresa | | | | |
| **Product Owner** | | Ochoa Llusco Ever Cristian | | | | |
| **Versión** | | v1.0.0 | | | | |
| **Id** | **Rol** | **Funcionalidad** | **Razón/Resultado** | | | **Prioridad** |
| 1 | Análisis | Elaborar historia de usuario para usuario | Definir los requisitos de la hu usuario | | | 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  implementado | sobre | lo | 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  implementado | sobre | lo | Alta |
| **9** | Diseño | Diseñar Interfaz de Cliente | Diseño de interfaz | | | Alta |
| **10** | Desarrollo | Implementar Interfaz de Cliente | Desarrollo e implementación  de los requerimientos. | | | Alta |
| **11** | Desarrollo | Implementar CRUD de cliente | Desarrollo e implementación  de los requerimientos. | | | Alta |
| **12** | Diseño | diseñar interfaz para generar  orden de trabajo | Diseño de interfaz | | | Alta |
| **13** | Desarrollo | Integrar librería para generar QR | Integración de librerías en el  desarrollo del proyecto para generar qr | | | Alta |
| **14** | Desarrollo | Integrar API Maps para visualizar seguimiento | Integración de API en el desarrollo del proyecto para visualizar seguimiento | | | Alta |
| **15** | Diseño | Desplegar Mapa en interfaz  web y móvil | Poder visualizar el mapa en el  sitio web. | | | Alta |
| **16** | Desarrollo | Implementar Rutas de API Rest | Ayudan a reestructurar y dirigir sistemas internos  destinados a dar apoyo. | | | Alta |

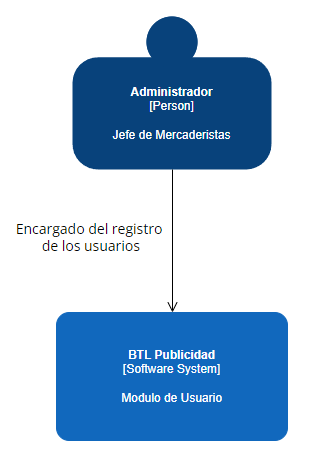
#### Objetivos del Sprint

Diseñar Interfaz, implementar CRUD de usuario

### Historias de Usuario

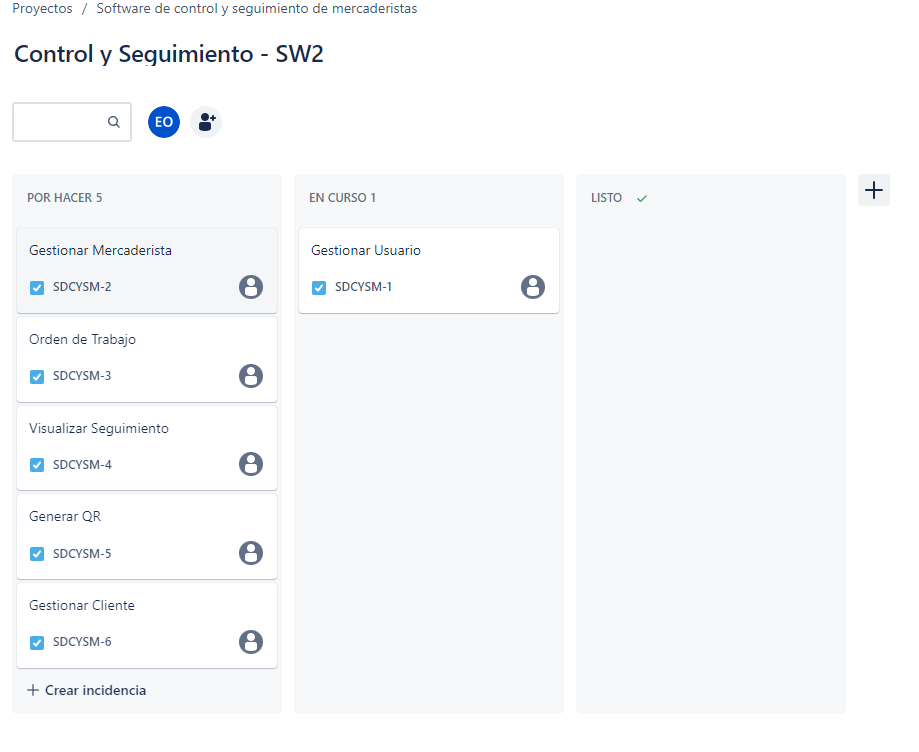
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | |  | | | |
| **HU-Nro.** | **Nombre Corto del HU** | **Modulo** | | | **Estimación** | **Desarrollador** | |
| 1 | Gestionar Usuarios | Inicio | | | 2 día | Ochoa Ever | |
| **Como:** | Análisis | | | | | | |
| **Quiero:** | Poder iniciar sesión con mis credenciales de usuarios | | | | | | |
| **Para:** | Acceder a las funciones personalizadas de la aplicación | | | | | | |
| **Descripción:** | 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. | | | 1. 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** | | | | | | | |
|  | | | | | | | |
| **Fecha Publicada** | **Estado** | **Fecha Terminada** | | | | **Product Owner** | |
| 25/09/2023 | Terminada | 26/09/2023 | | | | Ever Ochoa | |
|  | | | | | | | |

### Contexto del Sistema



### Sprint backlog

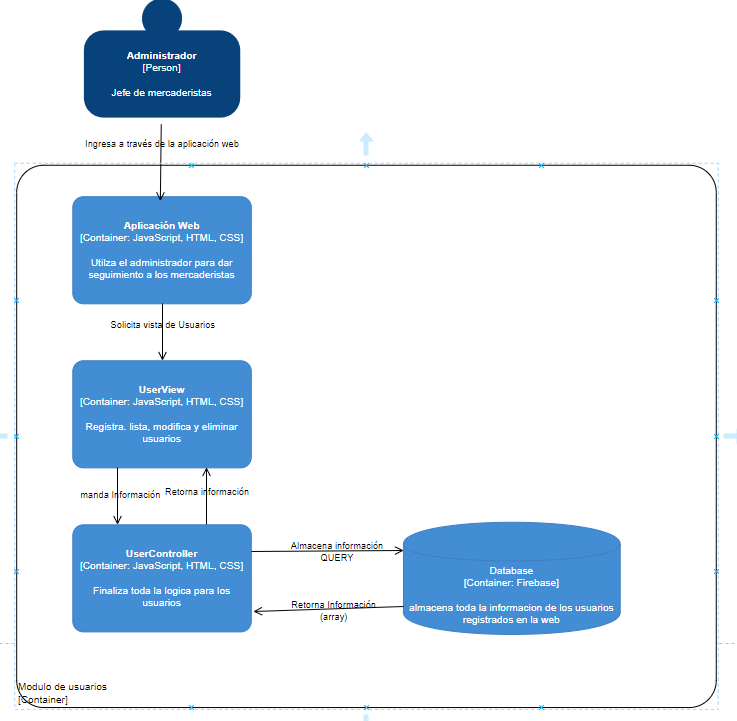
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SPRINT BACKLOG** | | | | | |
| **Numero de Sprint:** | | | **Tiempo Programado:** | | |
| **Objetivo:** Analisis, diseño, implementación y pruebas para la HU1.Usuarios | | | | | |
| **Fecha de inicio:** | | | **Fecha de finalización:** | | |
| **Id** | **Tarea** | **Tipo** | **Estimación** | **Responsable** | **Estado** |
| HU1-1 | Elaborar historia de  usuario para usuario | Análisis | 1 | Ever | Completado |
| HU1-2 | Diseñar interfaz de  usuario para usuario | Diseño | 1 | Ever | Completado |
| HU1-3 | Implementar la historia  de usuario de usuario | Implementación/  Desarrollo | 1 | Ever | Completado |
| HU1-4 | Realizar pruebas a la  implementación | Pruebas | 1 | Ever | Completado |



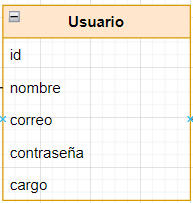
Patrón de desarrollo por Historia de Usuario

### 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 |

# Manual de Calidad (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.

## Misión

Somos una empresa comercializadora de productos y servicios software, con el talento humano comprometido a brindar el soporte necesario.

## Visión

Convertirnos en la empresa de desarrollo de software líder en la región brindando a las empresas, software de alta calidad, Innovando y aplicando Tecnología de vanguardia

## Estándares

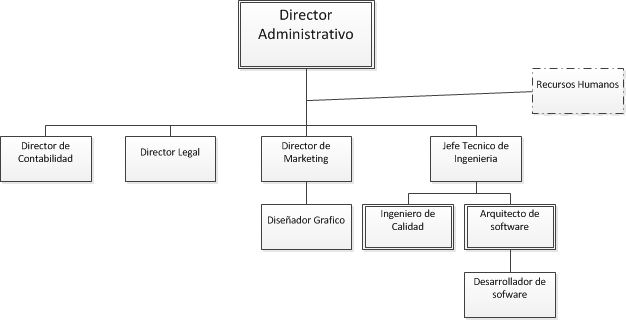
Para alcanzar la calidad total de los productos y la mejora continua, se utilizan los siguientes estándares:

* **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 termino de calidad, eficiencia y costo.
* Nuestros clientes
  + Nuestra razón de ser.
* Nuestros Colaboradores
  + Es la fortaleza de nuestra empresa para lograr nuestra misión.

## Organigrama



## Gestión

### Organización

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

### 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 |

### 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.

**Tabla Nro. 1 Productos y Responsables de Acciones Correctivas**

|  |  |
| --- | --- |
| Producto | Responsable |
| 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 Implantación | Jefe Técnico de Ing. |
| Estándar de Documentación de Usuario | Ing. De Calidad |
| Documentación de Usuario | Ing. De Calidad |
| Plan de Gestión de Configuración | Jefe Tecnico de Ing. |

### 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.

#### 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.

#### 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

#### 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.

#### 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.

## 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

1. adecuación a las necesidades
2. precisión de los resultados
3. interoperabilidad
4. seguridad de los datos

#### Confiabilidad

1. madurez
2. tolerancia a faltas
3. recuperabilidad (Ver si aplica)

#### Usabilidad

* 1. comprensible
  2. aprendible
  3. operable
  4. atractivo

#### Eficiencia

* 1. comportamiento respecto al tiempo (Ver si aplica)
  2. utilización de recursos

#### Mantenibilidad

* 1. analizable
  2. modificable
  3. estable, no se producen efectos inesperados luego de modificaciones
  4. verificable

#### Portabilidad

* 1. adaptable (Ver si aplica)
  2. instalable
  3. co-existencia
  4. 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

#### Plan de Verificación & Validación

El Plan de V & V deberá identificar y describir los métodos a ser utilizados en:

* La verificación de que:
  + los requerimientos descritos en el documento de requerimientos han sido aprobados por una autoridad apropiada. En este caso sería que cumplan con el acuerdo logrado entre el cliente y el equipo.
  + los requerimientos descritos en el documento de requerimientos son implementados en el diseño expresado en el documento de diseño.
  + el diseño expresado en el documento de diseño esta implementado en código.
* Validar que el código, cuando es ejecutado, se adecua a los requerimientos expresados en el documento de requerimientos.

#### Reportes de Verificación & Validación

Estos documentos deben especificar los resultados de la ejecución de los procesos descritos en el Plan de V & V.

#### Documentación de usuario

La documentación de usuario debe especificar y describir los datos y entradas de control requeridos, así como la secuencia de entradas, opciones, limitaciones de programa y otros elementos necesarios para la ejecución exitosa del software.

Todos los errores deben ser identificados y las acciones correctivas descritas.

Como resultado del proyecto el cliente obtendrá una documentación para el usuario de acuerdo a los requerimientos específicos del proyecto.

#### Plan de Gestión de configuración

El Plan de gestión de configuración debe contener métodos para identificar componentes de software, control e implementación de cambios, y registro y reporte del estado de los cambios implementados.

## Estándares, prácticas, convenciones y métricas

### Estándar de documentación

Como estándares de documentación se definirán dos documentos:

* Estándar de documentación técnica
* Estándar de documentación de usuario.

La documentación técnica del producto debe:

* Ser adecuada para que un grupo independiente del de desarrollo pueda encarar el mantenimiento del producto.
* Incluir fuentes, Modelos de Casos de Uso, Objetos

Para la escritura de documentos se han definido plantillas para ser utilizadas en la elaboración de entregables.

En estas plantillas se definen:

* encabezado y pie de página.
* fuente y tamaño de fuente para estilo normal
* fuente y tamaño de fuente para los títulos a utilizar
* datos mínimos que se deben incluir: fecha, versión y responsables.

### Estándar de verificación y prácticas

Se utilizan las prácticas definidas en el Plan de Verificación y Validación.

Como estándar se utiliza el documento de: STD 1012-1986 IEEE Standard for Software Verification and Validation Plans.

### Estándar de codificación

La codificación del software se realizará en la plataforma JavaScript y archivos planos para la persistencia de datos.

Las normas de codificación se definen de la siguiente forma:

**Organización del código:**

El software debe estructurarse en módulos independientes según el diseño establecido.

**Comentarios de Archivo:**

Se deben incluir comentarios al inicio de cada archivo para describir:

/\*

\* Nombre del archivo

\*

\* Información de versión

\*

\* Fecha de creación

\*

\* Aviso de derechos de autor

\*/

**Declaraciones iniciales**

Las declaraciones import y export deben ir al principio del archivo, después de los comentarios iniciales.

**Variables y Funciones:**

Las variables y funciones deben seguir un orden claro y consistente:

Constructores y métodos.

Variables de instancia.

Variables de clase (estáticas).

Implementación de la clase o interfaz.

**Longitud de línea:**

Evitar líneas de código que excedan los 80 caracteres para mantener la legibilidad en diferentes entornos.

Las líneas de documentación deben ser más cortas, preferiblemente de menos de 70 caracteres.

**Expresiones en múltiples líneas:**

Cuando una expresión no quepa en una sola línea, se debe continuar en la siguiente, siguiendo estos principios:

Terminar en una coma (,) si es posible.

Terminar antes de un operador (+, -, \*, /, %).

Alinear la nueva línea al principio de la expresión y al mismo nivel que la línea anterior.

**Comentarios y Documentación:**

Cada módulo debe explicar claramente su función y objetivo.

Todas las variables deben estar debidamente comentadas para explicar su función y propósito.

Cada función debe estar documentada explicando su funcionalidad y los parámetros que recibe.

Los mensajes de error o excepciones deben indicar claramente su origen y la función/procedimiento relacionado.

**Nomenclatura de Variables:**

Utilizar nombres descriptivos para las variables que reflejan su funcionalidad.

Seguir las convenciones de nomenclatura comunes (camelCase, PascalCase) para variables, funciones y clases.

**Comentarios en el Código:**

Utilizar comentarios para explicar el propósito de una acción.

Mantener los comentarios separados del código para facilitar la legibilidad.

Emplear comentarios de bloque (/\* ... \*/) para descripciones largas y comentarios de línea (//) para especificaciones breves.

**Convenciones de Codificación:**

Se utilizarán las convenciones de codificación ampliamente aceptadas por la comunidad de desarrollo de JavaScript.

Estas normas y convenciones deben seguirse para mantener la coherencia y facilitar la comprensión y mantenimiento del código.

**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:

**Funcionalidad**

* Se utilizarán las Convenciones de Codificación en Java de **Sun Microsystems, Inc**

### Estándares 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.
* {} para comentarios de bloque
* // para comentarios de línea

### Responsable de verificar el rendimiento

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

* El jefe del equipo de desarrollo.
* La organización del SQA.

## Revisiones y auditorías

### Objetivo

Definición de las revisiones y auditorías técnicas y de gestión que se realizarán.

Especificación de cómo serán llevadas a cabo dichas revisiones y auditorías.

### Requerimientos mínimos

**Revisión de requerimientos**

Esta revisión se realiza para asegurar que se cumplió con los requerimientos especificados por el Cliente.

**Revisión de diseño preliminar**

Esta revisión se realiza para asegurar la consistencia y suficiencia técnica del diseño preliminar del software.

**Revisión de diseño crítico**

Esta revisión se realiza para asegurar la consistencia del diseño detallado con la especificación de requerimientos.

**Revisión del Plan de Verificación & Validación**

Esta revisión se realiza para asegurar la consistencia y completitud de los métodos especificados en el Plan de V & V.

**Auditoría funcional**

Esta auditoría se realiza previa a la liberación del software, para verificar que todos los requerimientos especificados en el documento de requerimientos fueron cumplidos.

**Auditoría física**

Esta revisión se realiza para verificar que el software y la documentación son consistentes y están aptos para la liberación.

**Auditorías internas al proceso**

Estas auditorías son para verificar la consistencia: del código versus el documento de diseño, especificaciones de interface, implementaciones de diseño versus requerimientos funcionales, requerimientos funcionales versus descripciones de testeo.

**Revisiones de gestión**

Estas revisiones se realizan periódicamente para asegurar la ejecución de todas las actividades identificadas en este Plan. Deben realizarse por una persona ajena al grupo de trabajo (en caso de que sea posible).

**Revisión del Plan de gestión de configuración**

Esta revisión se realiza para asegurar la consistencia y completitud de los métodos especificados en el Plan de gestión de configuración.

**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.

### Auditorias administrativas

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

Los responsables por estas revisiones son los miembros de la organización SQA, con la participación de todo elemento de la organización que tengan que ver con los requerimientos, estos elementos son: Diseñadores de software, pruebas de software, garantía de calidad del software, ingeniería de sistemas.

## Verificación

Se debe identificar todas las verificaciones que no fueron identificadas en el Plan de V & V para el software y debe especificar los métodos a ser usados.

## Reporte de problemas y acciones correctivas

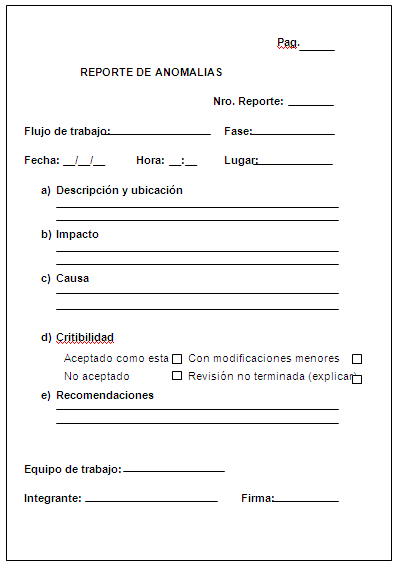
### Informe de Verificación y Validación

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

### Reporte sumario de fase V&V

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **REPORTE SUMARIO DE FASE V&V**  **FASE:** | | | **Pág…….** | |
| # De Reporte:  **Lugar:** | | **Fecha: / /**  **Hora:** | | |
| 1. Descripción de las tareas de V&V realizadas:        1. Sumario de resultados de tareas:        1. Sumario de anomalías y resolución:        1. Evaluación de calidad del software:        1. Recomendaciones: | | | | |
|  |  | | |  |
| **Equipo de Trabajo:**  **Nombre: Firma** | | |
|  | | | | |

### Reporte de Anomalías

****

### Reporte Final de V&V

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **REPORTE FINAL DE V&V**  **FASE:** | | | **Pág……..** | |
| # De Reporte:  **Lugar:** | | **Fecha: / /**  **Hora:** | | |
| a) Resumen de todas las tareas V&V, durante el ciclo de vida del software:      b) Resumen de resultados de tareas:      c) Resumen de anomalías, resolución:       1. Evaluación total de la calidad del software:        1. Recomendaciones: | | | | |
|  |  | | |  |
| **Equipo de Trabajo:**  **Nombre: Firma** | | |
|  | | | | |

## Herramientas, técnicas y metodologías

### Herramientas

Las herramientas de software a utilizar son las siguientes:

* Sistema operative Windows10
* Documentación o manual de ayuda

### Técnicas

Las técnicas empleadas:

* [ANSI/IEEE-STD-830]
* [ANSI/IEEE-STD-1016]
* [ANSI/IEEE-STD-1063]

### Metodología

La metodología para la garantía de calidad del software es el conjunto integral de las técnicas.

## Gestión de riesgos

Ver Documento de Riesgos.

## Referencia Bibliográfica

ANSI/IEEE STD 730.1-1989, IEEE Standard for Software Quality Assurance Plans.

[www.inf.utfsm.cl/~visconti/titulacion/FundamentosSQA.pdf](http://www.inf.utfsm.cl/~visconti/titulacion/FundamentosSQA.pdf)

[www.intelinfo.com/newly\_researched\_free\_training/software\_Quality\_Assurance\_and\_Testing.html](http://www.intelinfo.com/newly_researched_free_training/software_Quality_Assurance_and_Testing.html)

# Productividad basada en Herramientas CASE

## Enterprise Architect – Herramienta de diseño UML

Es una herramienta completa de diseño UML orientada a grandes equipos de desarrollo. Compatible con un rico conjunto de características incluyendo la auditoria del modelo, control de versiones, seguridad basada en roles, mapas conceptuales, la opción para un repositorio DBMS, creación y utilización de Tecnología MDG, modelos compartidos, depuración y visualización para ejecutar aplicaciones, y mucho más. El acceso a la interfaz de automatización proporciona a otras aplicaciones la habilidad de crear, acceder y modificar los elementos UML de Enterprise Architect.

La habilidad de importar y exportar modelos conformes con XMI permite la distribución de datos y comparación del modelo en líneas bases establecidas. Rápida generación de reportes en HTML y formatos de Texto enriquecidos, produciendo reportes de calidad profesional en segundos. La edición Corporativa también soporta seguridad de usuario, replicación y bloqueo a nivel de usuario de elementos y diagramas. Este extenso orden de características lo hace perfecto para cualquier equipo de desarrollo grande.

EA le ayuda a administrar la complejidad con herramientas para rastrear las dependencias, soporte para modelos muy grandes, control de versiones con proveedores CVS o SCC, Líneas Base por cada punto del tiempo, la utilidad de comparar (diff) para seguir los cambios del modelo, interfaz intuitiva y de alto rendimiento con vista de proyecto como un "explorador".

A provee una generación poderosa de documentos y herramientas de reporte con un editor de plantilla completo WYSIWYG. Genera reportes detallados y complejos de EA con la información que usted necesita en el formato que su compañía o cliente demanda.

EA soporta generación e ingeniería inversa de código fuente para muchos lenguajes populares, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic y PHP. También hay Add-ins gratis para CORBA y Python disponibles. Con un editor de código fuente con "resaltador de sintaxis" incorporado, EA le permite navegar y explorar su modelo de código fuente en el mismo ambiente. Para aquellos que trabajan en Eclipse o Visual Studio.Net, Sparx Systems también vende puentes livianos para estas IDE's, permitiéndole modelar en EA y saltar directamente al código fuente en su editor preferido. Las plantillas de generación de código le permiten personalizar el código fuente generado a las especificaciones de su compañía.

EA le ayuda a visualizar sus aplicaciones soportando ingeniería inversa de un amplio rango de lenguajes de desarrollo de software y esquemas de repositorios de base de datos. Ingrese frameworks completos desde código fuente o archivos Java .jar - o aún ensambladores binarios .Net! Importando frameworks y librerías de código, Ud. puede maximizar la re-utilización y entendimiento de su inversión existente.

EA soporta transformaciones de Arquitectura avanzada dirigida por Modelos (MDA) usando plantillas de transformaciones de desarrollo y fáciles de usar. Con transformaciones incorporadas para DDL, C#, Java, EJB y XSD, Ud. puede rápidamente desarrollar soluciones complejas desde los simples "modelos independientes de plataforma" (MIP) que son el objetivo en "modelos específicos de plataforma" (MEP). Un MIP se puede usar para generar y sincronizar múltiples MIP's - proveyendo un aumento de productividad significativo.

### Características de Enterprise Architect

### Soporta los 13 diagramas UML 2.1

* Los diagramas de comportamiento incluyen: Casos de Uso, Actividades, Estado, Interacción, Secuencia y Comunicación
* Los diagramas de Estructura Incluyen: Paquetes, Clases, Objetos, Composición, Componentes y Despliegue.

### Interfaz de usuario intuitiva

Amplio rango de barras de herramientas, ventanas acoplables, y estilos visuales.

* Guarde y restaure disposiciones de ventanas personalizadas.
* Modifique y personalice las barras de herramientas y menús.
* Cree sus propios aceleradores.
* Puede "Desplazar" las ventanas acopladas para maximizar el espacio de las ventanas y mejorar la eficiencia de su trabajo.
* Amplio sistema de menús para tener control de su modelo.
* Los accesos rápidos permiten la creación de elementos de diagramas y conexiones sensitivas al contexto

### Soporte para Transformaciones MDA

La Arquitectura Dirigida por Modelos permite transformar elementos simples del modelo en complejos.

* Transformaciones Completas conducidas por plantillas.
* Fácil para escribir y modificar transformaciones de plantillas.
* Realizar transformaciones para DDL, Java, C#, EJB, XSD.
* Generar y sincronizar su Modelo Específico de Plataforma del Modelo Independiente de Plataforma.
* Cada PIM puede soportar múltiples PSMs.

### Documentación flexible y comprensible.

Plantillas conducidas por generador RTF

* Las Plantillas soportan todas las características de los elementos del modelo de EA y datos extendidos (tales como Pruebas, Riesgos, Recursos, Cambios, etc.)
* Las plantillas soportan encabezados, pies, tablas de contenidos, imágenes embebidas, índices, títulos de páginas, tablas jerarquizadas y más.

### Salida en formato de texto enriquecido

Opciones simplificadas con filtros y criterios de selección.

Guarde las plantillas reportadas para una reutilización posterior

* Puede crear documentos maestros personalizados con vínculos sostenibles a las porciones generadas de la salida de EA
* Enterprise Architect permite exportar un modelo completo o una rama simple del modelo a páginas web en HTML
* Fije sus modelos en Internet o en su Intranet usando el generador de reportes HTML
* Los reportes adicionales incluyen para el uso de métricas de casos de uso, pruebas y mas

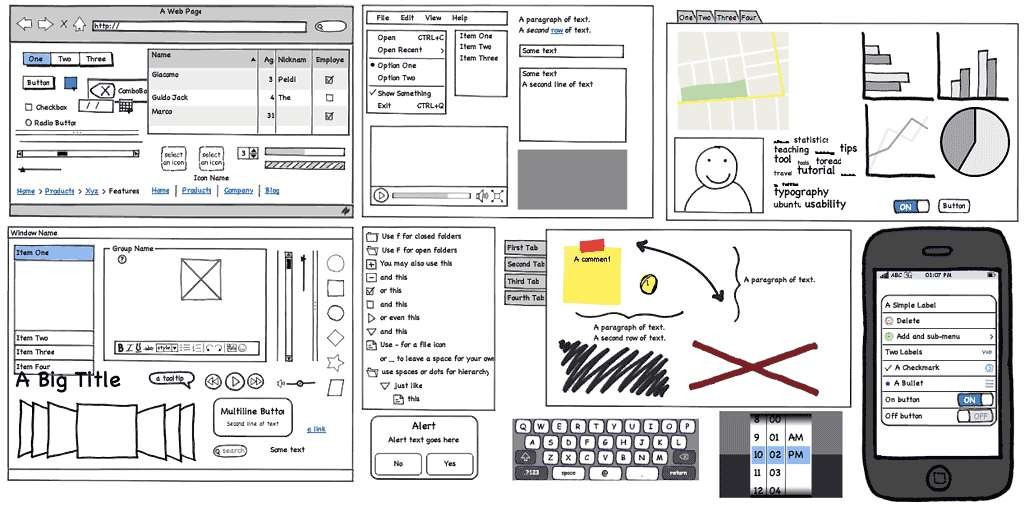
### Ingeniería de Código Directa e Inversa

Las plantillas completas encaminan al generador de ingeniería de código - modifique las plantillas incorporadas o empiece a escribirlas de cero

* Agregue lenguajes adicionales
* Sintaxis resaltada del editor del código fuente con capacidad para "guardar y sincronizar" con rapidez
* Puede generar código fuente C++, Java, C#, VB.Net, Visual Basic, Delphi, PHP, Python y ActionScript
* Soporte para Corba también disponible como "plug-in" libre

## Balsamiq Mockup

Herramienta utilizada para generar diseños de interfaz a través de prototipos



# Aspectos Legales

## 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.



### 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

### 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)

### 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.

### 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.

### 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) 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.

### 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.

## Derecho de Autor

### SENAPI

El Servicio Nacional de Propiedad Intelectual es una institución pública desconcentrada 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 desconcentrada 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.

# Infraestructura para la Producción del Software

## Gestión de Configuración de Software

### 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.

### 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.

### 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.

### 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.

### 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:

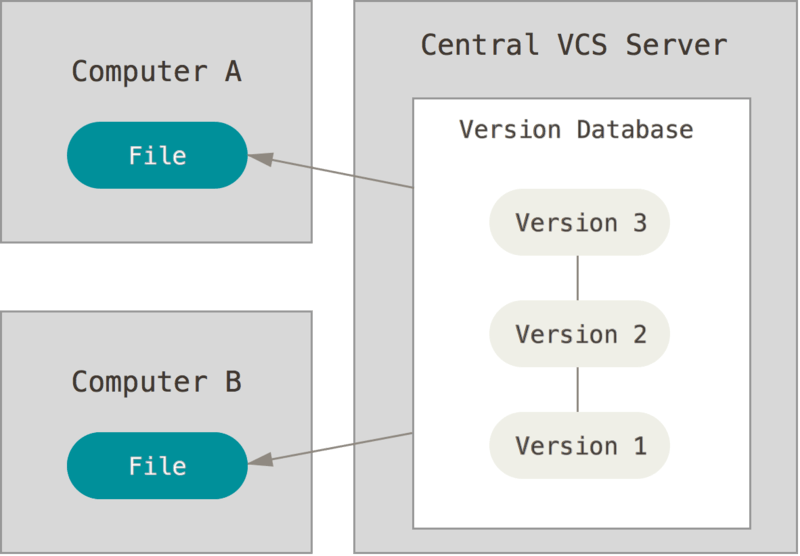
* + **Objetos básicos:** es una unidad de texto que pudo ser generada en el
  + análisis, el diseño, la codificación o prueba.
  + **Objetos compuestos:** son varios objetos simples, como por ejemplo la
  + especificación de diseño.

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.

## 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 puedas 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.

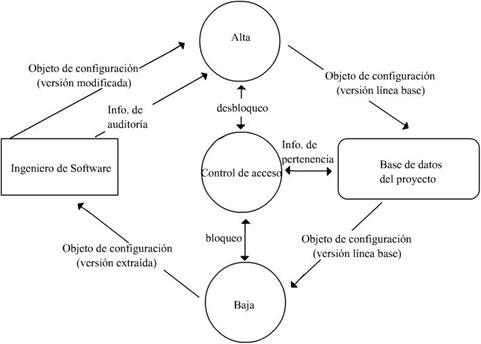
El sistema de control de versiones ayuda a registrar los cambios realizados al código. Aún más, registra quién realizó los cambios y puede restaurar el código borrado o modificado.

### 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 mejorarlos. 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.



## Herramientas utilizadas en el proyecto

### 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 unpull 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.

# Desarrollo del Sitio Web

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

* + Inicio
  + Servicios
  + Nosotros
  + Contactos
  + Chatbot

## Inicio





## Servicios



## Nosotros



**Proyectos:**



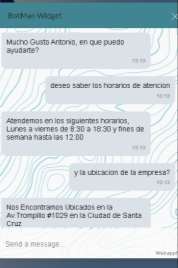
## Contactos



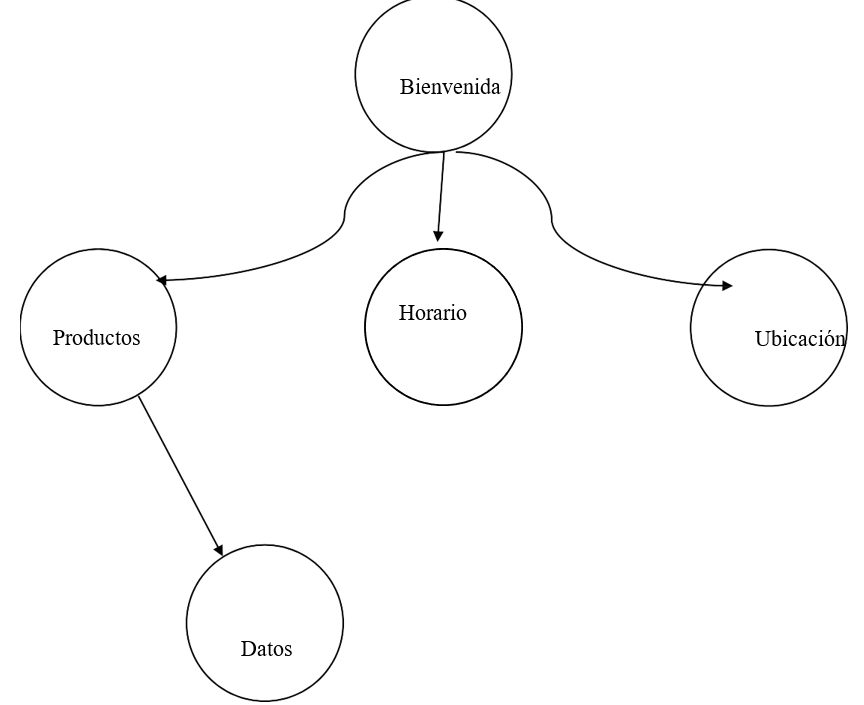
**Arquitectura de la Página Web**



## Chatbot



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

# Estudio de Mercado

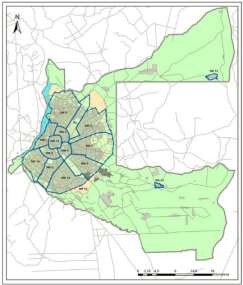
## Área del mercado

### 8.1.2. Extensión Geográfica

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

### 8.1.3. 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.

### 8.1.3. Ingresos

La segmentación de la población de acuerdo con los niveles de ingresos está orientada a toda diversidad de clases, empresas que cuenten con acceso a internet, teléfono celular y una computadora.

## Estudio de la demanda

De acuerdo a encuestas realizadas, el uso de una herramienta de control y seguimiento para los empleados, es algo que muchas empresas requieren, pero al ser la mayoría de las empresas que trabaja con promotores a trabajar por comisión, no tiene la necesidad de estar buscando a sus empleados ya que se podría decir que ellos mismos se ganan su sueldo.

## Estudio de la oferta

En cuanto a la oferta, actualmente en el mercado se encuentran unas cuantas aplicaciones que se posicionan como competencia inmediata:

**Find my device** o encontrar mi dispositivo de Google, realiza la búsqueda de un dispositivo móvil realizando algunos simples pasos, solo debes iniciar sesión para ver que

dispositivos tienes vinculados. Con este sistema se puede mostrar la ubicación o bloquear la pantalla, para proteger el dispositivo.

### Monetización

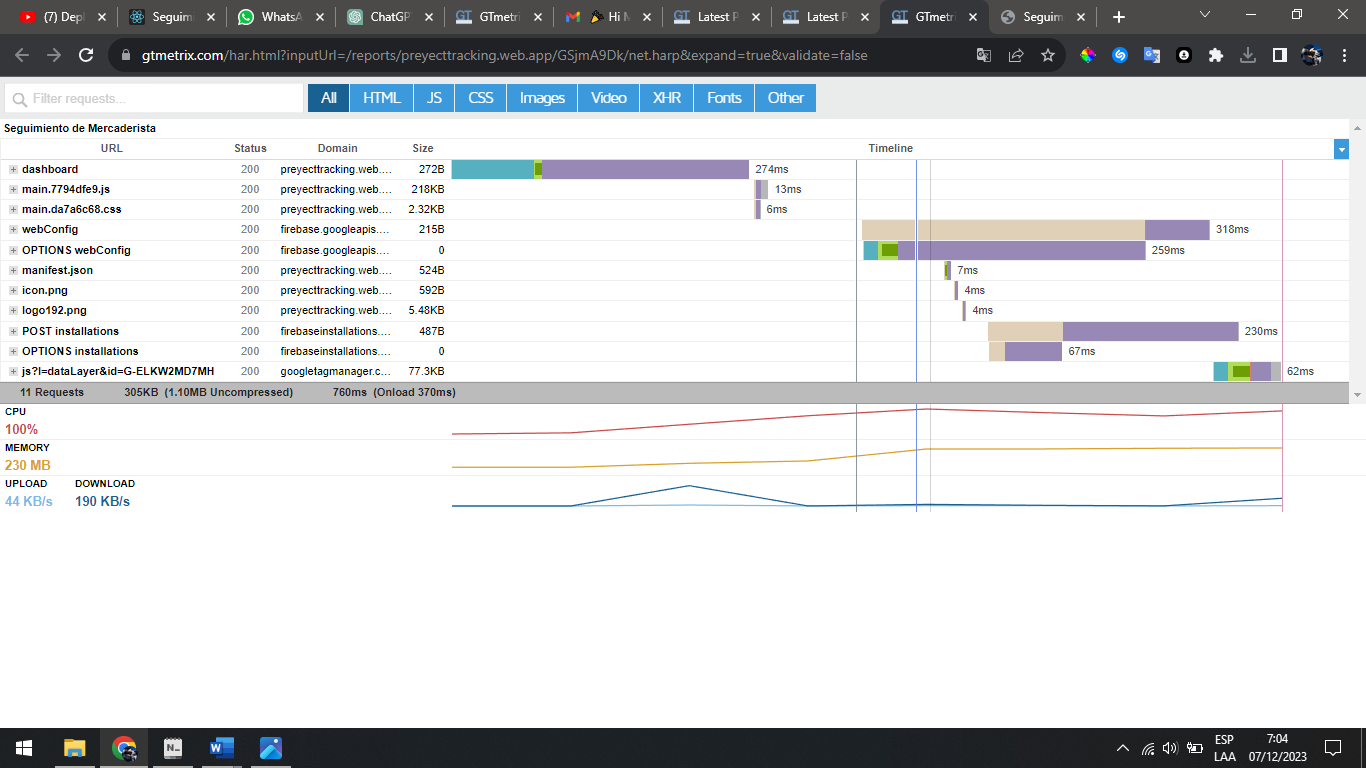
**Planes de servicio para empleadores:** Las empresas encargadas de usar el servicio de control y seguimiento, deberán pagar monetariamente planes mensuales y/o anuales para usar las funcionalidades del proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
| **Planes** | **Libre** | **Mensual** | **Anual** |
| **Descripción** | Permite el registro de usuarios en la base de datos  mediante el sistema. | Además de las funciones gratuitas, se permitirá visualizar un mapa con la ubicación exacta del vendedor de su empresa. | |
| **Costo aproximado** | Gratis | 9.99 $us | 99.99 $us |

# Pruebas en el Software

## 9.2. Pruebas de Rendimiento

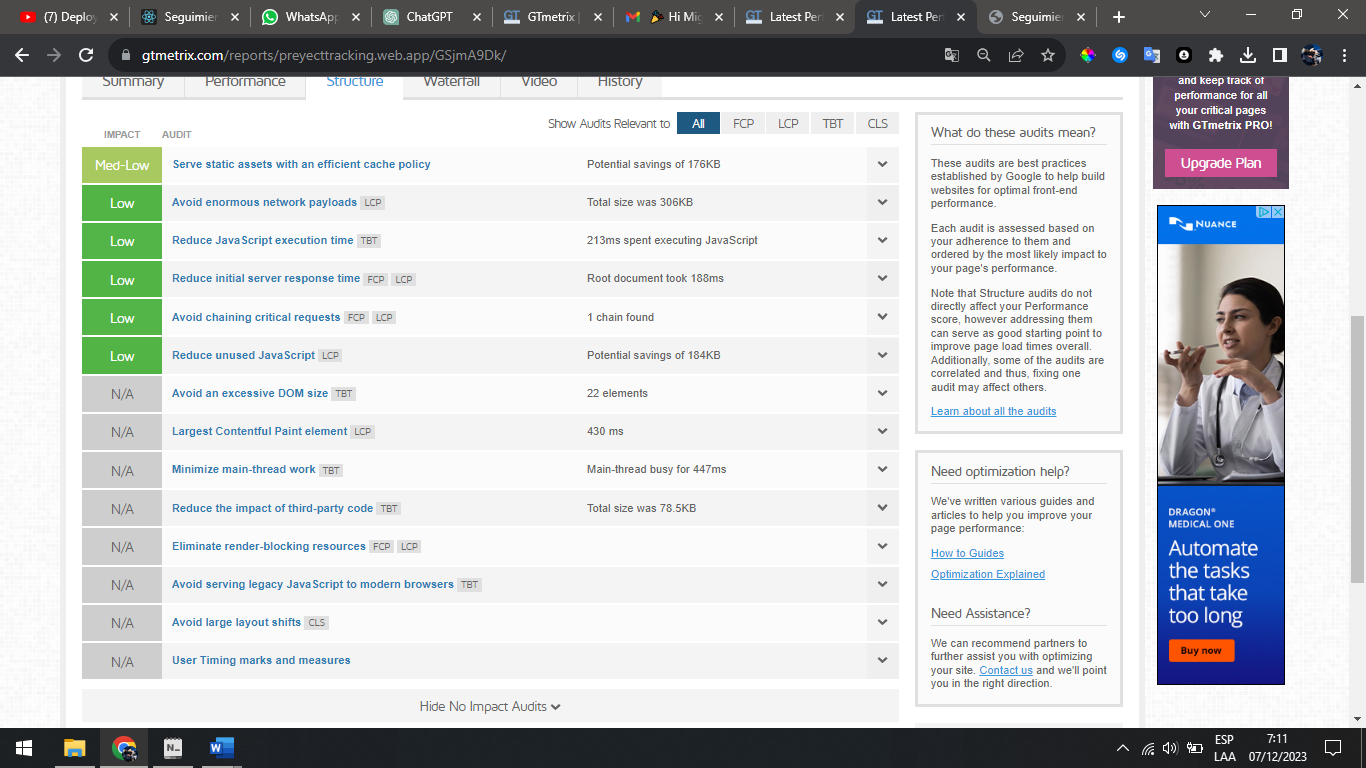
En cualquier aplicación web, lograr un rendimiento óptimo y una carga eficiente de recursos es fundamental. Esto se traduce en tiempos de espera más cortos al acceder al software y la capacidad de renderizar la aplicación incluso en situaciones con una señal de internet débil.



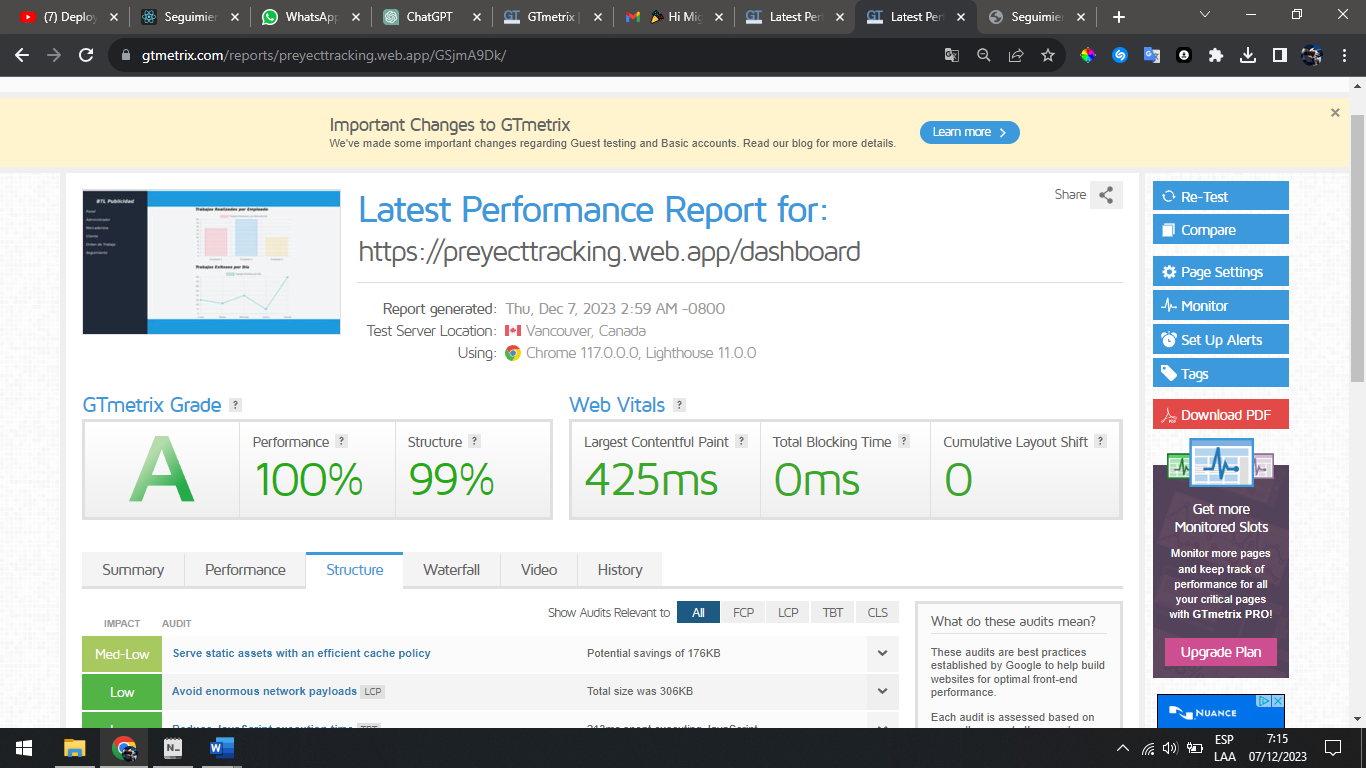
Utilizando la herramienta automatizada Gtmetrix, obtenemos un gráfico que resume el consumo de recursos de nuestra aplicación en cada solicitud. Observamos que se requieren al menos 230MB de RAM libre, una cantidad relativamente baja. Además, el uso de ancho de banda de carga y descarga de nuestra aplicación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

## 9.2. 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.



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.

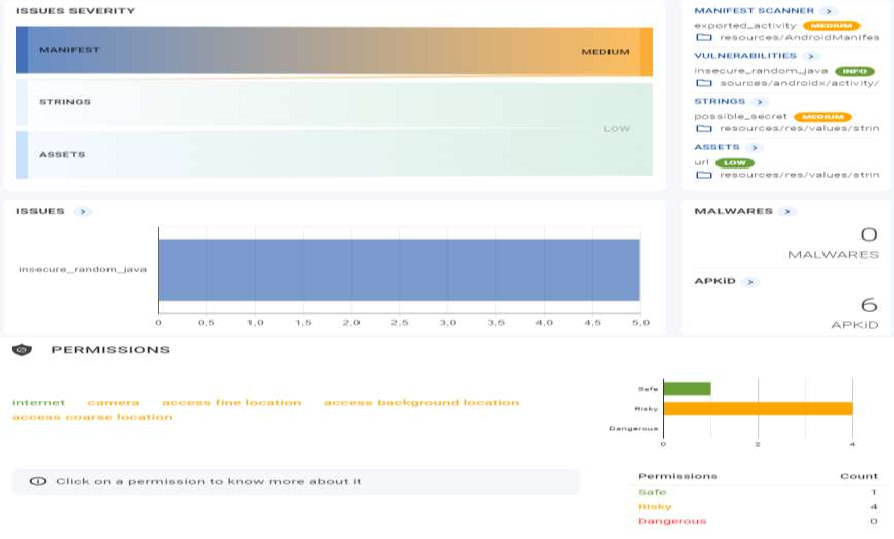


En resumen, según Gtmetrix, nuestra aplicación web recibe una calificación de A, lo que indica que no presenta problemas graves de vulnerabilidad y su rendimiento está estructurado correctamente en un 99%.

## 9.3. Pruebas en el entorno Móvil

### 9.3.1. Pruebas de vulnerabilidad

Dado que nuestra aplicación sigue una arquitectura SOA, existe la posibilidad de que un atacante pueda interceptar las solicitudes a los puntos finales de nuestra aplicación web. Para contrarrestar este riesgo potencial, implementaremos una herramienta automatizada conocida como BeVigil. Esta herramienta se encargará de escanear minuciosamente cada componente utilizado en el desarrollo de nuestra aplicación móvil, buscando mitigar posibles amenazas.



En resumen, nuestra aplicación no representa un riesgo alto en términos de vulnerabilidad y tampoco cuenta con permisos significativos o peligrosos para el dispositivo del usuario final. Por consiguiente, cumple con los estándares de calidad establecidos en el SQAP propuesto.

# Marketing

## 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 especifico. 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.

## Organización

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

## 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.

## 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.

## Estrategias de marketing

### 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 productos(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.

## 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:

### 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.

### 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.

### 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.

## Medida de éxito

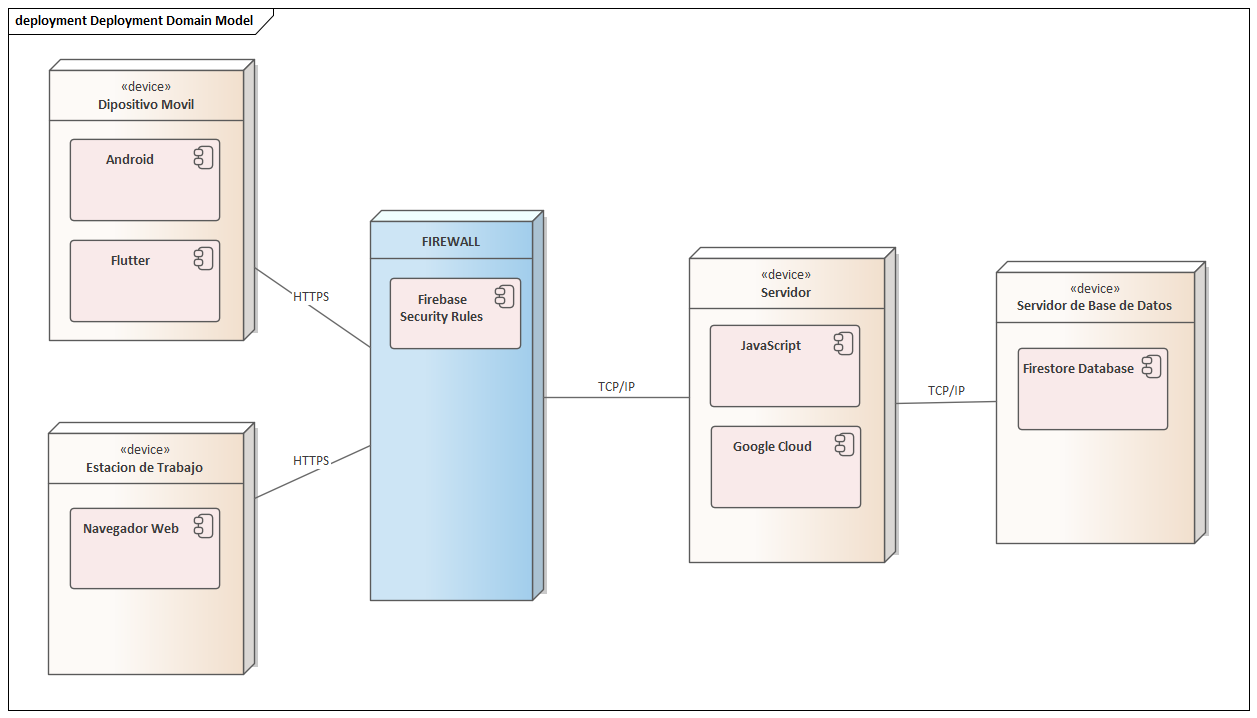
Entre las medidas de Éxito que se tomarán en cuenta se tiene las siguientes:

* 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.

# Despliegue

## 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](https://opensource.org/licenses/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/.

## 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:

* ID del usuario
* 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

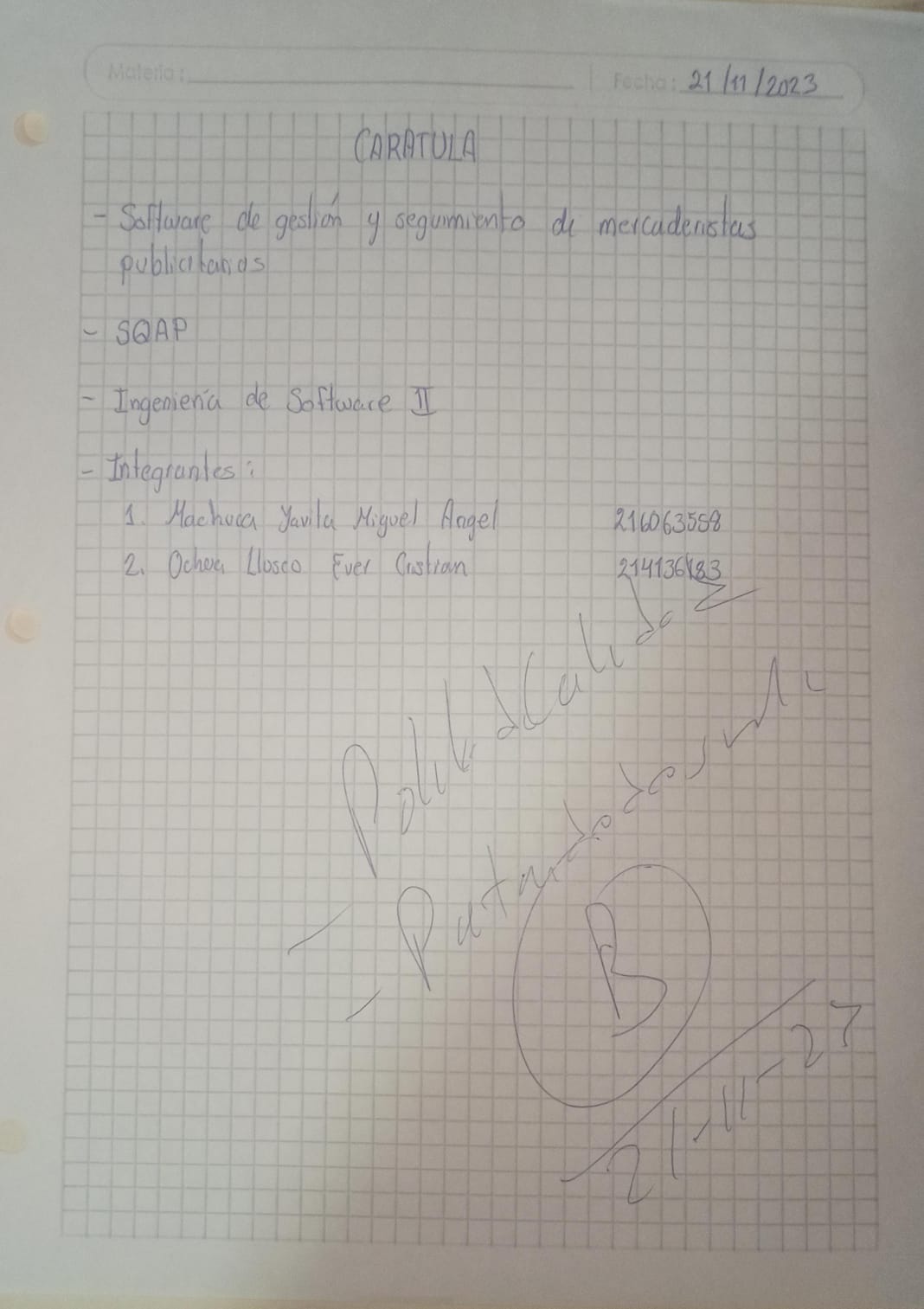
Nos reservamos el derecho de actualizar nuestra Política de Privacidad en cualquier momento. Cualquier cambio será publicado en esta página. Le recomendamos que revise periódicamente esta Política para estar informado sobre cómo protegemos la información que recopilamos.

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/.

# ANEXOS

## CARATULA SQAP



## Caratula de TALLER DE GRADO

