**UNIVERSIDAD CONTINENTAL**

**FACULTAD DE INGENIERÍA**

**ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**



**PROYECTO**

**“Desarrollo de un Sistema de Emergencia Basado en GPS y Botón SOS para Personas en Situación de Riesgo en el Perú”**

# PORTADA

**PRESENTADO POR:**

|  |  |
| --- | --- |
| **APELLIDOS Y NOMBRES** | **CÓDIGO** |
| **BASTIDAS VILCA Nikole** | **73498635** |
|  |  |
|  |  |
|  |  |

**ASESOR:**

**MG. ARANA CAPARACHIN, Maglioni**

**HUANCAYO – PERÚ**

**2023**

# LISTA DE CONTENIDO

[PORTADA 1](#_Toc144910742)

[LISTA DE CONTENIDO 2](#_Toc144910743)

[LISTA DE TABLAS 7](#_Toc144910744)

[LISTA DE FIGURAS 8](#_Toc144910745)

[CAPÍTULO 1 9](#_Toc144910746)

[PLANTEAMIENTO DEL ESTUDIO 9](#_Toc144910747)

[1.1. Aspectos Generales de la Empresa 9](#_Toc144910748)

[1.1.1. Organigrama 9](#_Toc144910749)

[1.1.2. Misión y visión 9](#_Toc144910750)

[1.2. Diagnóstico del Problema 9](#_Toc144910751)

[1.3. Procesos de la Empresa 9](#_Toc144910752)

[1.4. Oportunidad Encontrada 9](#_Toc144910753)

[1.5. Detalles del Proyecto 9](#_Toc144910754)

[1.5.1. Solución planteada 9](#_Toc144910755)

[1.5.2. Objetivos generales 10](#_Toc144910756)

[CAPÍTULO 2 11](#_Toc144910757)

[ESTUDIO DE FACTIBILIDAD 11](#_Toc144910758)

[2.1. Alternativas de Solución 11](#_Toc144910759)

[2.2. Factibilidad Técnica 11](#_Toc144910760)

[2.2.1. Hardware: Servidor 11](#_Toc144910761)

[2.3. Factibilidad Económica 12](#_Toc144910762)

[2.3.1. Gastos generales 12](#_Toc144910763)

[2.4. Factibilidad Operacional 12](#_Toc144910764)

[2.4.1. Actividad del sistema de ventas 13](#_Toc144910765)

[2.4.1.1. Fase de análisis. 13](#_Toc144910766)

[2.4.1.2. Fase de diseño y desarrollo. 13](#_Toc144910767)

[2.4.1.3. Fase de transición. 14](#_Toc144910768)

[2.4.1.4. Fase de implementación. 14](#_Toc144910769)

[CAPÍTULO 3 15](#_Toc144910770)

[ANÁLISIS DE REQUERIMIENTOS 15](#_Toc144910771)

[3.1. Metas del Sistema de Información 15](#_Toc144910772)

[3.2. Requisitos del Sistema 15](#_Toc144910773)

[3.2.1. Requerimientos funcionales 15](#_Toc144910774)

[3.2.2. Requerimientos no funcionales 16](#_Toc144910775)

[3.3. Identificación de Actores del Sistema 17](#_Toc144910776)

[3.3.1. Administrador 17](#_Toc144910777)

[3.3.2. Cajero 17](#_Toc144910778)

[CAPÍTULO 4 18](#_Toc144910779)

[PLANIFICACIÓN DEL PROYECTO 18](#_Toc144910780)

[4.1. Definición de Roles de Trabajo 18](#_Toc144910781)

[4.1.1. Product owner 18](#_Toc144910782)

[4.1.2. Scrum master 18](#_Toc144910783)

[4.1.3. Team member 18](#_Toc144910784)

[4.1.4. Tester 18](#_Toc144910785)

[4.2. Product Backlog 19](#_Toc144910786)

[4.3. Sprint Backlog 20](#_Toc144910787)

[4.3.1. Sprint 1 20](#_Toc144910788)

[4.3.2. Sprint 2 21](#_Toc144910789)

[4.3.3. Sprint 3 21](#_Toc144910790)

[4.3.4. Sprint 4 22](#_Toc144910791)

[4.4. Planificación de Sprints 22](#_Toc144910792)

[4.4.1. Historias de usuario 22](#_Toc144910793)

[4.5. Cronograma de Actividades 24](#_Toc144910794)

[4.6. Gestión de Riesgos 25](#_Toc144910795)

[CAPÍTULO 5 26](#_Toc144910796)

[DISEÑO DEL SISTEMA DE INFORMACIÓN 26](#_Toc144910797)

[5.1. Diseño de Diagramas UML 26](#_Toc144910798)

[5.1.1. Diagramas de casos de uso 26](#_Toc144910799)

[5.1.2. Diagramas de secuencia 26](#_Toc144910800)

[5.1.3. Diagramas de colaboración 26](#_Toc144910801)

[5.1.4. Diagramas de clases 26](#_Toc144910802)

[5.2. Diseño de Base de Datos 26](#_Toc144910803)

[5.2.1. Diseño conceptual (E/R) 26](#_Toc144910804)

[5.2.2. Diseño lógico 26](#_Toc144910805)

[5.2.3. Diseño físico 26](#_Toc144910806)

[5.2.4. Modelado de base de datos 26](#_Toc144910807)

[5.3. Diseño de Interfaces Básicas 26](#_Toc144910808)

[5.3.1. Acceso login 27](#_Toc144910809)

[5.3.2. Interfaz … 27](#_Toc144910810)

[CAPÍTULO 6 28](#_Toc144910811)

[CODIFICACIÓN DEL SOFTWARE 28](#_Toc144910812)

[6.1. Desarrollo del Sprint 1 28](#_Toc144910813)

[6.1.1. Sprint planning 28](#_Toc144910814)

[6.1.2. Sprint backlog 28](#_Toc144910815)

[6.1.3. Historias de usuarios 28](#_Toc144910816)

[6.1.4. Taskboard 28](#_Toc144910817)

[6.1.5. Daily scrum 28](#_Toc144910818)

[6.1.6. Sprint review 28](#_Toc144910819)

[6.1.7. Criterios de aceptación 28](#_Toc144910820)

[6.1.8. Resultados del sprint 28](#_Toc144910821)

[6.1.8.1. Evidencias. 28](#_Toc144910822)

[6.1.8.2. Prueba de desarrollo. 28](#_Toc144910823)

[6.1.8.3. …. 29](#_Toc144910824)

[6.1.9. Sprint retrospective 29](#_Toc144910825)

[6.2. Desarrollo del Sprint 2 29](#_Toc144910826)

[6.2.1. Sprint planning 29](#_Toc144910827)

[6.2.2. Sprint backlog 29](#_Toc144910828)

[6.2.3. Historias de usuarios 29](#_Toc144910829)

[6.2.4. Taskboard 29](#_Toc144910830)

[6.2.5. Daily scrum 29](#_Toc144910831)

[6.2.6. Sprint review 29](#_Toc144910832)

[6.2.7. Criterios de aceptación 29](#_Toc144910833)

[6.2.8. Resultados del sprint 29](#_Toc144910834)

[6.2.8.1. Evidencias. 29](#_Toc144910835)

[6.2.8.2. Prueba de desarrollo. 30](#_Toc144910836)

[6.2.8.3. …. 30](#_Toc144910837)

[6.2.9. Sprint retrospective 30](#_Toc144910838)

[6.3. Desarrollo del Sprint 3 30](#_Toc144910839)

[6.3.1. Sprint planning 30](#_Toc144910840)

[6.3.2. Sprint backlog 30](#_Toc144910841)

[6.3.3. Historias de usuarios 30](#_Toc144910842)

[6.3.4. Taskboard 30](#_Toc144910843)

[6.3.5. Daily scrum 30](#_Toc144910844)

[6.3.6. Sprint review 30](#_Toc144910845)

[6.3.7. Criterios de aceptación 30](#_Toc144910846)

[6.3.8. Resultados del sprint 30](#_Toc144910847)

[6.3.8.1. Evidencias. 31](#_Toc144910848)

[6.3.8.2. Prueba de desarrollo. 31](#_Toc144910849)

[6.3.8.3. …. 31](#_Toc144910850)

[6.3.9. Sprint retrospective 31](#_Toc144910851)

[6.4. Desarrollo del Sprint 4 31](#_Toc144910852)

[6.4.1. Sprint planning 31](#_Toc144910853)

[6.4.2. Sprint backlog 31](#_Toc144910854)

[6.4.3. Historias de usuarios 31](#_Toc144910855)

[6.4.4. Taskboard 31](#_Toc144910856)

[6.4.5. Daily scrum 31](#_Toc144910857)

[6.4.6. Sprint review 31](#_Toc144910858)

[6.4.7. Criterios de aceptación 31](#_Toc144910859)

[6.4.8. Resultados del sprint 32](#_Toc144910860)

[6.4.8.1. Evidencias. 32](#_Toc144910861)

[6.4.8.2. Prueba de desarrollo. 32](#_Toc144910862)

[6.4.8.3. …. 32](#_Toc144910863)

[6.4.9. Sprint retrospective 32](#_Toc144910864)

[CONCLUSIONES 33](#_Toc144910865)

[RECOMENDACIONES 34](#_Toc144910866)

[ANEXOS 35](#_Toc144910867)

[Anexo 01. Manual Técnico 36](#_Toc144910868)

[Anexo 02. Manual de Usuario 37](#_Toc144910869)

# LISTA DE TABLAS

[Tabla 1. Modelo de CPU - Servidor. 11](#_Toc144307328)

[Tabla 2. Gastos generales presupuestados. 12](#_Toc144307329)

# LISTA DE FIGURAS

[Figura 1. Recursos presupuestados para el proyecto. 12](#_Toc144307330)

# CAPÍTULO 1

# PLANTEAMIENTO DEL ESTUDIO

## Aspectos Generales de la Empresa

El presente proyecto tiene como propósito desarrollar un sistema de alerta y geolocalización satelital llamado **CELSOS**, orientado a brindar asistencia tecnológica en situaciones de riesgo o desaparición de personas. El sistema se basa en el uso de una **Raspberry Pi** equipada con un módulo **GPS**, un **botón físico SOS** y conexión a una plataforma web, donde se visualizará la ubicación del usuario en tiempo real. El contexto de desarrollo es académico, en el marco del curso **Pruebas y Calidad de Software** de la Universidad Continental. Sin embargo, el objetivo es que esta solución pueda proyectarse hacia aplicaciones reales, escalables y con potencial de implementación en instituciones como **la Policía Nacional del Perú (PNP)**, el **Ministerio de la Mujer y Poblaciones Vulnerables**, y organizaciones como **INABIF**, que atienden casos de personas desaparecidas, niños en situación de abandono, y mujeres víctimas de violencia. La problemática actual muestra que, si bien existen tecnologías móviles con GPS, **no todos los usuarios tienen acceso continuo a un celular** o conexión estable. Además, los dispositivos móviles no están optimizados para emitir alertas discretas e inmediatas en momentos de peligro. En respuesta, CELSOS ofrece una alternativa sencilla, autónoma y confiable.

### Organigrama

El proyecto será desarrollado por un equipo académico conformado por cuatro integrantes, distribuidos de acuerdo a las siguientes responsabilidades:

|  |
| --- |
|  |

### Misión y visión

### Misión:

Diseñar y desarrollar un sistema de alerta y geolocalización portátil que permita identificar la ubicación de una persona en riesgo en tiempo real, ofreciendo una herramienta tecnológica accesible y de impacto social.

**Visión:**

Ser una solución tecnológica referente a nivel nacional en materia de seguridad y localización de personas vulnerables, con miras a incorporar tecnologías emergentes como la comunicación cuántica para reforzar la protección de los datos y la seguridad de las alertas

## Diagnóstico del Problema

En el contexto peruano, diariamente se reportan casos de personas desaparecidas, incluyendo niños, adultos mayores y mujeres víctimas de violencia. Las estrategias tradicionales de búsqueda, como afiches, redes sociales o llamadas a autoridades, suelen ser lentas, desorganizadas y dependen en gran medida del acceso a tecnología o del testimonio de terceros. Esta situación genera un alto nivel de incertidumbre, estrés y, en muchos casos, desenlaces trágicos que podrían haberse evitado con una respuesta más rápida y automatizada. Por otro lado, aunque los dispositivos móviles modernos cuentan con funciones de geolocalización, **no todas las personas tienen acceso constante a un smartphone**, ya sea por limitaciones económicas, edad, deterioro cognitivo, o simplemente por no saber usar la tecnología. Además, en situaciones de emergencia, un teléfono puede estar descargado, fuera de cobertura, o no ser práctico de manipular. Frente a esta problemática, se plantea la necesidad de un **sistema especializado, autónomo y accesible** que pueda ser activado con facilidad por cualquier persona en peligro, y que permita a sus familiares o autoridades conocer su ubicación con precisión y en tiempo real, incluso en zonas sin cobertura móvil tradicional. CELSOS surge como una propuesta tecnológica enfocada en brindar una herramienta de respuesta inmediata, a través de un dispositivo físico portátil con botón SOS y GPS, cuya información se envía a una plataforma web mediante servicios satelitales o redes de datos, con posibilidad de expansión futura a tecnologías más seguras como la comunicación cuántica.

## Procesos de la Empresa

El sistema CELSOS contempla varios procesos internos que permiten su funcionamiento de forma efectiva. A continuación, se describen los principales:

1. **Activación del botón SOS**: El usuario, al encontrarse en una situación de riesgo, presiona un botón físico que activa el sistema.
2. **Captura de coordenadas GPS**: La Raspberry Pi obtiene en tiempo real la latitud y longitud del usuario, utilizando un módulo GPS.
3. **Transmisión de datos**: La información geográfica y la alerta se envían automáticamente a un servidor backend, usando una red disponible.
4. **Visualización web**: En una interfaz web, se muestra la ubicación del usuario en un mapa dinámico, permitiendo a los familiares o autoridades tomar decisiones rápidas.
5. **Registro de historial**: Las alertas y ubicaciones son almacenadas para permitir una revisión posterior o análisis estadístico.

## Oportunidad Encontrada

La oportunidad radica en aplicar herramientas de bajo costo y tecnologías disponibles, como la Raspberry Pi y los sistemas de código abierto (Leaflet.js, Flask, etc.), para crear una solución social con alto impacto. Aunque existen aplicaciones móviles de seguridad, **no hay un sistema portátil, autónomo y especializado** como el que se propone con CELSOS, al menos no en el mercado peruano con enfoque en poblaciones vulnerables. Además, el proyecto se alinea con los Objetivos de Desarrollo Sostenible (ODS), particularmente con el objetivo 16: "Paz, justicia e instituciones sólidas", al contribuir a la protección de personas en riesgo y facilitar la acción inmediata ante casos de desaparición.

## Detalles del Proyecto

### Solución planteada

CELSOS se propone como una solución tecnológica de respuesta inmediata ante casos de emergencia personal, especialmente útil en situaciones como:

* Desaparición de menores de edad.
* Adultos mayores con riesgo de desorientación.
* Mujeres en peligro por violencia de género.
* Personas extraviadas en zonas rurales o de difícil acceso.

La solución está compuesta por tres módulos principales:

* **Módulo físico**: Dispositivo con botón SOS, GPS y Raspberry Pi.
* **Módulo de comunicación**: API REST para recepción y procesamiento de datos.
* **Módulo visual**: Plataforma web con mapa para visualización en tiempo real.

La arquitectura permite que el sistema funcione de forma autónoma, incluso en zonas sin cobertura celular, si se conecta a una red satelital o módulos alternativos de comunicación de baja potencia.

### Objetivos generales

Diseñar e implementar un sistema de alerta y localización satelital basado en Raspberry Pi, que permita identificar en tiempo real la ubicación de personas en riesgo, integrando estándares de calidad de software y utilizando la metodología SCRUM para su desarrollo ágil.

# CAPÍTULO 2

# ESTUDIO DE FACTIBILIDAD

## Alternativas de Solución

Durante el análisis inicial se consideraron tres alternativas principales:

**1. Aplicaciones móviles tradicionales:** Plataformas de Android que utilizan el GPS del celular y el botón de emergencia digital. Desventaja: requieren conexión estable, batería suficiente y conocimientos básicos de uso del celular.

**2. Dispositivos comerciales tipo rastreador:** Algunos dispositivos de geolocalización se encuentran en el mercado, pero son costosos, dependen de suscripciones mensuales y no están adaptados al contexto social y económico peruano.

**3. CELSOS – Solución propuesta:** Un sistema desarrollado con hardware de bajo costo, con botón físico y GPS, que permite enviar una alerta rápida en caso de emergencia, visualizar la ubicación en una plataforma web, y todo bajo estándares de calidad y metodología ágil SCRUM. Esta es la alternativa seleccionada.

## Factibilidad Técnica

CELSOS se plantea como un sistema factible desde el punto de vista técnico, ya que utiliza tecnologías disponibles, económicas y documentadas. La arquitectura modular facilita el desarrollo progresivo de cada componente (hardware, backend y frontend).

### Hardware: Servidor

El sistema puede operar desde un servidor local o plataformas gratuitas como Firebase o Railway. Las pruebas locales se harán desde una PC.

Tabla 1. Modelo de CPU - Servidor.

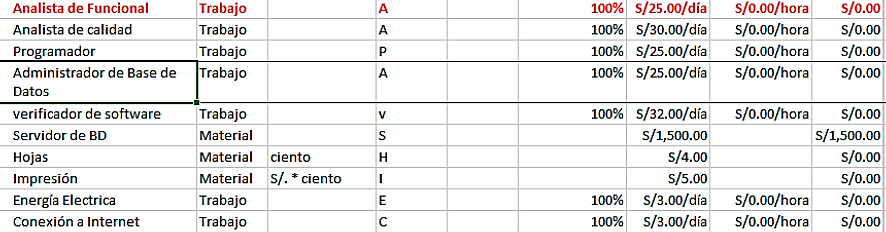
|  |  |  |
| --- | --- | --- |
|  | **Características** | **Valores** |
| **1** | Número de núcleos | 12 |
| **2** | Memoria caché | 128 MB |
| **3** | Frecuencia mínima | 1.2GHz |
| **4** | Frecuencia base máxima turbo | 3.3GHz |
| **5** | RAM compatible | DDR4 |
| **6** | Memoria ECC compatible | No |
| **7** | Número de PCI | 1x16 + 1x8 + 1x4 |
| **8** | Máximo PCI Express | 8GT/s |
| **9** | TDP | 140w |
| **10** | Disco Duro | 1TB |
| **11** | Tarjeta de red | 1300Mbps – 2.4G |
| **12** | Tarjeta de video | 16GB GDDR5 |

Fuente. Elaboración propia.

## Factibilidad Económica

El proyecto tiene alta viabilidad económica por el uso de componentes accesibles y software libre.

Figura 1. Recursos presupuestados para el proyecto.



Fuente. Elaboración propia.

### Gastos generales

Tabla 2. Gastos generales presupuestados.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del Recurso** | **Tasa Estándar** | **Días** | **Total** |
| Project Manager (Sponsor) | S/ 40.00 | S/ 60.00 | S/ 2,400.00 |
| Analista funcional | S/ 25.00 | S/ 60.00 | S/ 1,500.00 |
| Analista de calidad | S/ 30.00 | S/ 60.00 | S/ 1,800.00 |
| Programador | S/ 25.00 | S/ 60.00 | S/ 1,500.00 |
| Administrador de base de datos | S/ 25.00 | S/ 60.00 | S/ 1,500.00 |
| Tester de software | S/ 32.00 | S/ 60.00 | S/ 1,920.00 |
| Servidor de base de datos | S/ 35.00 | S/ 60.00 | S/ 2,100.00 |
| Hojas bond | S/ 4.00 | S/ 60.00 | S/ 240.00 |
| Impresión de documentos | S/ 5.00 | S/ 60.00 | S/ 300.00 |
| Conexión a internet | S/ 3.00 | S/ 60.00 | S/ 180.00 |
| Energía eléctrica | S/ 3.00 | S/ 60.00 | S/ 180.00 |
|  |  |  | **S/ 13,620.00** |

Fuente. Elaboración propia.

* + Raspberry Pi 4B: S/ 180.00  
    • Módulo GPS: S/ 40.00  
    • Botón SOS + cables: S/ 10.00  
    • Fuente energía: S/ 60.00  
    • Internet: S/ 30.00  
    • Documentación e impresión: S/ 35.00  
    Total aproximado: S/ 355.00

## Factibilidad Operacional

La factibilidad operacional permite a la empresa obtener un sistema de ventas aprovechando los beneficios que ofrece y facilita el buen manejo de recursos. A continuación, se describe las actividades que buscan completarse para llegar a cabo la implantación exitosa de la solución propuesta para las ventas en la Empresa Dolphis Consulting.

### Actividad del sistema de ventas

#### Fase de análisis.

* Estudio previo y entrevista con el cliente.
* Identificación de requisitos de usuario.
* Identificación de requisitos de rendimiento y de hardware y software.
* Identificación de requisitos de interfaz.
* Especificación de requerimientos del producto.
* Análisis de procesos.

#### Fase de diseño y desarrollo.

* Ingreso al sistema mediante acceso login.
* Elaboración de interfaz de gestión de usuarios.
* Maquetación del sistema de información.
* Elaboración del diseño de base de datos.
* Elaboración de consultas para interfaz login.
* Desarrollo de código para acceso login.
* Planificación de gestión de usuarios.
* Elaboración de consultas para registro de usuarios.
* Codificación de interfaz para registro de usuarios.
* Planificación de gestión de clientes.
* Elaboración de consultas para registro de clientes.
* Codificación de interfaz para registro de clientes.
* Planificación de gestión de proveedores.
* Elaboración de consultas para registro de proveedores.
* Codificación de interfaz para registro de proveedores.
* Planificación de gestión de productos.
* Elaboración de consultas para registro de productos.
* Codificación de interfaz para registro de productos.
* Planificación de ingreso de ventas realizadas.
* Realizar las consultas para generar el ingreso de ventas.
* Codificación de interfaz para ventas.
* Planificación de reportes y consultas del registro de ventas.
* Elaboración de querys para reportes y consultas.
* Codificación de interfaz de reportes y consultas.
* Elaboración de interfaz principal.

#### Fase de transición.

* Carga de productos.
* Reporte y consultas de stock de productos.
* Pruebas internas.
* Pruebas de usuarios.

#### Fase de implementación.

* Entrega final del aplicativo.
* Elaboración del manual del aplicativo.
* Capacitaciones al personal.

El recurso de tiempo queda plasmado en un cronograma de desarrollo, en el cual se determinan plazos, prioridad, duración y responsables, para cada una de las actividades descritas con anterioridad.

# CAPÍTULO 3

# ANÁLISIS DE REQUERIMIENTOS

## Metas del Sistema de Información

* Diseñar e implementar un sistema de alerta personal que funcione mediante un botón físico y transmita en tiempo real la ubicación de una persona en situación de riesgo.
* Ofrecer una solución tecnológica accesible, confiable y de fácil uso, capaz de operar con o sin conexión móvil convencional.
* Visualizar la ubicación del usuario en tiempo real a través de una plataforma web con mapas interactivos, permitiendo a familiares o autoridades reaccionar rápidamente.
* Registrar de forma segura las alertas generadas, junto con su fecha, hora y coordenadas, para ser consultadas por usuarios autorizados.
* Garantizar la calidad del software aplicando criterios definidos por la norma ISO 9126, así como principios ágiles mediante SCRUM.

## Requisitos del Sistema

Para iniciar con el desarrollo del sistema CELSOS, es fundamental establecer claramente los requerimientos funcionales y no funcionales que el sistema debe cumplir. Estos requerimientos fueron definidos en base a entrevistas con potenciales usuarios y el análisis de casos reales de desapariciones en el Perú. A continuación, se describen:

### Requerimientos funcionales

* El sistema contará con una activación de emergencia mediante un botón físico integrado al dispositivo.
* El sistema deberá captar la ubicación GPS del usuario en tiempo real al momento de activar la alerta.
* El dispositivo deberá enviar automáticamente los datos captados (latitud, longitud, fecha y hora) a un servidor backend vía internet o red local.
* El backend almacenará los datos recibidos en una base de datos para registro y auditoría.
* La plataforma web mostrará en tiempo real la ubicación del usuario alertante en un mapa interactivo (Leaflet.js).
* La interfaz web permitirá visualizar el historial de alertas registradas.
* El sistema deberá operar de manera continua sin necesidad de intervención constante del usuario.
* Solo personal autorizado (familiares o autoridades) podrá visualizar los datos del sistema.
* El sistema permitirá pruebas simuladas para testeo técnico sin activar una alerta real.
* El sistema podrá ser actualizado sin necesidad de reconfiguración total del dispositivo.

### Requerimientos no funcionales

* El sistema deberá operar en una Raspberry Pi con al menos 1 GB de RAM y conexión a internet o red local.
* La visualización web será compatible con navegadores modernos como Chrome, Firefox y Edge.
* El sistema será desarrollado con herramientas de código abierto: Python, Flask, Leaflet.js y Firebase o MongoDB.
* El tiempo de respuesta del sistema desde que se presiona el botón hasta la visualización en el mapa no deberá exceder los 3 segundos.
* El sistema deberá estar disponible el 99% del tiempo durante pruebas.
* Se implementarán protocolos de seguridad para evitar accesos no autorizados a los datos personales y ubicaciones.
* Se proporcionará una guía técnica con instrucciones de instalación, configuración y recuperación ante fallos.
* El sistema deberá poder escalarse para múltiples dispositivos conectados a una sola plataforma.

## Identificación de Actores del Sistema

## En el contexto del sistema CELSOS, se identifican como actores todas las entidades (personas o sistemas) que interactúan con el software o reciben los beneficios del mismo. Cada actor tiene roles y permisos específicos.

# CAPÍTULO 4

# PLANIFICACIÓN DEL PROYECTO

## Definición de Roles de Trabajo

Siguiendo los primeros pasos de LA METODOLOGÍA SCRUM se procede a definir los roles. Los roles son funciones asignadas a los participantes del proyecto y los colaboradores con los que se contará para su desarrollo. En la siguiente tabla mostraremos los principales roles establecidos luego de la primera reunión con el cliente:

Tabla 3. Definiión de roles.

|  |  |
| --- | --- |
| **Roles** | **Responsables** |
| Product Owner | Gerente de … |
| Scrum Master | … |
| Team Member | … |
| Tester | … |

Fuente. Elaboración propia.

### Product owner

Es el Gerente Administrativo de la Empresa Dolphis Consulting; el señor …, el cual es el encargado de ese rol, debido a que representa a la empresa y también el que recibe los requerimientos de los stakeholders.

### Scrum master

El estudiante …, es el encargado de este rol, debido a que cuenta con una mayor experiencia en el campo del desarrollo de aplicativos con metodologías ágiles.

### Team member

Los miembros encargados de cumplir este rol son los estudiantes …; debido a que cuentan con el conocimiento necesario para el desarrollo de aplicaciones de escritorio, modelamiento de bases de datos, gestión de base de datos, etc.

### Tester

Esta tarea será llevada a cabo por el estudiante …, esto debido a que cuenta con el conocimiento necesario para el desarrollo de pruebas de software.

## Product Backlog

El siguiente paso en la **Metodología SCRUM**, es definir los requisitos detallando prioridades y complejidad, conocido como **Product Backlog**. Para ello, se ha enumerado los requerimientos recolectados en presencia del **Product Owner**, durante una segunda reunión donde se definieron detalles; llegando a obtener la siguiente tabla detallada:

Tabla 4. Product Backlog del Sistema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nro.** | **Descripción de Requerimientos** | **Prioridad** | **Descripción** | **Complejidad** |
| R01 | Módulo de acceso login | 1 | Ingresar un nombre de usuario y una clave ya registrada previamente en el sistema. | 1 |
| R02 | Diseño de Base de Datos | 1 | Realizar el diseño del diagrama de base de datos que permita generar la BD para el software en desarrollo. | 2 |
| R03 | Diseño de Interfaces | 2 | Diseñar una interfaz desde donde se pueda acceder a todas las opciones permitidas en el software. | 1 |
| R04 | Módulo de Clientes | 3 | Registrar, modificar y eliminar los datos de los clientes registrados en el sistema. | 3 |
| R05 | Módulo de Productos | 3 | Registrar, modificar y eliminar los datos de los productos registrados en el sistema. | 3 |
| R06 | Módulo de Usuarios | 3 | Registrar, modificar y eliminar los datos de los usuarios registrados en el sistema. | 3 |
| R07 | Módulo de Ventas | 4 | Permite ingresar los datos del cliente vía una interfaz marcada, donde se detalla los productos a vender. | 4 |
| R08 | Módulo de Orden de Ventas | 4 | Permite observar los datos de la venta realizada, marcada en un formato de recibo de ventas. | 4 |
| R09 | Módulo de Proveedores | 4 | Registrar, modificar y eliminar los datos de los proveedores registrados en el sistema. | 4 |
| R10 | Módulo de Reportes | 4 | Genera un reporte donde se muestra las ventas realizadas en días y meses elegidos. | 4 |

Fuente. Elaboración propia.

## Sprint Backlog

Luego de conocer el Product Backlog, la **Metodología SCRUM** menciona que el equipo de desarrollo debe realizar el Sprint Backlog, donde se detalla las tareas a realizar para el cumplimiento de los requerimientos señalados.

Aquí se estima los tiempos de desarrollo, seleccionando los requerimientos y preparándolos en cada Sprints (4 para este proyecto) y sus correspondientes iteraciones; cada requerimiento se dividió en tareas las cuales fueron ordenadas de acuerdo a las necesidades del desarrollo (modelamiento y creación de base de datos, desarrollo de interfaces que sirven para otras interfaces del sistema).

### Sprint 1

La primera iteración cuenta con una duración de **20 días**, se planea realizar reuniones con el **Product Owner**, para proceder a realizar los avances realizados, el equipo de desarrollo realiza reuniones diarias, estas reuniones cuentas con una duración de **20 minutos aproximadamente**, realizadas para conocer los avances de cada tarea asignada para el presente Sprint.

El objetivo para este Sprint es generar las interfaces para los requerimientos elegidos, diseñar e implementar la base de datos, dependiendo de la prioridad establecida en el **Product Backlog**. La elección de requerimientos se realizó durante la Reunión del Sprint Planning Meeting, llevada a cabo entre el Product Owner, Scrum Master y el Scrum Team.

Tabla 5. Requerimientos para Sprint 1.

|  |  |  |
| --- | --- | --- |
| **Nro.** | **Requisitos** | **Descripción** |
| R01 | Módulo de acceso login | Ingresar un nombre de usuario y una clave ya registrada previamente en el sistema. |
| R02 | Diseño de Base de Datos | Realizar el diseño del diagrama de base de datos que permita generar la BD para el software en desarrollo. |
| R03 | Diseño de Interfaces | Diseñar una interfaz desde donde se pueda acceder a todas las opciones permitidas en el software. |

Fuente. Elaboración propia.

### Sprint 2

La segunda iteración cuenta con una duración de **25 días**, se planea realizar reuniones con el **Product Owner**, para proceder a realizar los avances realizados, el equipo de desarrollo realiza reuniones diarias, estas reuniones cuentas con una duración de 20 minutos aproximadamente, realizadas para conocer los avances de cada tarea asignada para el presente Sprint.

El objetivo para este Sprint es generar las interfaces para los requerimientos elegidos, dependiendo de la prioridad establecida en el **Product Backlog**. La elección de requerimientos se realizó durante la Reunión del Sprint Planning Meeting, llevada a cabo entre el Product Owner, Scrum Master y el Scrum Team.

Tabla 6. Requerimientos para Sprint 2.

|  |  |  |
| --- | --- | --- |
| **Nro.** | **Requisitos** | **Descripción** |
| R04 | Módulo de Clientes | Registrar, modificar y eliminar los datos de los clientes registrados en el sistema. |
| R05 | Módulo de Productos | Registrar, modificar y eliminar los datos de los productos registrados en el sistema. |
| R06 | Módulo de Usuarios | Registrar, modificar y eliminar los datos de los usuarios registrados en el sistema. |

Fuente. Elaboración propia.

### Sprint 3

La tercera iteración cuenta con una duración de **25 días**, se planea realizar reuniones con el **Product Owner**, para proceder a realizar los avances realizados, el equipo de desarrollo realiza reuniones diarias, estas reuniones cuentas con una duración de 20 minutos aproximadamente, realizadas para conocer los avances de cada tarea asignada para el presente Sprint.

El objetivo para este Sprint es generar las plantillas para los requerimientos elegidos, dependiendo de la prioridad establecida en el **Product Backlog**. La elección de requerimientos se realizó durante la Reunión del Sprint Planning Meeting, llevada a cabo entre el Product Owner, Scrum Master y el Scrum Team.

Tabla 7. Requerimientos para Sprint 3.

|  |  |  |
| --- | --- | --- |
| **Nro.** | **Requisitos** | **Descripción** |
| R07 | Módulo de Ventas | Permite ingresar los datos del cliente vía una interfaz marcada, donde se detalla los productos a vender. |
| R08 | Módulo de Orden de Ventas | Permite observar los datos de la venta realizada, marcada en un formato de recibo de ventas. |

Fuente. Elaboración propia.

### Sprint 4

La cuarta iteración cuenta con una duración de **30 días**, se planea realizar reuniones con el **Product Owner**, para proceder a realizar los avances realizados, el equipo de desarrollo realiza reuniones diarias, estas reuniones cuentas con una duración de 20 minutos aproximadamente, realizadas para conocer los avances de cada tarea asignada para el presente Sprint.

El objetivo para este Sprint es generar los reportes según los requerimientos elegidos, dependiendo de la prioridad establecida en el **Product Backlog**. La elección de requerimientos se realizó durante la Reunión del Sprint Planning Meeting, llevada a cabo entre el Product Owner, Scrum Master y el Scrum Team.

Tabla 8. Requerimientos para Sprint 4.

|  |  |  |
| --- | --- | --- |
| **Nro.** | **Requisitos** | **Descripción** |
| R09 | Módulo de Proveedores | Registrar, modificar y eliminar los datos de los proveedores registrados en el sistema. |
| R10 | Módulo de Reportes | Genera un reporte donde se muestra las ventas realizadas en días y meses elegidos. |

Fuente. Elaboración propia.

## Planificación de Sprints

Según la **Metodología SCRUM**, nos indica que el grupo de trabajo tiene que decidir y plasmar las historias de usuario, para almacenar las tareas que se necesita realizar para desarrollar el software y cual tiene que ser su criterio de aceptación.

### Historias de usuario

Durante las sesiones de reunión; luego de discutirlo con el equipo de desarrollo y el principal interesado del proyecto, se obtuvo las siguientes historias de usuario:

Tabla 9. Lista de historias de usuario.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Sprint** |
| HU01 | Interfaz de acceso login. |  | 1 |
| HU02 | Código para ingreso login. |  | 1 |
| HU03 | Modelo E/R de la base de datos. |  | 1 |
| HU04 | Modelo lógico de la base de datos. |  | 1 |
| HU05 | Modelo físico de la base de datos. |  | 1 |
| HU06 | Implementar base de datos. |  | 1 |
| HU07 | Interfaz principal (dashboard). |  | 1 |
| HU08 | Código para desplazamiento de menú. |  | 1 |
| HU09 | Interfaz de módulo cliente. |  | 2 |
| HU10 | Registrar de nuevo cliente. |  | 2 |
| HU11 | Modificar cliente registrado. |  | 2 |
| HU12 | Eliminar cliente registrado. |  | 2 |
| HU13 | Listar clientes registrados. |  | 2 |
| HU14 | Interfaz de módulo producto. |  | 2 |
| HU15 | Registrar de nuevo producto. |  | 2 |
| HU16 | Modificar producto registrado. |  | 2 |
| HU17 | Eliminar producto registrado. |  | 2 |
| HU18 | Listar productos registrados. |  | 2 |
| HU19 | Interfaz de módulo usuario. |  | 2 |
| HU20 | Registrar de nuevo usuario. |  | 2 |
| HU21 | Modificar usuario registrado. |  | 2 |
| HU22 | Eliminar usuario registrado. |  | 2 |
| HU23 | Listar usuarios registrados. |  | 2 |
| HU24 | Interfaz de módulo ventas. |  | 3 |
| HU25 | Obtener datos del cliente |  | 3 |
| HU26 | Obtener productos con stock. |  | 3 |
| HU27 | Agregar productos a lista de ventas. |  | 3 |
| HU28 | Eliminar producto de lista de ventas. |  | 3 |
| HU29 | Modificar producto de lista de ventas. |  | 3 |
| HU30 | Registrar nueva venta. |  | 3 |
| HU31 | Plantilla de recibo de venta. |  | 3 |
| HU32 | Rellenar recibo de venta. |  | 3 |
| HU33 | Exportar ventas en otro formato. |  | 3 |
| HU34 | Interfaz de módulo proveedor. |  | 4 |
| HU35 | Registrar de nuevo proveedor. |  | 4 |
| HU36 | Modificar proveedor registrado. |  | 4 |
| HU37 | Eliminar proveedor registrado. |  | 4 |
| HU38 | Listar proveedores registrados. |  | 4 |
| HU39 | Plantilla de reporte de ventas. |  | 4 |
| HU40 | Consulta de reporte de ventas. |  | 4 |
| HU41 | Visualizar reporte de ventas. |  | 4 |

Fuente. Elaboración propia.

## Cronograma de Actividades

Tabla 10. Cronograma de actividades.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **% Completado** | **Nombre de Tarea** | **Duración** | **Comienzo** | **Final** |
| **0%** | **Desarrollo de un Software de Ventas** | **74.5 días** | **12/09/23** | **25/12/23** |
| **0%** | **Fase Análisis** | **6.2 días** | **12/09/23** | **20/09/23** |
| … % | Estudio previo y entrevistas con el cliente | 2 días | 12/09/23 | 13/09/18 |
| … % | Identificación de los requisitos de usuario | 3 días | 12/09/23 | 17/09/23 |
| … % | Identificación de los requisitos de rendimiento y de HW/SW | 2 días | 12/09/23 | 18/09/23 |
| … % | Identificación de los requisitos de la interfaz | 2 días | 12/09/23 | 20/09/23 |
| … % | Especificación de requerimientos del aplicativo | 1 día | 12/09/23 | 13/09/23 |
| …% | Análisis de procesos | 3 días | 14/09/23 | 18/09/23 |
| **100%** | **Fin Fase Análisis** | **0 días** | **18/09/23** | **18/09/23** |
| **0%** | **Fase Diseño y Desarrollo** | **37 días** | **19/09/23** | **08/11/23** |
| **…%** | **Ingreso al sistema** | **11 días** | **19/09/23** | **03/10/23** |
| …% | Elaboración de la administración de usuarios | 2 días | 19/09/23 | 20/09/23 |
| …% | Maquetación del aplicativo | 0 días | 21/09/23 | 24/09/23 |
| …% | Elaboración del modelo lógico y físico de la base de datos | 3 días | 25/09/23 | 27/09/23 |
| …% | Elaboración de consultas para la interfaz de login. | 2 días | 28/09/23 | 02/10/23 |
| …% | Codificación de la interfaz de login. | 1 día | 3/10/23 | 03/10/23 |
| **…%** | **Módulo de usuarios** |  |  |  |
| …% | … | … | … | … |
| **100%** | **Fin Fase Análisis** | **6.2 días** | **12/09/23** | **20/09/23** |
| … | … | … | … | … |

Fuente. Elaboración propia.

## Gestión de Riesgos

Una vez conocida las actividades a realizar durante el proyecto, LA METODOLOGÍA XP indica que tenemos que realizar un pequeño plan de riesgos que se pueden presentar durante el desarrollo del proyecto. Ahora procedemos a listar posibles riesgos que puedan ocurrir para retrasar o hacer fracasar el proyecto:

Tabla 11. Gestión de riesgos del proyecto.

|  |  |
| --- | --- |
| **Riesgos** | **Descripción** |
| El equipo de trabajo tiene que laborar en paralelo en sus funciones diarias. | El avance del proyecto no puede interrumpir las labores diarias; para mitigar el riesgo se atiene a aplicar prioridades. |
| El gerente de la empresa cuenta con una agenda recargada, dificultando la comunicación. | La comunicación y reuniones constantes entre el equipo de trabajo y el product owner, es una fortaleza de SCRUM; para mitigar el riesgo se solicita coordinar los posibles días que se podrá tener una reunión con su presencia. |
| La duración del proyecto afecta a los requisitos del Cliente. | La empresa puede decidir una fecha tentativa distinta a la que se establece para el arranque del producto; para mitigar este riesgo, se informa a la Empresa sobre los avances realizados a fin de demostrar que se cumplirá con la fecha establecida. |
| El tiempo para la aprobación de un avance es muy extenso. | Esto afecta de modo negativo a la culminación del proyecto en el tiempo establecido; para mitigar el riesgo, se coordina con el Product Owner para la revisión oportuna y enviarlos con anticipación. |
| En la actualidad, la empresa se encuentra en remodelación. | El ruido generado debido a la remodelación y cambios que se aproximan en la empresa, incomodan al proceso de desarrollo del proyecto; para mitigar el riesgo, se coordina con la Empresa para establecer el momento en que la remodelación no afecte al desarrollo del proyecto. |

Fuente. Elaboración propia.