| DUOC UC - Escuela de informática y telecomunicaciones |
| --- |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto: EarthAlert* |
|  |
| **Revisión*: [01]*** |
| **[30-08-2024]** |

| Planificación y Especificación de Requisitos según estándares; IEEE 830, ISO9000 y PMI. |
| --- |

**Contenido**

[*DUOC UC - Escuela de informática y telecomunicaciones 1*](#_heading=h.gjdgxs)

[**Ficha del documento 4**](#_heading=h.1fob9te)

[**1. Introducción 5**](#_heading=h.3znysh7)

[1.1.](#_heading=h.2et92p0) Propósito 5

[1.2.](#_heading=h.tyjcwt) Ámbito del Sistema 5

[1.3.](#_heading=h.3dy6vkm) Definiciones, Acrónimos y Abreviaturas 5

[1.4.](#_heading=h.1t3h5sf) Referencias 5

[1.5.](#_heading=h.4d34og8) Visión General del Documento 5

[**2.**](#_heading=h.2s8eyo1) **Descripción General 6**

[2.1.](#_heading=h.17dp8vu) Perspectiva del Producto 6

[2.2.](#_heading=h.3rdcrjn) Funciones del Producto 6

[2.3.](#_heading=h.26in1rg) Características de los Usuarios 6

[2.4.](#_heading=h.lnxbz9) Restricciones 6

[2.5.](#_heading=h.35nkun2) Suposiciones y Dependencias 7

[2.6.](#_heading=h.1ksv4uv) Requisitos Futuros 7

[**3.**](#_heading=h.44sinio) **Requisitos Específicos 8**

[3.1](#_heading=h.2jxsxqh) Requisitos comunes de las interfaces 8

[*3.1.1*](#_heading=h.z337ya) *Interfaces de usuario 8*

[*3.1.2*](#_heading=h.3j2qqm3) *Interfaces de hardware 8*

[*3.1.3*](#_heading=h.1y810tw) *Interfaces de software 8*

[3.2](#_heading=h.4i7ojhp) Requisitos funcionales 9

[3.3](#_heading=h.2xcytpi) Requisitos no funcionales (Organización) y de calidad (Producto) 9

[*3.3.1*](#_heading=h.1ci93xb) *Requisitos de Rendimiento o Eficiencia 9*

[*3.3.2*](#_heading=h.3whwml4) *Requisitos de Seguridad 9*

[*3.3.3*](#_heading=h.2bn6wsx) *Requisitos de Usabilidad 9*

[*3.3.4*](#_heading=h.qsh70q) *Requisitos de Disponibilidad 9*

[*3.3.5*](#_heading=h.3as4poj) *Requisitos de Portabilidad 9*

[*3.3.6*](#_heading=h.1pxezwc) *Requisitos de Mantenibilidad 9*

[*3.3.7*](#_heading=h.49x2ik5) *Requisitos de Funcionalidad 9*

[3.4](#_heading=h.2p2csry) Requisitos No funcionales Organizacionales 10

[**4. Propuesta de Planificación 11**](#_heading=h.147n2zr)

[4.1 Descripción general acerca de la Planificación 11](#_heading=h.3o7alnk)

[*4.1.2 Definición del Equipo de Trabajo 11*](#_heading=h.23ckvvd)

[*4.1.3 Definición de Actividades principales del Proyecto 11*](#_heading=h.ihv636)

[*4.1.4 Resumen Costos del Desarrollo del Proyecto 11*](#_heading=h.32hioqz)

[*4.1.6 Carta Gantt 11*](#_heading=h.1hmsyys)

[5. Anexos 12](#_heading=h.41mghml)

[*5.1 Acta de Proyecto 12*](#_heading=h.2grqrue)

[*5.2 Matriz Especificación de Requerimientos 12*](#_heading=h.vx1227)

[*5.3 Prototipado de Software 12*](#_heading=h.3fwokq0)

[*5.4 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo 12*](#_heading=h.1v1yuxt)

[*5.5 Planilla Carta Gantt 12*](#_heading=h.4f1mdlm)

# Ficha del documento

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
| *30-08-2024* | *1* | *Nicolas Palma y Miguel Montenegro* | *Introducción* |
|  |  |  |  |

Documento validado por las partes en fecha:

**Integrantes:**

| **Nombre Integrante del Equipo** | **Rol Definido** |
| --- | --- |
| [NICOLAS IGNACIO PALMA MARIN](mailto:ni.palmam@duocuc.cl) | *Scrum master/Desarrollador* |
| *Miguel Montenegro Sanchesz* | *Product owner/Desarrollador* |
|  |  |
|  |  |
|  |  |

# 1. Introducción

Este documento de Especificación de Requisitos de Software (ERS) describe los requisitos para el desarrollo de la aplicación web y móvil "EarthAlert". El documento está dirigido a los stakeholders del proyecto, incluyendo al equipo de desarrollo, los sponsors y los usuarios finales.

## 1.1. Propósito

El propósito de este documento ERS es proporcionar una descripción detallada y completa de los requisitos para el desarrollo de la aplicación web y móvil EarthAlert. Este documento servirá como base para la comunicación entre los stakeholders del proyecto y como guía para el diseño, desarrollo, pruebas e implementación del sistema.

## 1.2. Ámbito del Sistema

Nombre del Sistema: EarthAlert

Descripción: EarthAlert es una aplicación web y móvil diseñada para proporcionar alertas tempranas de sismos a los usuarios en tiempo real. El sistema se conectará a redes de sensores sísmicos y fuentes de datos oficiales para obtener información sobre la actividad sísmica. Mediante algoritmos avanzados, EarthAlert determinará la ubicación, magnitud y tiempo estimado de llegada del sismo a la ubicación del usuario. Una vez detectado un sismo, la aplicación enviará una alerta inmediata a los usuarios, proporcionándoles información crítica y guías de acción para protegerse.

Beneficios:

* Proveer a la población con una herramienta de alerta temprana para sismos, permitiéndoles tomar medidas preventivas y reducir el riesgo de daños y lesiones.
* Mejorar la capacidad de respuesta ante desastres naturales.
* Brindar información relevante y confiable sobre la actividad sísmica en tiempo real.

Objetivos:

* Desarrollar una aplicación web y móvil funcional que cumpla con los requisitos especificados.
* Integrar la aplicación con redes de sensores sísmicos y fuentes de datos oficiales.
* Implementar algoritmos de detección y predicción de sismos.
* Diseñar una interfaz de usuario intuitiva y fácil de usar.

Alcances:

* El sistema se enfocará en proporcionar alertas tempranas de sismos.
* La precisión de las alertas dependerá de la calidad y disponibilidad de los datos sísmicos.
* El sistema no será responsable por la toma de decisiones de los usuarios en caso de un sismo.

## 1.3. Definiciones, Acrónimos y Abreviaturas

| **Término** | **Descripción** |
| --- | --- |
| ERS | Especificación de Requisitos de Software |
| App | Aplicación |
| IoT | Internet de las Cosas |
| API | Interfaz de Programación de Aplicaciones |
| UI | Interfaz de Usuario |
| UX | Experiencia de Usuario |

## 1.4. Referencias

En esta subsección se mostrará una lista completa de todos los documentos referenciados en la ERS.

## 1.5. Visión General del Documento

Este documento ERS está organizado de la siguiente manera:

Sección 1: Introducción: Presenta el propósito, alcance, definiciones, referencias y visión general del documento.

Sección 2: Descripción General: Describe el contexto del producto, las funciones principales, los usuarios, las restricciones, las suposiciones y los requisitos futuros.

Sección 3: Requisitos Específicos: Detalla los requisitos funcionales y no funcionales del sistema, incluyendo las interfaces de usuario, hardware y software.

Sección 4: Propuesta de Planificación: Describe la planificación del proyecto, incluyendo el equipo de trabajo, las actividades principales, los costos y la carta Gantt.

Sección 5: Anexos: Incluye documentos adicionales relevantes para la especificación de requisitos.

# 2. Descripción General

Esta sección describe el contexto general del producto EarthAlert, incluyendo su perspectiva dentro del ecosistema de aplicaciones móviles, sus funciones principales, las características de sus usuarios, las restricciones que podrían afectar su desarrollo, las suposiciones y dependencias consideradas, y las posibles mejoras o características que se podrían incorporar en el futuro.

## 2.1. Perspectiva del Producto

EarthAlert se concibe como una aplicación móvil independiente que se integra con fuentes de datos externas para proporcionar alertas tempranas de sismos. No depende de otros productos de software para su funcionamiento básico. Sin embargo, se beneficiará de la integración con:

* Redes de sensores sísmicos: Para obtener datos en tiempo real sobre la actividad sísmica.
* Agencias gubernamentales y centros de monitoreo sísmico: Para acceder a información oficial sobre sismos y alertas.
* Plataformas de mensajería y redes sociales: Para facilitar la difusión de alertas y la comunicación entre usuarios.

## 2.2. Funciones del Producto

EarthAlert ofrecerá las siguientes funciones principales:

* Monitoreo en tiempo real de la actividad sísmica: La aplicación se conectará constantemente a fuentes de datos sísmicos para detectar eventos sísmicos.
* Alertas tempranas personalizadas: Los usuarios recibirán notificaciones push en sus dispositivos móviles con información sobre la magnitud del sismo, la ubicación del epicentro y el tiempo estimado de llegada de las ondas sísmicas a su ubicación.
* Visualización de mapas interactivos: La aplicación mostrará mapas con la ubicación del epicentro, el alcance del sismo y la ubicación del usuario.
* Información sobre seguridad y medidas de prevención: EarthAlert proporcionará a los usuarios información sobre cómo prepararse para un sismo, qué hacer durante un evento sísmico y cómo actuar después de un terremoto.
* Historial de sismos: Los usuarios podrán acceder a un registro de sismos recientes, incluyendo su magnitud, ubicación y fecha.
* Sistema de donaciones: Los usuarios podrán donar a organizaciones de ayuda ante desastres

## 2.3. Características de los Usuarios

El público objetivo de EarthAlert es amplio e incluye a cualquier persona que viva, trabaje o visite zonas propensas a sismos. Se espera que los usuarios tengan diferentes niveles de familiaridad con la tecnología y la prevención de desastres.

Tipos de Usuarios:

* Público en general: Usuarios sin conocimientos técnicos específicos que buscan una herramienta confiable para recibir alertas tempranas de sismos.
* Turistas y visitantes: Personas que visitan zonas propensas a sismos y necesitan información sobre seguridad y alertas en tiempo real.
* Autoridades y organismos de gestión de desastres: Usuarios que pueden utilizar EarthAlert como una herramienta complementaria para el monitoreo de sismos y la respuesta ante emergencias.

## 2.4. Restricciones

El desarrollo de EarthAlert se enfrentará a las siguientes restricciones:

* Disponibilidad y precisión de los datos sísmicos: La eficacia del sistema depende de la calidad y la rapidez de los datos proporcionados por las redes de sensores sísmicos y las agencias gubernamentales.
* Conectividad a Internet: La aplicación requiere una conexión a Internet estable para funcionar correctamente. En áreas con cobertura limitada o durante un corte de energía, la funcionalidad de la aplicación podría verse afectada.
* Autonomía del dispositivo móvil: El uso intensivo de la aplicación, especialmente la recepción frecuente de notificaciones push, puede afectar la duración de la batería del dispositivo móvil.
* Cumplimiento de las políticas de privacidad: La aplicación debe cumplir con las leyes y regulaciones de privacidad de datos, especialmente en lo que respecta a la ubicación del usuario.

## 2.5. Suposiciones y Dependencias

El desarrollo de EarthAlert se basa en las siguientes suposiciones:

* Disponibilidad de APIs: Se asume que las redes de sensores sísmicos y las agencias gubernamentales relevantes proporcionarán APIs públicas para acceder a sus datos en tiempo real.
* Adopción de la tecnología móvil: Se asume que el público objetivo tiene acceso a dispositivos móviles con conexión a Internet y está familiarizado con el uso de aplicaciones móviles.

## 2.6. Requisitos Futuros

Se prevé que EarthAlert incorpore las siguientes características en futuras versiones:

* Integración con dispositivos inteligentes: La aplicación podría conectarse con dispositivos domésticos inteligentes para activar medidas de seguridad automatizadas en caso de un sismo, como el apagado automático del gas o la apertura de puertas.
* Funcionalidades de comunidad: Se podrían incorporar funciones que permitan a los usuarios reportar daños, solicitar ayuda o conectarse con otros usuarios en la zona afectada después de un sismo.
* Personalización avanzada: Los usuarios podrían personalizar las alertas en función de la magnitud del sismo, la distancia al epicentro y sus propias preferencias.
* Soporte multilingüe: Para llegar a un público más amplio, la aplicación se traducirá a varios idiomas.

# 3. Requisitos Específicos

Esta sección describe con mayor detalle los requisitos específicos, tanto funcionales como no funcionales, que deberá cumplir la aplicación EarthAlert.

## 3.1 Requisitos comunes de las interfaces

En esta sección se definen los requisitos para las interfaces de usuario, hardware y software de la aplicación EarthAlert.

### 3.1.1 Interfaces de usuario

Requisitos generales:

* Diseño intuitivo y fácil de usar: La interfaz de usuario (UI) de EarthAlert debe ser intuitiva y fácil de usar para usuarios con diferentes niveles de experiencia tecnológica.
* Diseño responsive: La UI debe ser responsive y adaptarse a diferentes tamaños de pantalla, incluyendo teléfonos inteligentes y tabletas.
* Claridad en la información: La información mostrada en la UI debe ser clara, concisa y fácil de entender.
* Accesibilidad: La UI debe cumplir con los estándares de accesibilidad para personas con discapacidades.

Pantallas principales:

* Pantalla de inicio: Mostrará un mapa con la ubicación actual del usuario y un resumen de la actividad sísmica reciente. También permitirá acceder a las diferentes secciones de la aplicación.
* Pantalla de alertas: Mostrará una lista de las alertas de sismos recibidas, incluyendo la magnitud, la ubicación y el tiempo estimado de llegada.
* Pantalla de detalles del sismo: Proporcionará información detallada sobre un sismo específico, incluyendo la magnitud, la profundidad, el epicentro, un mapa del área afectada y consejos de seguridad.
* Pantalla de configuración: Permitirá a los usuarios personalizar la configuración de la aplicación, como el idioma, las unidades de medida, la distancia de las alertas y las notificaciones.

Elementos de la UI:

* Mapas: Los mapas utilizados en la aplicación deben ser claros, precisos y fáciles de navegar. Se utilizarán iconos y colores intuitivos para representar la ubicación del usuario, el epicentro del sismo y el alcance del mismo.
* Notificaciones: Las notificaciones push se utilizarán para alertar a los usuarios sobre sismos en tiempo real. Las notificaciones serán personalizables y permitirán a los usuarios ajustar la frecuencia, el sonido y la vibración.
* Botones y menús: Los botones y menús deben ser claros, concisos y fáciles de entender. Se utilizarán iconos y etiquetas de texto para identificar las diferentes acciones.

### 3.1.2 Interfaces de hardware

* Sensores del dispositivo: La aplicación utilizará los sensores del dispositivo móvil, como el GPS y el acelerómetro, para determinar la ubicación del usuario y detectar vibraciones.
* Conexión a Internet: La aplicación requerirá una conexión a Internet estable, ya sea a través de datos móviles o Wi-Fi, para funcionar correctamente.

### 3.1.3 Interfaces de software

* APIs de datos sísmicos: La aplicación se integrará con APIs de redes de sensores sísmicos y agencias gubernamentales para obtener información en tiempo real sobre la actividad sísmica.
* Servicios de notificaciones push: Se utilizarán servicios de notificaciones push, como Google Firebase Cloud Messaging o Apple Push Notification Service, para enviar alertas a los usuarios.
* Plataformas de mapas: La aplicación se integrará con plataformas de mapas, como Google Maps o Apple Maps, para mostrar la ubicación del usuario, el epicentro del sismo y el alcance del mismo.

## 3.2 Requisitos funcionales

Los requisitos funcionales deben estar enumerados (diagrama caso uso) y pueden ser divididos en sub-secciones.

3.2.1 Permitir al usuario configurar el tipo de alertas que desea recibir (por magnitud, ubicación, etc.).

3.2.4 Permitir al usuario registrar contactos de emergencia y acceder a ellos rápidamente desde la aplicación.

3.2.5 Un botón que, al ser presionado, envía una alerta a los contactos de emergencia con la ubicación del usuario.

3.2.6 Mostrar al usuario recomendaciones de seguridad periodicamente

3.2.7 Permitir a los usuarios comunicarse entre sí, compartir información y reportar daños.

3.2.9 Proporcionar acceso a información educativa sobre sismos, medidas de prevención y planes de emergencia.

3.2.12 Permitir a los usuarios acceder a un historial de sismos

3.2.13 Integrar un sistema para que los usuarios puedan realizar donaciones a organizaciones de ayuda en caso de terremotos.

3.2.14 Permitir a los usuarios buscar sismos históricos por fecha, magnitud, ubicación y otros criterios.

3.2.15 Mostrar las placas tectónicas en el mapa para contexto geológico.

3.2.17 Permitir el acceso a funcionalidades básicas de la aplicación (como contactos de emergencia y recomendaciones) sin conexión a internet.

3.2.19 Permitir a los usuarios cambiar el idioma de la interfaz de la aplicación.

3.2.20 Mostrar un tutorial interactivo al usuario la primera vez que utiliza la aplicación para explicar las funciones principales.

3.2.21 Permitir a los usuarios enviar comentarios y sugerencias a los desarrolladores de la aplicación.

## 3.3 Requisitos no funcionales (Organización) y de calidad (Producto)

### 3.3.1 Requisitos de Rendimiento o Eficiencia

R-22 Rendimiento: La aplicación debe ser rápida y responsiva, incluso bajo una carga de usuarios alta.

* El tiempo de respuesta para mostrar las alertas de sismos no debe superar los 2 segundos.
* El tiempo de carga de los mapas y la información del sismo no debe superar los 5 segundos.

R-26 Escalabilidad: La aplicación debe ser capaz de manejar un gran número de usuarios y datos sin afectar negativamente su rendimiento.

* El sistema debe ser capaz de enviar notificaciones push a miles de usuarios simultáneamente.
* La base de datos debe estar optimizada para manejar grandes volúmenes de datos de sismos.

### 3.3.2 Requisitos de Seguridad

R-24 Seguridad: La aplicación debe proteger la información del usuario y la integridad del sistema contra accesos no autorizados, uso indebido y modificaciones maliciosas.

R-30 Seguridad de Datos: Los datos de los usuarios, como su ubicación y configuración, deben estar protegidos contra accesos no autorizados.

* Cifrado de datos: Los datos sensibles deben cifrarse durante el almacenamiento y la transmisión.
* Control de acceso: Implementar roles de usuario y permisos para restringir el acceso a la información sensible.

R-34 Backup y Recuperación: Se debe contar con un sistema de backup y recuperación de datos para prevenir la pérdida de información en caso de un fallo del sistema.

* Backups periódicos: Realizar copias de seguridad de la base de datos y archivos importantes de forma regular.
* Plan de recuperación ante desastres: Definir un plan para restaurar la aplicación y los datos en caso de un fallo catastrófico.

### 3.3.3 Requisitos de Usabilidad

R-25 Usabilidad: La aplicación debe ser fácil de usar e intuitiva para usuarios con diferentes niveles de experiencia tecnológica.

* La interfaz de usuario debe ser clara, concisa y fácil de navegar.
* Se debe proporcionar información de ayuda y tutoriales para guiar a los usuarios en las funciones de la aplicación.

### 3.3.4 Requisitos de Disponibilidad

R-23 Disponibilidad: La aplicación debe estar disponible 24/7, con un tiempo de inactividad mínimo.

* Se debe implementar un sistema de monitoreo para detectar y alertar sobre cualquier problema que afecte la disponibilidad de la aplicación.
* Se debe contar con un plan de recuperación ante desastres para restaurar el servicio lo antes posible en caso de una interrupción.

### 3.3.5 Requisitos de Portabilidad

R-29 Portabilidad: La aplicación debe ser compatible con diferentes plataformas y dispositivos móviles, incluyendo Android e iOS.

* Diseño responsivo: La interfaz de usuario debe adaptarse a diferentes tamaños de pantalla (teléfonos inteligentes, tablets).
* Se deben realizar pruebas en diferentes dispositivos y versiones de sistemas operativos para garantizar la compatibilidad.

### 3.3.6 Requisitos de Mantenibilidad

R-28 Mantenibilidad: El código fuente de la aplicación debe ser fácil de entender, modificar y mantener para facilitar futuras actualizaciones y mejoras.

* Código limpio y modular: Utilizar buenas prácticas de programación para un código legible y organizado.
* Documentación: El código debe estar documentado para facilitar su comprensión y modificación.

R-36 Documentación del Sistema: Se debe documentar la arquitectura, el código y la configuración de la aplicación para facilitar su comprensión y mantenimiento.

* Documentación técnica: Describir la arquitectura, el diseño y el código de la aplicación.
* Documentación de usuario: Proporcionar una guía de usuario para ayudar a los usuarios a utilizar la aplicación.

R-37 Gestión de configuración: Se debe gestionar y controlar las versiones del código y la configuración de la aplicación para facilitar el seguimiento de los cambios y la resolución de problemas.

* Control de versiones: Utilizar un sistema de control de versiones como Git para gestionar el código fuente.
* Gestión de la configuración: Utilizar herramientas para gestionar y controlar la configuración de la aplicación.

## 3.4 Requisitos No funcionales Organizacionales

R-27 Fiabilidad: La aplicación debe funcionar correctamente y sin errores, proporcionando información precisa y confiable a los usuarios.

* Manejo de errores: La aplicación debe capturar y manejar errores de manera adecuada, mostrando mensajes informativos al usuario.
* Pruebas exhaustivas: Se deben realizar pruebas unitarias, de integración y de sistema para garantizar la calidad del software.

R-31 Privacidad: La aplicación debe cumplir con las leyes y regulaciones de privacidad de datos, protegiendo la información personal de los usuarios.

* Política de privacidad: Definir claramente cómo se recopilan, utilizan, almacenan y comparten los datos de los usuarios.
* Consentimiento del usuario: Obtener el consentimiento explícito del usuario antes de recopilar o utilizar cualquier dato personal.

R-32 Internacionalización: La aplicación debe ser adaptable a diferentes idiomas y regiones para llegar a un público global.

* Traducción de la interfaz: La aplicación debe estar disponible en varios idiomas.
* Formato de fecha y hora: Adaptar el formato de fecha y hora según la región del usuario.

R-33 Accesibilidad: La aplicación debe ser accesible para personas con discapacidades, siguiendo los estándares de accesibilidad web.

* Cumplimiento de WCAG: La interfaz de usuario debe cumplir con las Pautas de Accesibilidad al Contenido Web (WCAG).
* Opciones de accesibilidad: Proporcionar opciones para ajustar el tamaño del texto, el contraste y la navegación por teclado.

R-35 Monitorización: Se debe monitorizar el rendimiento y la disponibilidad de la aplicación para detectar y solucionar problemas de forma proactiva.

* Herramientas de monitorización: Utilizar herramientas para rastrear el tiempo de actividad, el rendimiento y los errores de la aplicación.
* Alertas: Configurar alertas para notificar a los administradores sobre problemas críticos.

# 4. Propuesta de Planificación

Esta sección describe la planificación del proyecto EarthAlert, incluyendo la descripción general, el equipo de trabajo, las actividades principales, los costos de desarrollo y la carta Gantt.

## 4.1 Descripción general acerca de la Planificación

La planificación del proyecto EarthAlert se basa en un enfoque iterativo e incremental, utilizando metodologías ágiles para el desarrollo del software. Se busca obtener una versión funcional de la aplicación en un corto plazo, para luego ir incorporando nuevas funcionalidades y mejoras en iteraciones posteriores.

El proyecto se divide en cuatro fases principales:

* Inicio: Definición de los requisitos, el alcance del proyecto y la creación del prototipo inicial.
* Desarrollo: Implementación del código fuente, pruebas unitarias e integración de componentes.
* Pruebas y Despliegue: Pruebas de sistema, pruebas de usuario y lanzamiento de la aplicación en las tiendas de aplicaciones.
* Mantenimiento: Corrección de errores, actualizaciones de seguridad y desarrollo de nuevas funcionalidades.

### 4.1.2 Definición del Equipo de Trabajo

El equipo de desarrollo de EarthAlert está compuesto por:

* Nicolás Palma: Desarrollador Full-Stack, responsable del desarrollo del backend, la integración de APIs y la implementación de la base de datos.
* Miguel Montenegro: Desarrollador Full-Stack, responsable del diseño y desarrollo de la interfaz de usuario, la experiencia de usuario y las notificaciones push.
* Profesor Fabian Saldaño: Tutor del proyecto, encargado de la supervisión general, la asesoría técnica y la validación de las soluciones implementadas.

### 4.1.3 Roadmap

Fase 1: Inicio (2 semanas)

* A1.1 Definición detallada de requisitos y casos de uso (Nicolás Palma, Miguel Montenegro).
* A1.2 Diseño de la arquitectura de la aplicación (Nicolás Palma).
* A1.3 Diseño de la interfaz de usuario y prototipo interactivo (Miguel Montenegro).
* A1.4 Configuración del entorno de desarrollo y repositorio de código (Nicolás Palma).

Fase 2: Desarrollo (6 semanas)

* A2.1 Desarrollo del backend y la API REST (Nicolás Palma).
* A2.2 Desarrollo de la base de datos y la lógica de acceso a datos (Nicolás Palma).
* A2.3 Desarrollo de la interfaz de usuario y la lógica de presentación (Miguel Montenegro).
* A2.4 Integración de APIs de datos sísmicos (Nicolás Palma).
* A2.5 Implementación de notificaciones push (Miguel Montenegro).
* A2.6 Pruebas unitarias y de integración (Nicolás Palma, Miguel Montenegro).

Fase 3: Pruebas y Despliegue (2 semanas)

* A3.1 Pruebas de sistema y de usuario (Nicolás Palma, Miguel Montenegro, Profesor Fabian Saldaño).
* A3.2 Corrección de errores y mejoras (Nicolás Palma, Miguel Montenegro).
* A3.3 Preparación para el lanzamiento en las tiendas de aplicaciones (Miguel Montenegro).
* A3.4 Lanzamiento de la aplicación (Miguel Montenegro).

Fase 4: Mantenimiento (Continuamente)

* A4.1 Monitoreo del rendimiento y la estabilidad de la aplicación (Nicolás Palma).
* A4.2 Corrección de errores y problemas reportados por los usuarios (Nicolás Palma, Miguel Montenegro).
* A4.3 Desarrollo de nuevas funcionalidades y mejoras (Nicolás Palma, Miguel Montenegro).

### 4.1.4 Resumen Costos del Desarrollo del Proyecto

| **Descripción** | **Costo (CLP)** |
| --- | --- |
| Hardware y Software | 5.000.000 |
| Desarrollo | 30.000.000 |
| Pruebas y Despliegue | 5.000.000 |
| Mantenimiento (1 año) | 10.000.000 |
| **Total** | **50.000.000** |

### 4.1.6 Sprint Backlog

Agregar las fechas más importantes definidas en la carta Gantt dejando claro los hitos o las fases principales y la fecha de término del proyecto

## 5. Anexos

### 5.1 Acta de Proyecto

[Acta de constitucion de proyecto](https://docs.google.com/document/d/1VZ-D1hnjPrYe_1VILUcS7VKc3NXa_I_Z/edit?usp=sharing&ouid=114227852491417361906&rtpof=true&sd=true)

### 5.2 Matriz Especificación de Requerimientos

[historias de usuario](https://docs.google.com/spreadsheets/d/1Lku7Lz89zR5CUy0BcNgaTzVRNd-af3Ut/edit?usp=sharing&ouid=114227852491417361906&rtpof=true&sd=true)

### 5.3 Prototipado de Software

[mockups](https://docs.google.com/document/d/1ogKkemMC0PeTYKXXRbw_ckw9QNHxvNcH/edit?usp=sharing&ouid=114227852491417361906&rtpof=true&sd=true)

### 5.4 Planilla Carta Gantt / Sprint Backlog

[Sprint Backlog](https://docs.google.com/spreadsheets/d/1-356kES37dYCgksxPU32WWdVVJ1xJQ2_/edit?usp=sharing&ouid=114227852491417361906&rtpof=true&sd=true)