

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática



Plan de proyecto: Sistema de monitoreo, control y alerta sísmológico (QuakeSafe)

Autor(es): Tihare Cabello

Liliana Galvez

Cristian Huanca

Byron Santibáñez

Asignatura: Proyecto II

Profesor: Diego Aracena Pizarro

ARICA, 02 DICIEMBRE 2024

Historial de Cambios

Tabla 1. Historial de cambios

Fecha	Versión	Descripción	Autor(es)
13/08/2024	1.0	Versión preliminar del formato	Liliana Galvez Byron Santibáñez
04/09/2024	1.1	Desarrollo del Panorama general	Tihare Cabello
24/09/2024	1.2	Desarrollo de referencias, organización del informe	Tihare Cabello Liliana Galvez Byron Santibáñez Cristian Huanca
25/09/2024	1.3	Desarrollo de Planificación de procesos de gestión	Tihare Cabello Liliana Galvez Byron Santibáñez Cristian Huanca
29/09/2024	1.4	Finalización de Fase 1	Tihare Cabello Liliana Galvez Byron Santibáñez Cristian Huanca
30/10/2024	1.5	Desarrollo de Arquitectura Cliente-Servidor	Tihare Cabello Liliana Galvez Byron Santibáñez Cristian Huanca
04/11/2024	1.6	Finalización Casos de uso	Tihare Cabello Liliana Galvez Byron Santibáñez Cristian Huanca
02/12/2024	1.7	Realización de Implementación	Tihare Cabello Liliana Galvez Byron Santibáñez Cristian Huanca

Índice de contenido

Índice de contenido	3
Índice de tablas	5
Índice de Ilustraciones	6
Resumen	7
Abstract	7
Panorama General	8
Introducción.....	8
Propósito.....	9
Alcance.....	9
Objetivos.....	9
General.....	9
Específicos.....	9
Suposiciones.....	10
Respecto a nosotros.....	10
Respecto al usuario.....	10
Restricciones.....	10
Entregables.....	11
Organización del proyecto	11
Personal y entidades internas.....	11
Roles y responsabilidades.....	12
Mecanismos de comunicación.....	13
Planificación de los procesos de gestión	13
Planificación de estimaciones.....	13
Planificación de recursos humanos.....	15
Planificación de costo total.....	15
Lista de actividades (Carta Gantt)	16
Actividades de trabajo.....	16
Asignación de tiempo.....	17
Planificación de la gestión de riesgos	18
Planificación de Procesos Técnicos	21
Modelo de Diseño (Caso de uso general).....	21
Diagrama de Clases de Caso de Uso General.....	22
Casos de uso.....	23
C.U.S: Detectar Sismo.....	23
Diagrama de secuencia: “Detectar Sismo”.....	24
C.U.S: Notificar Alerta de Sismo.....	25
Diagrama de secuencia: “Notificar Alerta de Sismo”.....	25

C.U.S: Cortar Suministro Eléctrico.....	26
Diagrama de secuencia: “Cortar Suministro Eléctrico”	27
C.U.S: Activar Luces de Emergencia.....	28
Diagrama de Secuencia: “Activar Luces de Emergencia”	29
C.U.S: Desbloquear Puertas de Emergencia.....	30
Diagrama de secuencia: “Desbloquear Puertas de Emergencia”	31
C.U.S: Activar / Desactivar Sistema Manualmente.....	32
Diagrama de Secuencia: “Activar / Desactivar Sistema Manualmente”	33
Descripción de la Arquitectura.....	34
Herramientas y técnicas.....	36
Diseño de Interfaz de Usuario.....	37
Planificación de procesos de soporte.....	39
Planificación de documentación.....	39
Problemas encontrados.....	40
Problema con librería de GPIO.....	40
Solución.....	40
Problema con la conexión remota.....	40
Solución.....	40
Problemas con la conexión de relé.....	40
Solución.....	40
Problema con conexiones de pines.....	41
Solución.....	41
Implementación.....	42
Plan de integración.....	42
Módulo de Interfaz.....	43
Módulo de Sensor de vibración.....	46
Módulo de Relé.....	47
Módulo de Luces de emergencia.....	48
Módulo de ServoMotores.....	49
Conclusiones.....	50
Trabajo futuro.....	51
Optimización de Sensores.....	51
Ampliación del Sistema.....	51
Mejoras en la Interfaz de Usuario (UI/UX).....	51
Compatibilidad Multiplataforma.....	51
Estudio y Optimización Energética.....	51
Escalabilidad del Proyecto.....	51
Documentación y Capacitación.....	51
Referencias.....	52

Índice de tablas

Tabla 1. Historial de cambios.....	2
Tabla 2. Roles	12
Tabla 3. Responsabilidades.....	12
Tabla 4. Recursos Hardware.....	13
Tabla 5. Recurso Software.....	14
Tabla 6. Recursos Humanos.....	15
Tabla 7. Costo Total.....	15
Tabla 8. Actividades.....	16
Tabla 9. Estimación.....	18
Tabla 10. Ocurrencia.....	19
Tabla 11. Riesgos.....	19
Tabla 12. C.U.S Detectar Sismo.....	23
Tabla 13. C.U.S Notificar Alerta de Sismo.....	25
Tabla 14. C.U.S Cortar Suministro Eléctrico.....	26
Tabla 15. C.U.S Activar Luces de Emergencia.....	28
Tabla 16. C.U.S Desbloquear Puertas de Emergencia.....	30
Tabla 17. C.U.S Activar / Desactivar Sistema Manualmente.....	32

Índice de Ilustraciones

Figura 1. Carta Gantt.....	17
Figura 2. Caso de uso General.....	21
Figura 3. Diagrama de Clases.....	22
Figura 4. Diagrama de secuencia C.U.S Detectar Sismo.....	24
Figura 5. Diagrama de secuencia C.U.S Notificar Alerta de Sismo.....	25
Figura 6. Diagrama de secuencia C.U.S Cortar Suministro Eléctrico.....	27
Figura 7. Diagrama de secuencia C.U.S Activar luces de emergencia.....	29
Figura 8. Diagrama de secuencia C.U.S Desbloquear puertas de Emergencia.....	31
Figura 9. Diagrama de secuencia C.U.S Activar / Desactivar Sistema Manualmente.....	33
Figura 10. Arquitectura Cliente-Servidor.....	34
Figura 11. Interfaz de Usuario Control.....	37
Figura 12. Interfaz de Usuario Estado.....	38
Figura 13. Módulo Interfaz 1.....	43
Figura 14. Módulo Interfaz 2.....	44
Figura 15. Interfaz de Usuario 3.....	45
Figura 16. Módulo Sensor de vibración.....	46
Figura 17. Módulo Relé.....	47
Figura 18. Módulo Luces de emergencia.....	48
Figura 19. Módulo ServoMotors.....	49

Resumen

Chile es uno de los países más sísmicos del mundo, por lo que los ciudadanos son una población vulnerable que continuamente sufre procesos geofísicos que pueden provocar desastres sicionaturales, y uno de los más usuales serían los terremotos.

Pero, los sismos no solo producen movimientos irregulares en la arquitectura del país, sino, provoca cortocircuitos, debido a que, dependiendo del grado que tenga dicho movimiento o la calidad que tenga la infraestructura, puede provocar incendios y apagones, los que provocan pánico sobre la mayoría de la población. Y cabe mencionar que los sismos pueden producir obstrucciones de salidas o entradas de las viviendas por tales movimientos irregulares y anormales que las estructuras no pueden soportar.

Por medio del Raspberry Pi y la tecnología IoT, se desarrollara un software que implementa lo mencionado anteriormente, con un apartado capaz de controlar las puertas internas del recinto, igualmente como las luces de emergencia implementadas en los marcos de las puertas, monitorear el corte automático del suministro de electricidad del recinto, y, finalmente, alertar sobre el sismo ocurrido en la localidad.

Abstract

Chile is one of the most seismic countries in the world, making its citizens a vulnerable population that continuously experiences geophysical processes that can lead to socio-natural disasters, one of the most common being earthquakes.

However, earthquakes not only cause irregular movements in the country's architecture, but they also cause short circuits, depending on the magnitude of the movement or the quality of the infrastructure. These can lead to fires and blackouts, which trigger panic among most of the population. It is also worth mentioning that earthquakes can obstruct exits or entrances to homes due to the irregular and abnormal movements that structures cannot withstand.

Using Raspberry Pi and IoT technology, software will be developed to implement the aforementioned features, with a system capable of controlling the internal doors of the facility, as well as the emergency lights installed in the door frames, monitoring the automatic shutdown of the facility's power supply, and, finally, alerting about the earthquake that occurred in the area.

Panorama General

Introducción

En un mundo donde los desastres naturales representan una amenaza constante, la tecnología se erige como una herramienta clave para salvaguardar vidas humanas. La Internet de las cosas (IoT), con su capacidad de interconectar dispositivos y facilitar el intercambio de datos en tiempo real, ofrece soluciones innovadoras para enfrentar desafíos críticos. Este informe presenta QuakeSafe, un sistema de monitoreo, control y alerta diseñado específicamente para abordar las problemáticas asociadas a los terremotos en un país altamente sísmico como Chile.

El sistema QuakeSafe combina tecnologías avanzadas como IoT y plataformas de control integradas, ofreciendo respuestas automatizadas ante movimientos telúricos. Estas incluyen el corte de energía para evitar incendios, la apertura de puertas internas para facilitar evacuaciones y la activación de luces de emergencia, mitigando así el impacto de eventos sísmicos en la seguridad de las personas.

Para su desarrollo, se realizó un análisis exhaustivo de los incidentes sísmicos en Chile, identificando los principales desafíos enfrentados por las comunidades en estos contextos. A partir de esta investigación, se planteó una solución que no sólo responde de manera eficiente a los eventos sísmicos, sino que también garantiza una experiencia de usuario intuitiva y confiable a través de una aplicación que permite supervisar y gestionar el sistema en tiempo real.

La implementación del sistema QuakeSafe se llevó a cabo siguiendo un enfoque estructurado que asegura su operatividad y efectividad. En primer lugar, se elaboraron **casos de uso detallados**, que describen las funciones esenciales del sistema, permitiendo prever el comportamiento en escenarios de emergencia. Estos casos de uso incluyen tanto un diseño general del sistema como una descripción específica de componentes individuales, como el corte de energía, la apertura automática de puertas y la activación de luces de emergencia.

El desarrollo técnico se basó en una arquitectura modular que incorpora dispositivos como sensores de vibración, controladores de motores para puertas, relés para la desconexión eléctrica y luces LED de emergencia. La integración de estos elementos se realizó mediante una red IoT, la cual garantiza una comunicación rápida y eficiente entre los componentes. Esto es esencial para lograr respuestas automáticas en tiempo real durante un sismo.

Además, se definió un modelo de diseño que detalla las interacciones entre los sensores y el sistema de control central. Este diseño fue validado a través de simulaciones que aseguran que las acciones automáticas se activan correctamente ante la detección de movimientos sísmicos.

Propósito

El propósito de este proyecto es que debe ser de carácter de Monitoreo y Control IoT, de tal forma que sean aptos para el mundo de ciudades inteligentes, fábricas inteligentes, compra y ventas inteligentes.

Alcance

Este proyecto consiste en reducir los cortocircuitos, prevenir la obstrucción de salidas o entradas de viviendas y disminuir el pánico de la ciudadanía en desastres socio-naturales, en este caso, terremotos. En el monitoreo, el sistema se encargará de medir las vibraciones que ocurran en el sitio donde se implementen los sensores, de tal forma que dé señal al sistema, lo que provocará que las luces de emergencia disponibles se prendan de forma inmediata, mientras que las entradas del recinto, internas, se abran de forma automática. En el término de control, las luces se podrán controlar por medio del sistema, mientras que en alerta, el sistema dará una pequeña notificación, tipo alerta, del funcionamiento de los sensores de vibración activados.

Objetivos

General

Desarrollar un sistema que sea capaz de monitorear, controlar y alertar eventos sísmológicos, para que pueda ser implementado en casas resguardando el bienestar de la ciudadanía.

Específicos

- Identificar y desarrollar una problemática haciendo uso de sensores, raspberry, etc.
- Estudiar y analizar las posibles soluciones para la problemática.
- Planificar el diseño del proyecto, por medio de maquetas y modelado 3D.
- Estudiar y aplicar las distintas herramientas para el desarrollo del diseño.
- Desarrollar un software capaz de controlar y monitorear los dispositivos instalados.
- Construcción de maqueta y modelo 3D que demuestre la idea del proyecto.
- Estudio y aplicación de Modelos de diseño.

Suposiciones

Respecto a nosotros

1. El modelo 3D representa correctamente los escenarios en los que se utilizará el proyecto.
2. Todas las herramientas de trabajo requeridas estarán disponibles para ser usadas dentro del plazo del proyecto.
3. El grupo aprenderá uniformemente y en conjunto los conocimientos básicos para desarrollar e implementar el producto.
4. El producto final cumplirá con las especificaciones definidas en el plan del proyecto. Las cuáles son las siguientes:
 - Entrega del producto terminado con sus requerimientos completados.
 - Controlar las puertas internas del recinto para el cumplimiento de uno de los objetivos del proyecto, evitar que las puertas queden atascadas y termine en el aislamiento del individuo.
 - Controlar la fuente eléctrica para el cumplimiento de uno de los objetivos del proyecto, evitar un incendio en el recinto.
 - Monitorear los sismos ocurridos a la cercanía del dispositivo, con un rango que represente
 - Alertar por medio del software los movimientos irregulares que detecten los sensores.
5. Las actividades a realizar se llevarán a cabo en los plazos establecidos previamente, de acuerdo a la carta gantt.

Respecto al usuario

1. Se asume que los usuarios tendrán acceso a dispositivos móviles inteligentes (smartphone).
2. Se asume que los usuarios utilizarán la página web para administrar y monitorear el sistema.
3. Se asume que los motores serán colocados solo en las puertas internas de la vivienda.
4. Se asume que las luces de emergencia estarán colocadas arriba de las puertas.

Restricciones

1. **Maqueta:** Para llevar a cabo la representación física, el presupuesto asignado para la maqueta física se limita a \$15.000 pesos, priorizando el uso de materiales reciclados.
2. **Sensores:** Limitados a utilizar sensores que pertenezcan al kit de Raspberry Pi.
3. **Software:** El desarrollo del software debe incluir funcionalidades de monitoreo y control de los dispositivos instalados, que permita la interacción con el sistema implementado.
4. **Tiempo:** Cada etapa del proyecto tiene un plazo establecido que debe cumplirse según la planificación original.

Entregables

- **Informes:** En estos informes se detalla el progreso que se llevó a cabo en cada etapa en cuanto a los objetivos, resultados, desafíos experimentados. Aportando una visión clara en cuanto al proyecto.
- **Bitácoras:** Por cada reunión de trabajo se sube a redmine una bitácora informando lo que se avanzó durante la misma
- **Carta Gantt:** Contienen información clara de las actividades a realizar a lo largo del proyecto.
- **Maqueta:** Representación física del entorno donde se implementara el sistema.
- **Modelado 3D:** Representación digital del entorno físico en el cual se basará la maqueta para la identificación de toda la instalación.
- **Wiki:** Presentación del proyecto y todos sus ámbitos en el desarrollo del proyecto.
- **Manual de Usuario:** Documento que contiene información de la utilización del producto.
- **Póster:** Cartel con el objetivo de entregar las partes más importantes del proyecto, de forma resumida y atractiva visualmente.

Organización del proyecto

Personal y entidades internas

A cada integrante del equipo de trabajo del proyecto se le asignó una responsabilidad, de la cual estará a cargo para cumplir en el tiempo estipulado. Pero se debe resaltar que aunque existan los responsables de cada eje, cada uno de los integrantes debe auxiliar a cada parte del proyecto, para así, velar una composición digna de sus propias tareas como las de los demás colegas de trabajo.

- **Jefe de proyecto:** Responsable de la planificación, ejecución y control del proyecto, asegurando que se cumplan los plazos, presupuesto y objetivos. Coordina el equipo, gestiona riesgos y comunica con las partes interesadas.
- **Documentador:** Genera y gestiona toda la documentación necesaria para el proyecto, incluidos informes, manuales y guías técnicas. Asegura que la información esté actualizada y disponible para el equipo.
- **Analista programador:** Analiza los requisitos del proyecto, diseña soluciones técnicas y desarrolla el código necesario. Se encarga tanto del análisis funcional como de la programación y pruebas del software.
- **Programador:** Encargado de escribir y mantener el código fuente del proyecto, implementar nuevas funcionalidades, realizar pruebas y depurar errores.
- **Diseñador:** Responsable de la apariencia visual y la experiencia de usuario (UI/UX), creando interfaces atractivas y funcionales que aseguren una interacción eficiente y agradable para el usuario.

Roles y responsabilidades

Tabla 2. Roles .

ROL	ENCARGADO	INVOLUCRADO
Jefe de Proyecto	Tihare Cabello	Tihare Cabello
Documentador	Tihare Cabello	Tihare Cabello, Cristian Huanca
Analista Programador	Cristian Huanca	Cristian Huanca
Programador	Byron Santibáñez	Byron Santibáñez, Liliana Galvez, Cristian Huanca, Tihare Cabello
Diseñador	Liliana Galvez	Liliana Galvez, Byron Santibáñez

Tabla 3. Responsabilidades.

ENTREGABLES	ENCARGADO/S
Bitácoras	Tihare Cabello
Carta Gantt	Cristian Huanca
Modelado 3D	Byron Santibáñez
Maqueta	Tihare Cabello, Cristian Huanca, Liliana Galvez
Informe	Tihare Cabello, Cristian Huanca, Liliana Galvez, Byron Santibáñez
Wiki	Tihare Cabello, Cristian Huanca, Liliana Galvez, Byron Santibáñez
Modelo de diseño (Diagrama de clases)	Tihare Cabello, Cristian Huanca, Liliana Galvez, Byron Santibáñez
Modelo de Interacción (Diagrama de secuencia)	Tihare Cabello, Cristian Huanca, Liliana Galvez, Byron Santibáñez
Arquitectura de Cliente-Servidor	Tihare Cabello, Cristian Huanca, Liliana Galvez, Byron Santibáñez
Póster de Proyecto	Tihare Cabello, Liliana Galvez

Mecanismos de comunicación

- **WhatsApp:** Software que facilita las comunicaciones, por medio de mensajes y llamadas, audiovisuales.
- **Discord:** Plataforma que facilita la comunicación audiovisual.
- **Redmine:** Plataforma donde se suben los archivos que forman parte del proyecto.
- **Correo electrónico:** Plataforma para la comunicación formal mediante correos electrónicos institucionales.
 - Tihare.cabello.diaz@alumnos.uta.cl
 - Liliana.galvez.yanqui@alumnos.uta.cl
 - Cristian.huanca.calle@alumnos.uta.cl
 - Byron.santibañez.fernandez@alumnos.uta.cl

Planificación de los procesos de gestión

Planificación de estimaciones

Tabla 4. Recursos Hardware.

RECURSO HARDWARE			
PRODUCTO	CANTIDAD	COSTO UNIDAD	COSTO TOTAL
Raspberry pi	1	\$ 124.600	\$ 124.600
Arduino	1	\$ 117.990	\$ 117.990
Kit de cables de conexión	1	\$ 1790	\$ 1790
Luz LED	4	\$ 950	\$ 3.800
Motor de rotación	2	\$ 2.050	\$ 4.100
Sensor de vibración	2	\$ 2.000	\$ 4.000
Rele 220v	4	\$ 5.990	\$ 23.960
Kit resistores	1	\$ 1.990	\$ 1.990
Notebook	4	\$ 200.000	\$ 800.000
Total			\$ 1.082.230

Tabla 5. Recurso Software.

RECURSO SOFTWARE		
PRODUCTO	MESES	COSTO
Microsoft Office 365	4	\$ 46.000
Python	4	\$ 0
Canva	4	\$ 96.000
Blender	4	\$ 0
Unity	4	\$ 0
Total:		\$ 142.000

Planificación de recursos humanos

Consideraciones para el proyecto:

- La fase de inicialización del proyecto comenzará en la segunda semana del ciclo escolar, a partir del martes 13 de agosto de 2024.
- La duración total del proyecto es de 4 meses, considerando que se dedicarán 6 horas de trabajo en clase por semana. Esto da un total de 96 horas de trabajo en clase a lo largo del proyecto.
- Se ha establecido un horario fijo para reuniones semanales, las cuales tendrán una duración de 1 hora y se realizarán los días miércoles y viernes. Esto resulta en 32 horas de trabajo autónomo.

Tabla 6. Recursos Humanos.

INTEGRANTES	ROL	HORA TOTAL	SUELDO/HORA	SUELDO TOTAL
Tihare Cabello	Jefe de Proyecto Documentador	128	\$ 6.200 \$ 4.000	\$ 1.305.600
Liliana Galvez	Diseñador	128	\$ 4.500	\$ 576.000
Cristian Huanca	Analista Programador	128	\$ 5.500	\$ 704.000
Byron Santibáñez	Programador	128	\$ 5.200	\$ 665.600
Total :				\$ 3.251.200

Planificación de costo total

Tabla 7. Costo Total.

TIPO COSTO	COSTO
Costo Hardware	\$ 1.082.230
Costo Software	\$ 142.000
Costo Recurso Humanos	\$ 3.251.200
Total:	\$ 4.475.430

Por lo tanto, la estimación del costo del proyecto, con una duración de 16 semanas, se calcula teniendo en cuenta el tiempo de trabajo en clase, las horas de reuniones y cualquier recurso adicional necesario, resultando en un costo estimado de \$4.357.440 pesos.

Lista de actividades (Carta Gantt)

Actividades de trabajo

Tabla 8. Actividades.

ACTIVIDAD	DESCRIPCIÓN	RESPONSABLE
Bitácoras	Registro del avance de las actividades semanalmente.	Tihare Cabello
Organización	Asignación de actividades que realizará cada integrante.	Tihare Cabello
Establecer Problemática	Identificar y definir la problemática a resolver.	Liliana Galvez
Analizar diversas soluciones	Analizar distintas alternativas de soluciones para resolver la problemática.	Tihare Cabello Liliana Galvez Cristian Huanca Byron Santibáñez
Creación modelo 3D	Realizar un diseño del escenario aplicando la solución propuesta al problema abordado.	Byron Santibáñez
Construcción de maqueta	Armado físico basado en el modelado 3D con materiales reciclados.	Tihare Cabello Liliana Galvez Cristian Huanca
Presentación	Se presenta la solución al problema, desarrollo de la maqueta y los materiales usados en la construcción de esta.	Tihare Cabello Liliana Galvez Cristian Huanca Byron Santibáñez
Informe Final	Crear informe final: Implementación	Tihare Cabello Liliana Galvez Cristian Huanca Byron Santibáñez

Asignación de tiempo



Figura 1. Carta Gantt.

Planificación de la gestión de riesgos

Para gestionar los riesgos, se identificaron y categorizaron los siguientes tipos de riesgo y sus factores:

Tabla 9. Estimación.

TIPO DE RIESGO	DESCRIPCIÓN
Tecnológico	Retrasos en la entrega o problemas relacionados con el hardware o el software. Se reportan numerosos problemas tecnológicos.
Humano	Baja moral del personal, problemas de salud, malas relaciones entre los miembros del equipo y dificultades para encontrar personal disponible.
Herramientas	Resistencia del equipo a utilizar ciertas herramientas, quejas sobre las herramientas de trabajo, y solicitudes de estaciones de trabajo más potentes.
Requerimientos	Frecuentes cambios en los requerimientos, quejas del cliente.
Organizacional	Rumores dentro de la organización y falta de liderazgo por parte de la dirección principal.
Estimación	Incumplimiento de los plazos acordados y dificultades para eliminar defectos reportados.

Para los riesgos latentes que podrían surgir durante el desarrollo del proyecto, se clasificaron en los siguientes cuatro niveles de impacto:

- 1. Catastrófico:** Impacto crítico que puede poner en riesgo la continuidad o el éxito del proyecto.
- 2. Crítico:** Impacto significativo que requiere recursos adicionales para ser gestionado, pero el proyecto puede continuar.
- 3. Marginal:** Impacto leve que puede retrasar algunos aspectos del proyecto, pero sin afectar gravemente los resultados.
- 4. Despreciable:** Impacto mínimo que no requiere acción inmediata y no afectará de manera relevante el desarrollo del proyecto.

Tabla 10. Ocurrencia.

PROBABILIDAD DE OCURRENCIA	RANGO DE %
Alta	71% - 100%
Media	31% - 70%
Baja	0% - 30%

Tabla 11. Riesgos.

RIESGO	TIPO	PROBABILIDAD DE OCURRENCIA	NIVEL DE IMPACTO	ACCIÓN REMEDIAL
Daños o pérdida de la tarjeta SD.	Tecnológico	Alta	2	Realizar copias de seguridad periódicas y tener SD adicionales disponibles para reemplazo inmediato.
Fallo en los sensores de vibración.	Tecnológico	Media	2	Realizar pruebas exhaustivas y tener sensores de repuesto, además de un plan de mantenimiento.
Cambio de los requerimientos.	Requerimientos	Alta	2	Realizar una reunión con el cliente, en la que se discutirá la viabilidad de los requerimientos y su importancia.
Falta de coordinación entre miembros del equipo.	Organizacional	Media	3	Establecer reuniones de seguimiento frecuentes y usar herramientas de comunicación efectiva (WhatsApp, Discord, Redmine).
Falta de asistencia de reunión.	Humano	Media	3	Amonestar al responsable, y solicitar una penalización adecuada.
Errores en el software de control del sistema.	Requerimientos	Alta	2	Implementar pruebas continuas desde el inicio del desarrollo para detectar y corregir fallos.
Sobrecalentamiento del hardware.	Tecnológico	Baja	2	Instalar correctamente todos los componentes y monitorear la temperatura de los dispositivos.
Enfermedades del personal.	Humano	Baja	4	Redistribuir sus tareas entre los integrantes según sus habilidades.
Incompatibilidad entre los	Herramientas	Media	3	Verificar la compatibilidad de los componentes antes de comprarlos y

Proyecto II Sistema de monitoreo, control y alerta sísmológico

componentes del hardware.				realizar pruebas.
Incapacidad de completar la maqueta física dentro del presupuesto.	Estimación	Media	3	Utilizar la mayor cantidad de materiales reciclados posibles y monitorear constantemente los gastos para no exceder el presupuesto.
Personal sin experiencia.	Organizacional	Media	3	Implementar programas de formación para mejorar las habilidades del personal.
Error de cálculo de costos.	Estimación	Media	2	Realizar una reevaluación de los costos del proyecto para garantizar la fiabilidad de los cálculos. Si ocurre más de una vez, hacer cambio de responsable y realizar una amonestación respectiva.
Componentes defectuosos.	Tecnológico	Media	2	Comunicarse con el encargado de asignación de componentes para su debido cambio y notificar el problema del componente respectivo. Tener componentes de repuesto en caso de salir defectuosos
Salida de un integrante del proyecto.	Humano	Baja	2	Reorganización del plan del proyecto para entregar nuevas responsabilidades y roles correspondientes.
Tiempo de desarrollo subestimado.	Estimación	Media	3	Comunicarse con el responsable de la tarea, identificar los obstáculos y ajustar el cronograma si es necesario, para asegurar el desarrollo íntegro de la actividad. Realizar una adaptación del personal.

Planificación de Procesos Técnicos

Modelo de Diseño (Caso de uso general)

El modelo del proceso cuenta con los siguientes actores, para su correcto funcionamiento en el sistema:

- Sensor de vibración.
- Relé.
- Interruptor.
- Motor Rotatorio.
- Usuario.

Se ha modelado el siguiente diagrama, el funcionamiento e interacción de los autores del sistema, de la siguiente manera:

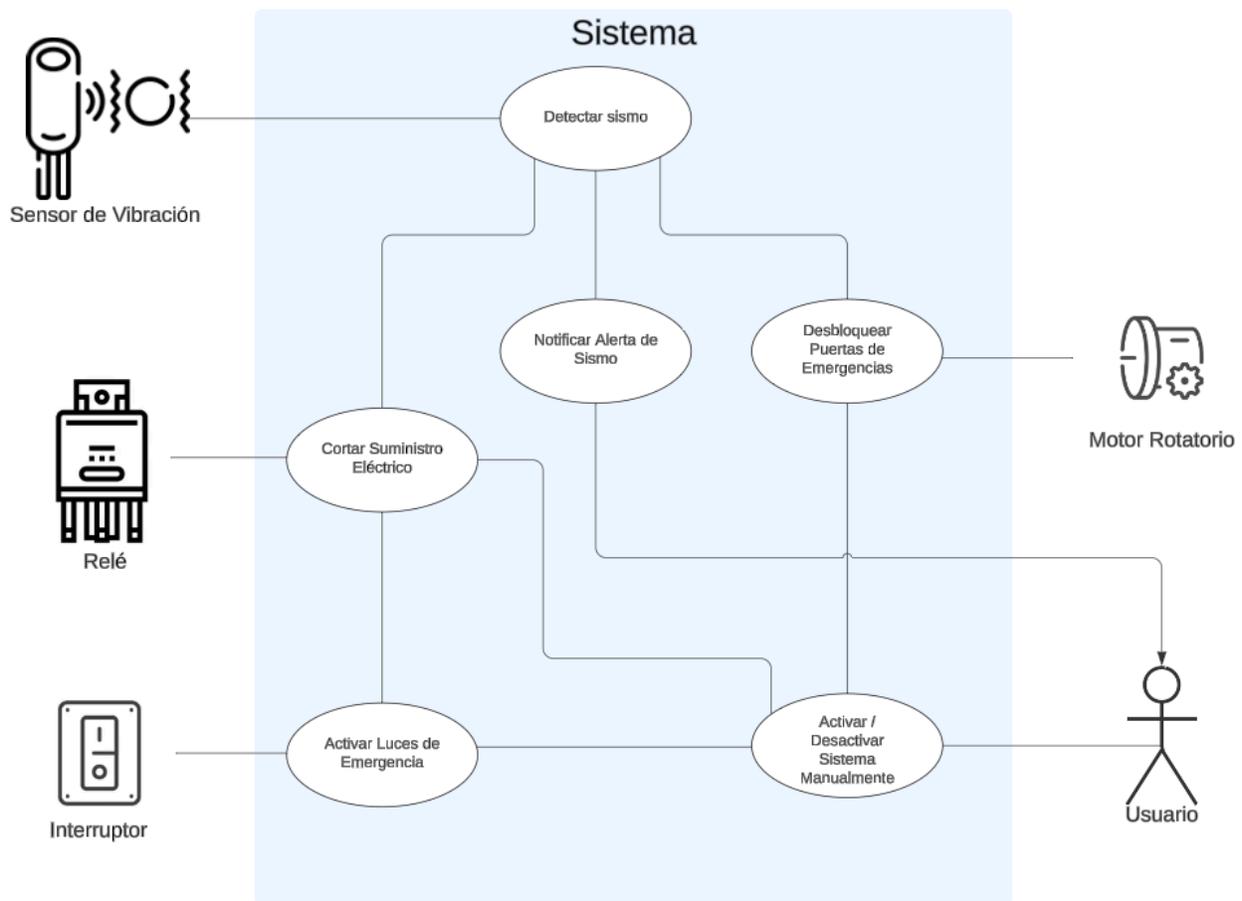


Figura 2. Caso de uso General.

Diagrama de Clases de Caso de Uso General

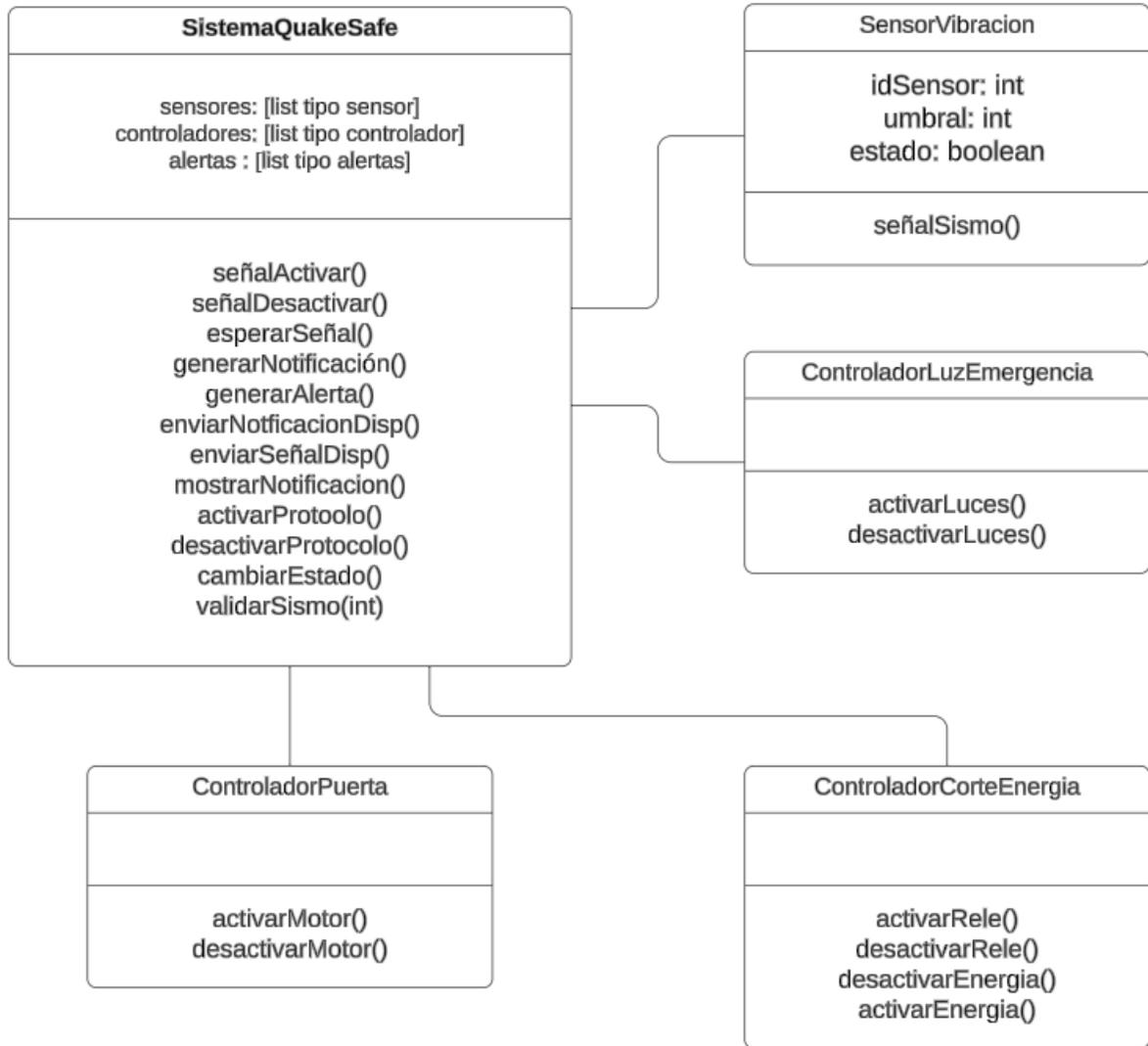


Figura 3. Diagrama de Clases

Casos de uso

C.U.S: Detectar Sismo

Tabla 12. C.U.S Detectar Sismo.

Nombre CUS: Detectar Sismo
Resumen: El sistema recibe información del sensor de vibración cada vez que se provoca un sismo para determinar si éste sobrepasa el umbral especificado y realizar las acciones correspondientes.
Actor: Sistema, Sensor de Vibración.
Precondición: El sistema se encuentra en un estado de espera, monitoreando continuamente las señales del sensor de vibración y verificando el umbral predefinido por cualquier movimiento sísmico.
Descripción: <ol style="list-style-type: none">1. El sensor de vibración detecta un movimiento que podría indicar un sismo y lo compara con un umbral predefinido.2. Si el movimiento supera el umbral, el sensor envía una señal al sistema.3. El sistema recibe y valida la señal, confirmando la existencia de un sismo.4. El sistema verifica que la magnitud del sismo cumple o excede el nivel establecido en el umbral de activación.5. Si se confirma la magnitud, el sistema envía una señal de activación a los dispositivos de respuesta, iniciando los protocolos de alerta.
Alternativa: <ol style="list-style-type: none">1.1 Si el movimiento detectado no supera el umbral predefinido, el sistema continúa en estado de espera, sin activar los protocolos de alerta.
Alternativa: <ol style="list-style-type: none">5.1 Si no se confirma la magnitud, el sistema no envía una señal de activación a los demás dispositivos.
Postcondición: El sistema está listo para iniciar la secuencia de respuesta y desplegar las alertas correspondientes.

Diagrama de secuencia: "Detectar Sismo"

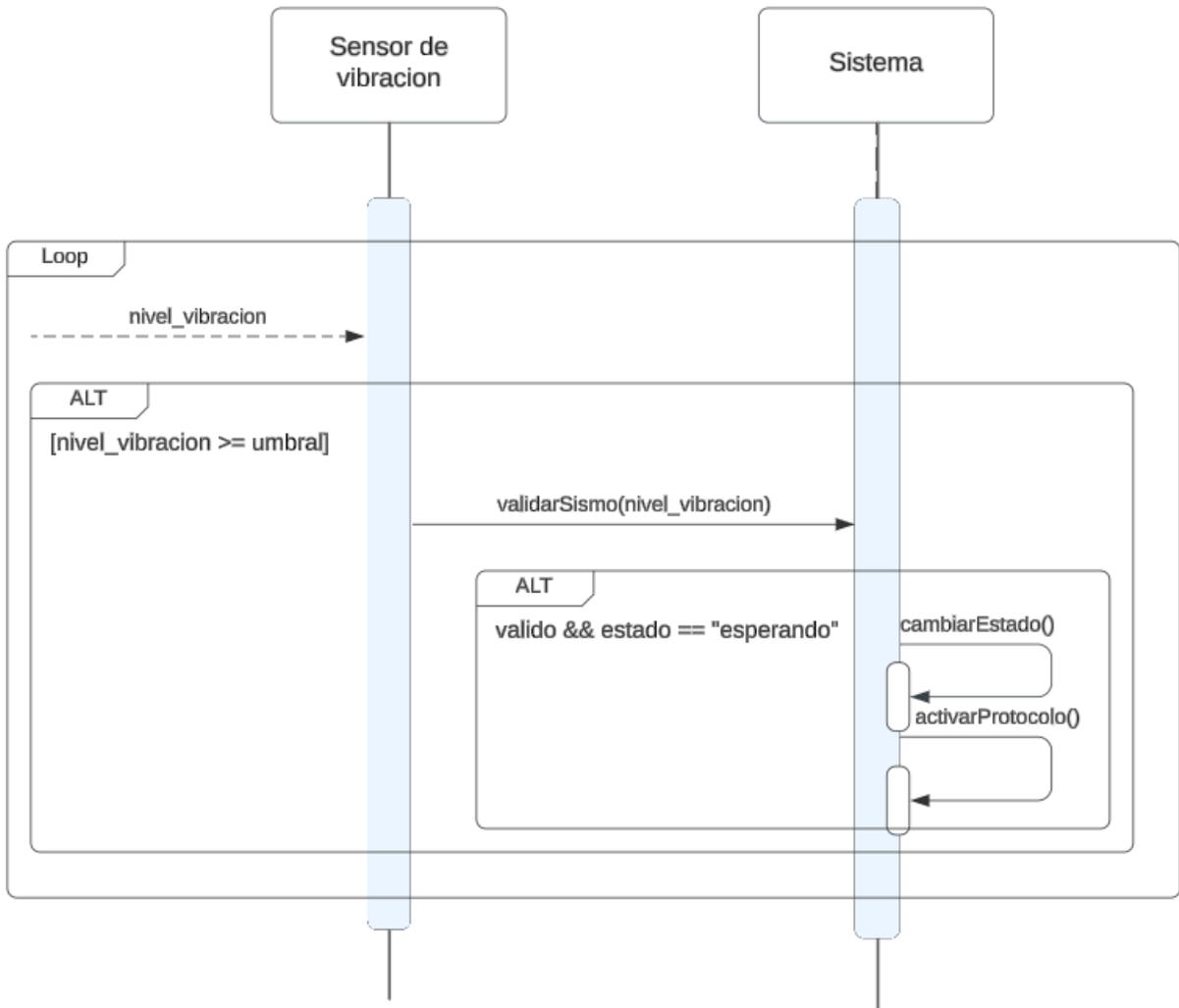


Figura 4. Diagrama de secuencia C.U.S Detectar Sismo

C.U.S: Notificar Alerta de Sismo

Tabla 13. C.U.S Notificar Alerta de Sismo.

Nombre CUS: Notificar Alerta de Sismo
Actor: Sistema
Resumen: Una vez detectado un sismo, el sistema genera y envía una alerta indicando el evento sísmico
Precondición: El sistema detecta un sismo.
Descripción: <ol style="list-style-type: none"> 1. Genera una notificación visual y sonora de alerta sísmica. 2. Muestra notificación en interfaz. 3. La notificación se envía a los dispositivos conectados. 4. Se envía una señal a los dispositivos conectados para su funcionamiento.
Postcondición: El sistema está listo para iniciar la secuencia de respuesta.

Diagrama de secuencia: “Notificar Alerta de Sismo”

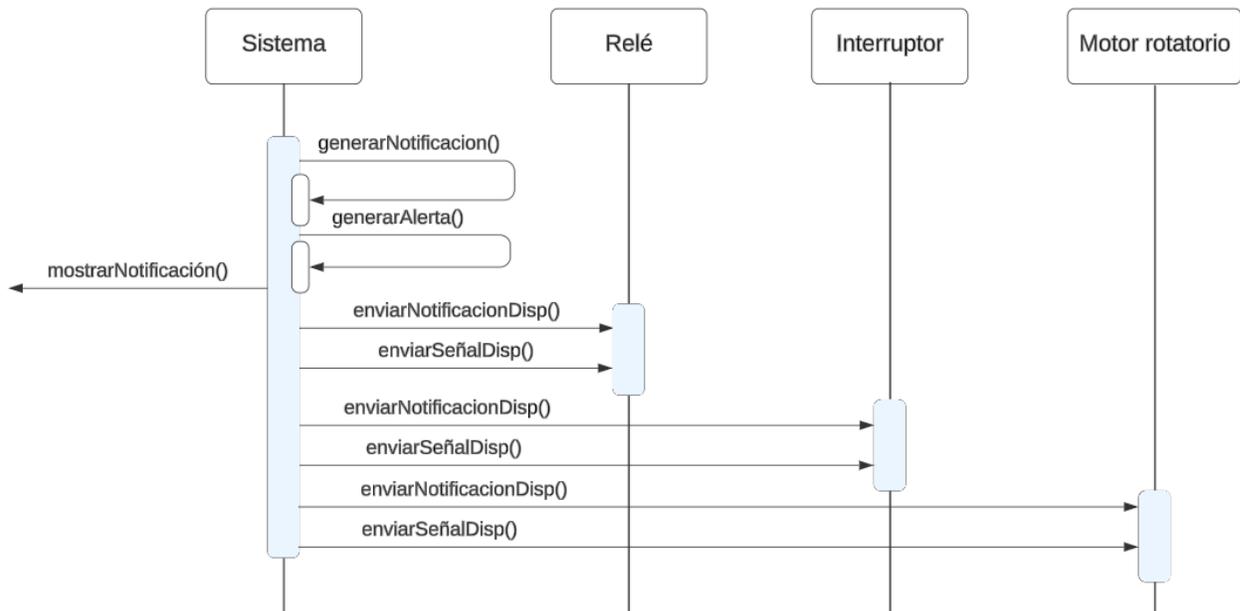


Figura 5. Diagrama de secuencia C.U.S Notificar Alerta de Sismo

C.U.S: Cortar Suministro Eléctrico

Tabla 14. C.U.S Cortar Suministro Eléctrico.

Caso de uso: Cortar Suministro Eléctrico
Actor: Sistema, Relé.
Resumen: El sistema corta el suministro eléctrico automáticamente al momento de detectar un sismo que exceda el umbral predefinido, minimizando riesgos eléctricos.
Precondición: Se debe haber detectado un sismo mediante el caso de uso “Detectar Sismo” y se ha enviado una señal de activación a los dispositivos de respuesta.
Descripción: <ol style="list-style-type: none">1. Se recibe una señal de activación.2. Envía una señal de corte eléctrico al relé.3. El Relé recibe una señal del sistema y verifica la disponibilidad de conexiones.4. El Relé corta el suministro eléctrico.5. Se envía una señal de confirmación, indicando que el suministro ha sido interrumpido.
Alternativa: <ol style="list-style-type: none">2.1. No se cumplen las condiciones de corte eléctrico, el sistema mantiene el suministro activo.
Alternativa: <ol style="list-style-type: none">4.1. Se estabiliza después del evento, el usuario puede restaurar el suministro eléctrico manualmente.
Postcondición: El suministro eléctrico se mantiene interrumpido hasta que se reciba una instrucción para su reactivación o hasta que las condiciones de seguridad permitan el restablecimiento.

Diagrama de secuencia: “Cortar Suministro Eléctrico”

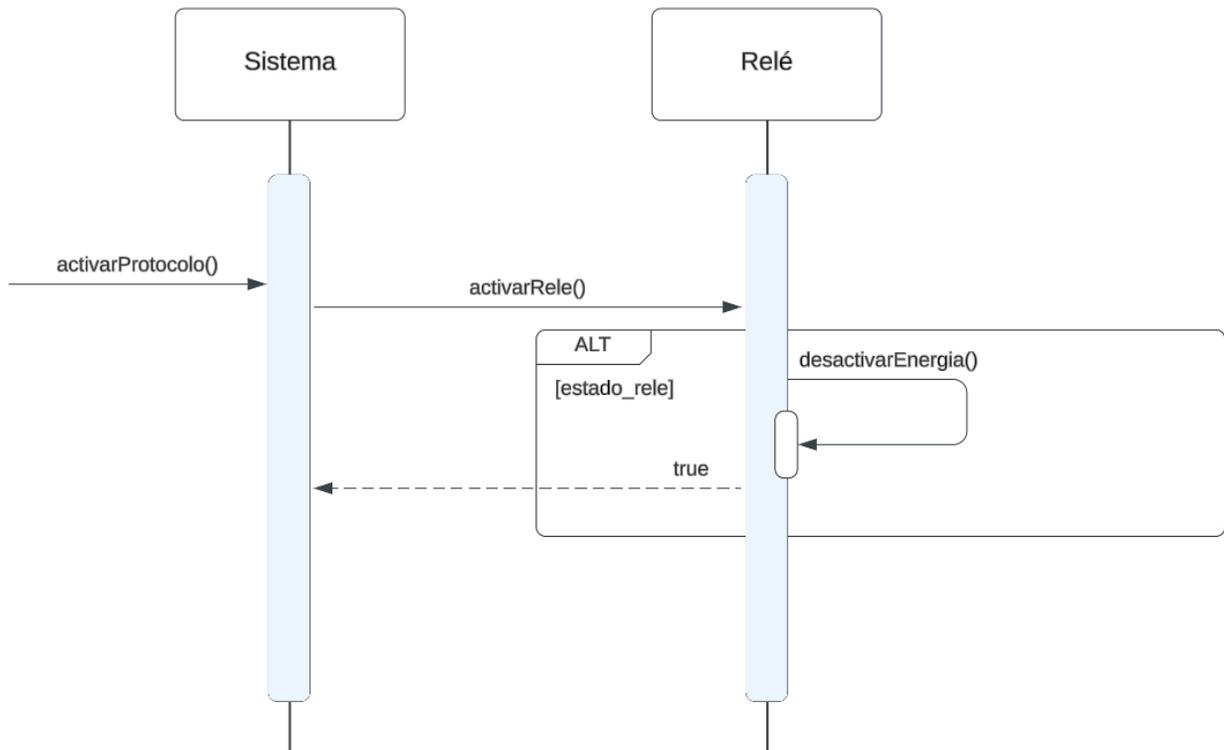


Figura 6. Diagrama de secuencia C.U.S Cortar Suministro Eléctrico

C.U.S: Activar Luces de Emergencia

Tabla 15. C.U.S Activar Luces de Emergencia.

Caso de uso: Activar Luces de Emergencia
Actor: Sistema, Interruptor.
Resumen: El sistema activa las luces de emergencia para garantizar la visibilidad de las zonas críticas tras la detección de un sismo.
Precondición: Se ha detectado un sismo que sobrepasa el umbral predefinido, mediante el caso de uso "Detectar Sismo" y se ha enviado una señal de activación a los dispositivos de respuesta.
Descripción: <ol style="list-style-type: none">1. Envía una señal al interruptor.2. Se activan las luces de emergencia en las áreas críticas..3. Se envía una señal para mantención del sistema activo.
Alternativa: <ol style="list-style-type: none">2.1. El usuario apaga manualmente las luces de emergencia, mediante el sistema.2.2. Envía una señal al interruptor.2.3. El interruptor desactiva las luces de emergencia.
Alternativa: <ol style="list-style-type: none">3.1. Si el sistema considera que no es necesario mantener las luces encendidas, se emite una señal para apagar las luces de emergencias, luego de un tiempo determinado.
Postcondición: Las luces de emergencia cambian a estado activo y se mantienen hasta que sean apagadas manualmente o programadas.

Diagrama de Secuencia: "Activar Luces de Emergencia"

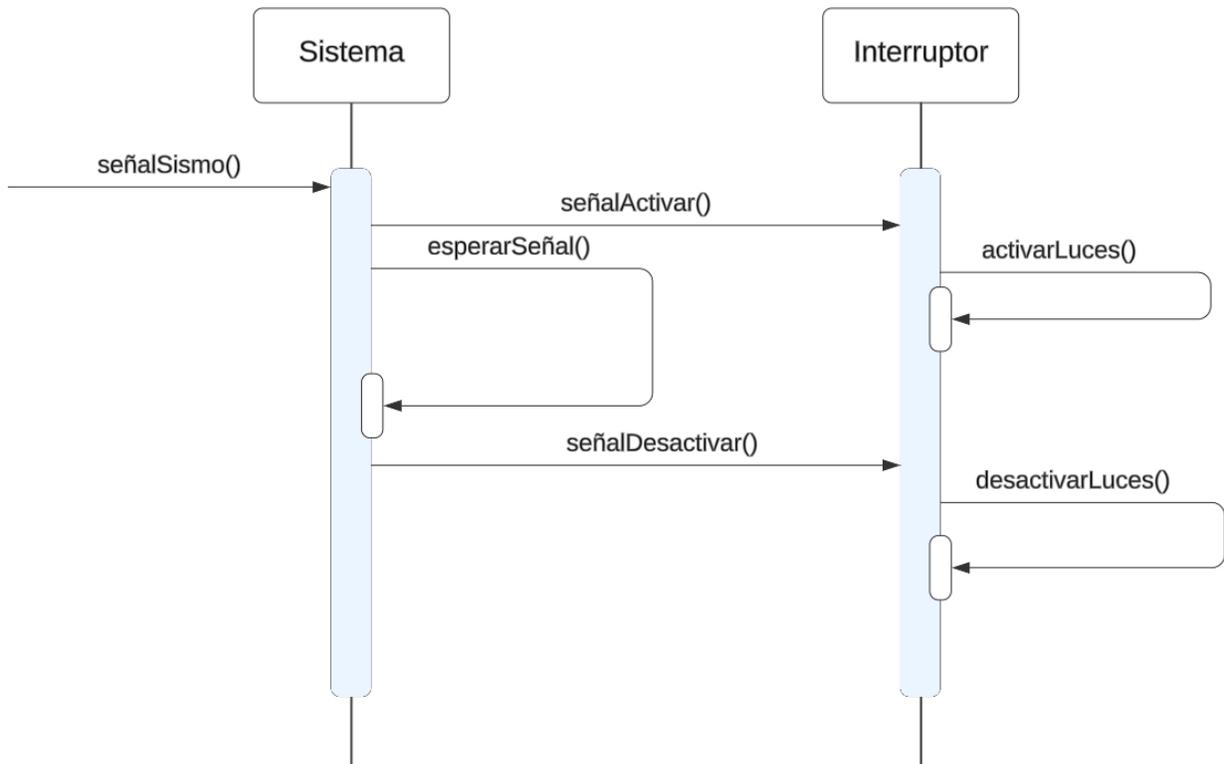


Figura 7. Diagrama de secuencia C.U.S Activar luces de emergencia.

C.U.S: Desbloquear Puertas de Emergencia

Tabla 16. C.U.S Desbloquear Puertas de Emergencia.

Caso de uso: Desbloquear Puertas de Emergencia.
Actor: Sistema, Motor de Rotación.
Resumen: El sistema acciona el motor de rotación para desbloquear y abrir las puertas de emergencia, facilitando la evacuación en caso de sismo.
Precondición: Se ha detectado y confirmado un sismo que cumpla con el umbral predefinido para activar los protocolos de emergencia.
Descripción: <ol style="list-style-type: none">1. Envía una señal de activación al Motor de Rotación de las puertas de emergencia para su activación.2. El motor de rotación se activa y opera.3. El motor de rotación desbloquea y abre las puertas de emergencia.4. Verifica que las puertas hayan sido completamente desbloqueadas y abiertas.
Alternativa: <ol style="list-style-type: none">2.1 Si el motor de rotación no responde, el sistema reintenta el envío de la señal hasta un máximo de tres intentos.2.2 Si el motor no se ha activado después de los intentos, el sistema notifica una alerta para intervención manual.
Postcondición: Las puertas de emergencia quedan abiertas y desbloqueadas, permitiendo la evacuación.

Diagrama de secuencia: “Desbloquear Puertas de Emergencia”

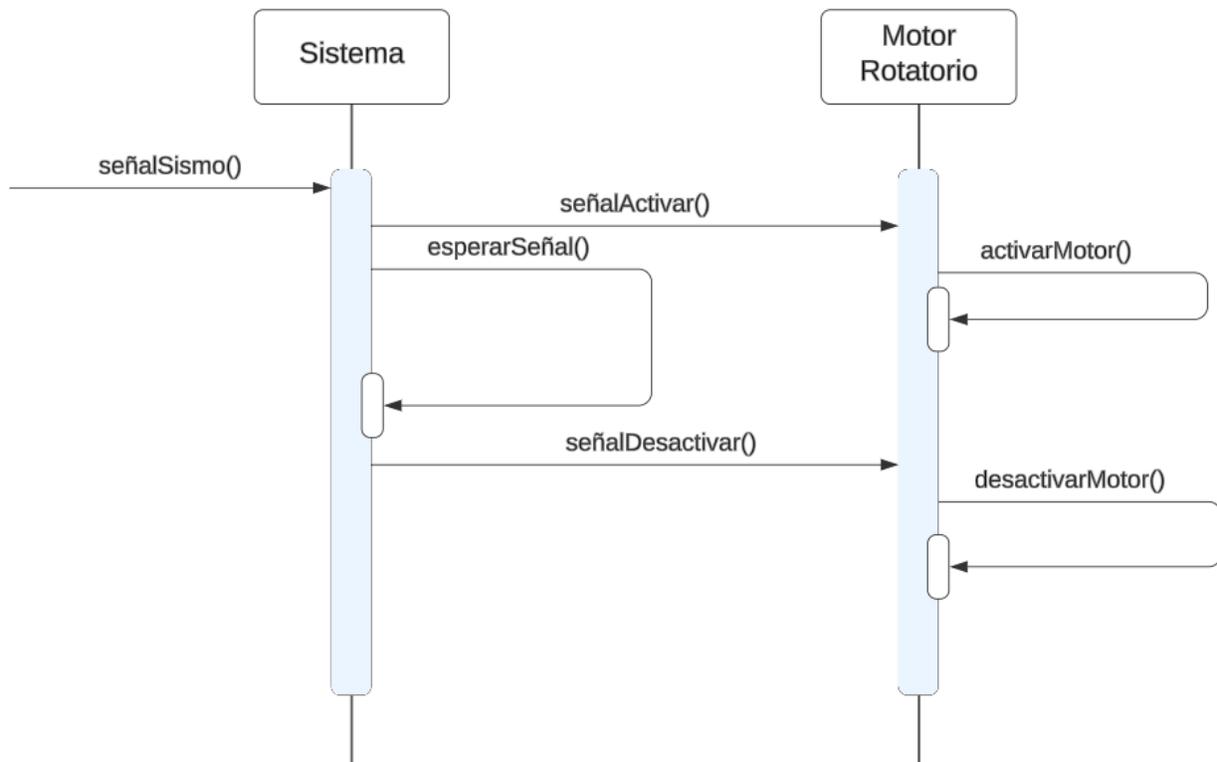


Figura 8. Diagrama de secuencia C.U.S Desbloquear puertas de Emergencia

C.U.S: Activar / Desactivar Sistema Manualmente

Tabla 17. C.U.S Activar / Desactivar Sistema Manualmente.

Caso de uso: Activar / Desactivar Sistema Manualmente.
Actor: Sistema
Resumen: El usuario tiene la capacidad de activar o desactivar el sistema de manera manual en situaciones de prueba, mantenimiento o emergencia.
Precondición: El sistema debe estar en estado de espera o en modo activo tras detectar un sismo.
Descripción: <ol style="list-style-type: none">1. Muestra interfaz con opciones de funcionalidades del sistema.2. El usuario envía una señal manual al sistema para activar o desactivar los protocolos de emergencia.3. Recibe la señal y activa las luces de emergencia, el sistema de corte de suministro eléctrico, o cualquier otro dispositivo de respuesta.4. Verifica el estado de cada dispositivo activado (como las luces de emergencia) para confirmar su correcto funcionamiento.
Alternativa: <ol style="list-style-type: none">2.1 El usuario envía una señal manual para apagar las luces de emergencia u otros dispositivos activados.2.2 Se encuentra en modo de prueba, los dispositivos se activan únicamente para verificar su estado sin realizar acciones permanentes (como cortar el suministro eléctrico de forma permanente).
Postcondición: El sistema queda en el estado definido por el usuario (activado o desactivado), y se genera un registro del cambio manual en el sistema para fines de auditoría.

Diagrama de Secuencia: “Activar / Desactivar Sistema Manualmente”

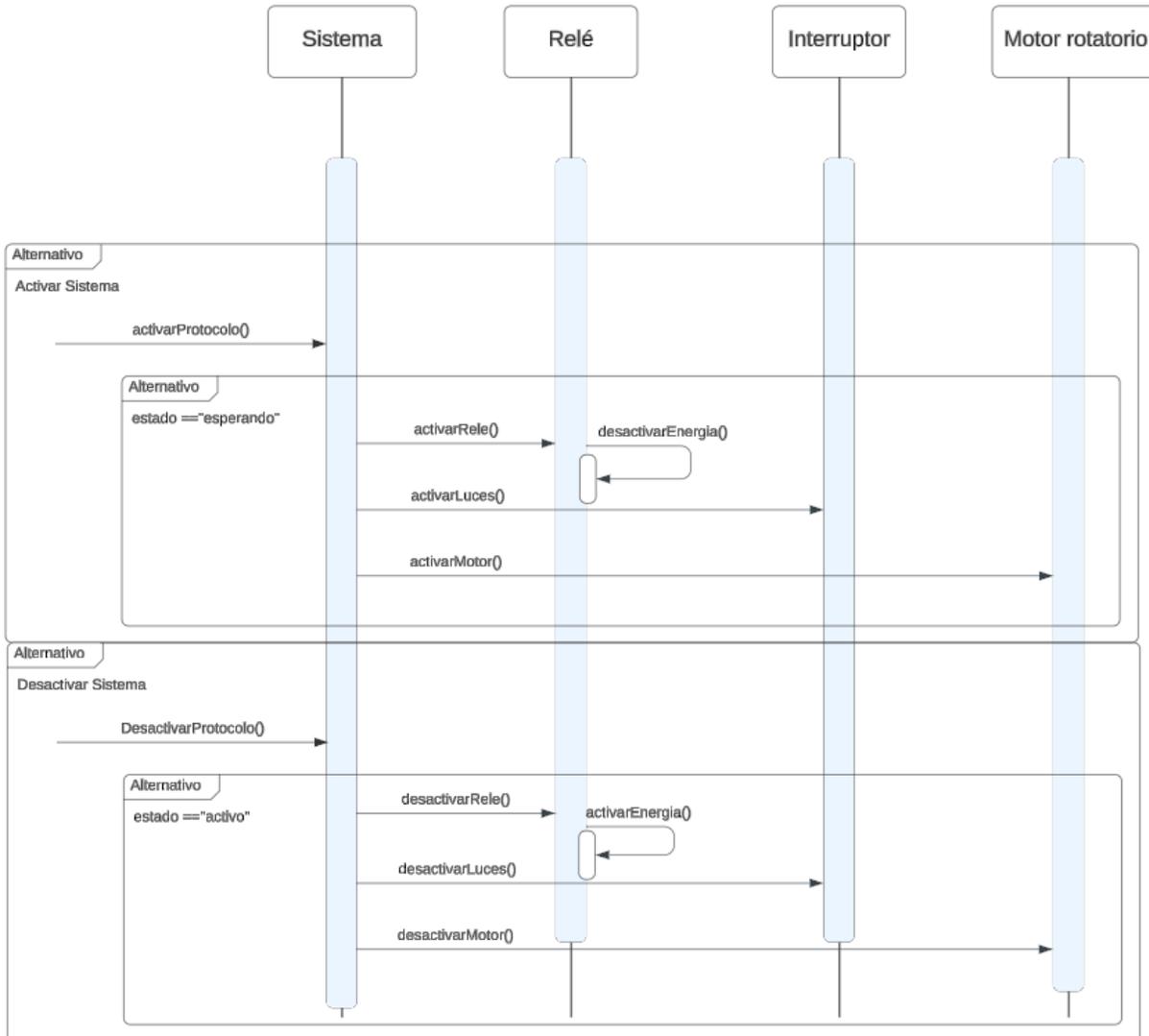


Figura 9. Diagrama de secuencia C.U.S Activar / Desactivar Sistema Manualmente

Descripción de la Arquitectura

Es necesario tener en cuenta el proceso de conexión entre el usuario final y el modelo en el cual se actuará. En este caso, se ha optado por la implementación de la Arquitectura de Cliente-Servidor, donde se encuentran los siguientes conceptos:

- **Cliente:** Aquella persona que utilizará un dispositivo móvil, en el que interactuara con el sistema, donde, además de recibir notificaciones y alertas, manejará ciertas funcionalidades de manera manual, como activar o desactivar una función.
- **Servidor:** Está compuesto por Raspberry Pi 4 y Arduino, donde se ejecutará un software que funciona como un intermediario entre el hardware y el usuario que interactúa con este. Se utilizarán las librerías de Python.

Para el buen funcionamiento de, tanto Cliente como Servidor, necesitan estar conectados a una red Wi-Fi, o si no, por una conexión alámbrica Ethernet.

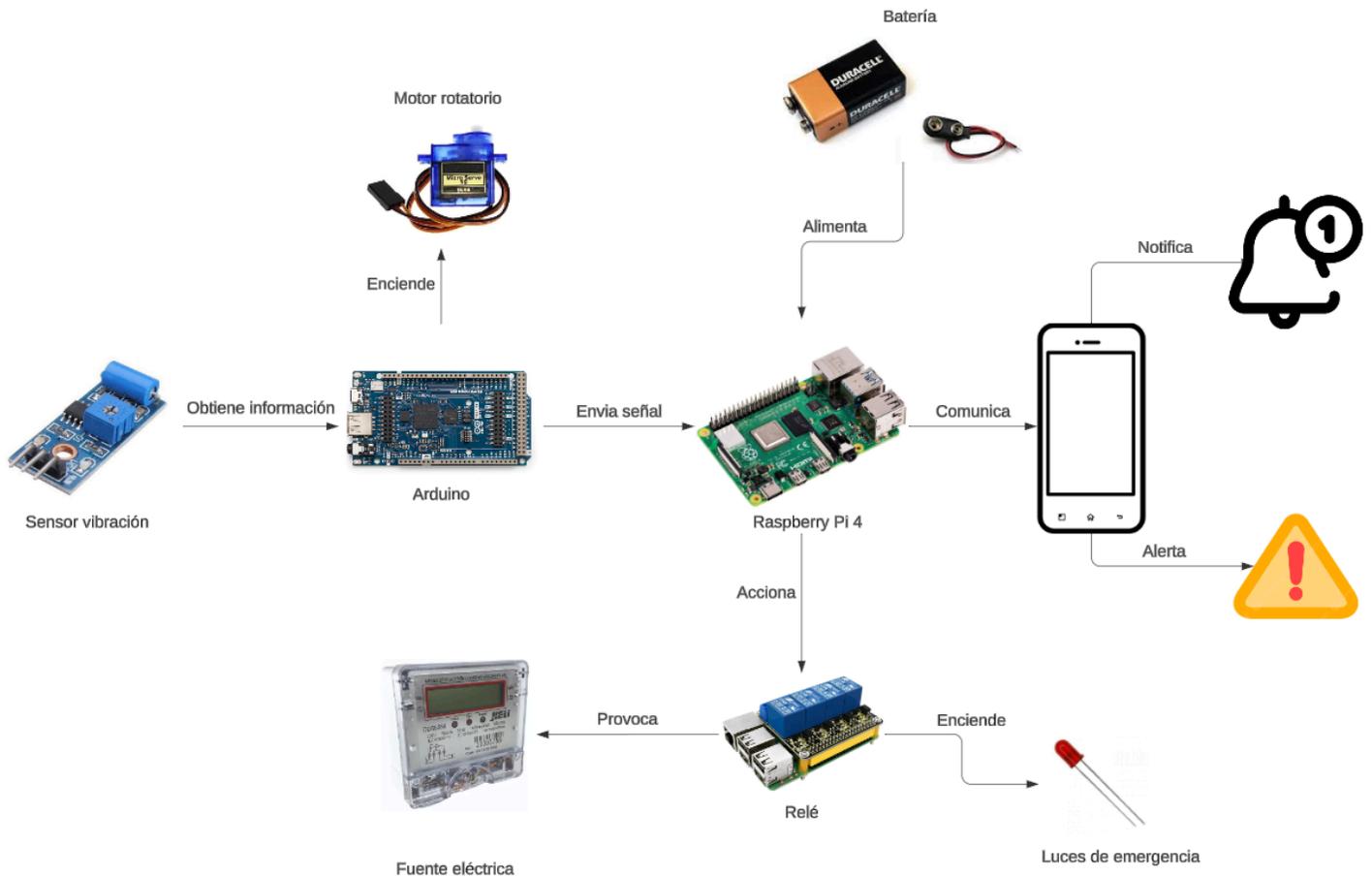


Figura 10. Arquitectura Cliente-Servidor

- **Sensor vibración:** Medirá la cantidad y la frecuencia de vibración que se encuentre en el sistema, las que se pueden usar para detectar desequilibrios u otros problemas cuando se encuentre activo.
- **Relé:** Funcionara como un interruptor, para abrir y cerrar el paso de la corriente eléctrica que se encuentre en el hogar, pero solo puede ser accionado eléctricamente, por conexiones.
- **Fuente eléctrica:** Aquel que se encargará de entregar la energía al hogar.
- **Luces de emergencia:** Son la iluminación para situaciones críticas, que se activará solo por señal de activación del sistema.
- **Servo Motor:** Aquel que se encargará de controlar las puertas internas, que harán posible el giro libre de las puertas, que serán accionadas solo por el sistema cuando sea necesario.
- **Batería:** Fuente de energía eléctrica almacenada con la cual funcionará el sistema en el momento que se accione el corte de la fuente eléctrica.
- **Arduino R1:** El Arduino es el cerebro del sistema. Su rol es procesar las señales de los sensores (como el sensor de vibración) y tomar decisiones basadas en ellas. También controla otros dispositivos, y el servo motor. El Arduino recibirá entradas (como vibraciones, accionar el motor).
- **Raspberry Pi 4:** Al igual que en el sistema con Arduino, el sensor de vibración detecta las vibraciones mecánicas en su entorno y genera una señal cuando ocurre alguna vibración significativa y esta le llega al Raspberry para realizar las acciones correspondientes.

Herramientas y técnicas

Todo proyecto, para su realización, necesita una variedad de herramientas para su respectivo desarrollo, en la que se pueden destacar las siguientes herramientas utilizadas:

Herramientas Implementadas:

- **Visual Studio Code:** Editor de código fuente, diseñado para, principalmente, ser ligero, rápido y altamente personalizable, además de tener un sistema IntelliSense (un autocompletado inteligente), métodos y funciones detalladas con una documentación integrada, depuración integrada y tener un buen control de versiones, soporta múltiples lenguajes de programación y tiene una interfaz intuitiva y de fácil utilización.
Se utilizará este editor de código para desarrollar y gestionar el proyecto.
- **Python:** Lenguaje de programación interpretado, orientado a objetos, y de alto nivel con una semántica dinámica. Por lo que su alto nivel construido de datos, combinado con una teclando dinámica y unión dinámica, lo hacen muy atractivo para el desarrollo de aplicaciones rápidas, así como para conectar sus componentes existentes juntos. Es simple, fácil de aprender a enfatizar, tiene una sintaxis legible y por lo tanto, reduce el costo del mantenimiento del programa. Además de tener una extensa biblioteca estándar disponible sin cargo y para todas las plataformas principales.
Este lenguaje de programación se utilizará para desarrollar parte de la lógica del proyecto.
- **C:** Es un lenguaje de programación estructurado, pero no estructurado por bloques, por lo que no es posible declarar pequeños trozos de código dentro de otros pequeños trozos de código. Es un lenguaje no rígido en la comprobación de datos, lo que permite fácilmente la conversión entre diferentes tipos de datos y la asignación entre tipos de datos diferentes.
Este lenguaje de programación se utilizará para desarrollar parte de la lógica del proyecto.
- **Raspberry OS:** Este es un sistema operativo específicamente para el correcto uso del dispositivo a utilizar en el proyecto, el Raspberry Pi, que está optimizado para el rendimiento de este mismo.
Proporcionará la interfaz y plataforma por la cual se desarrollará y ejecutará el sistema para el proyecto.

Herramientas y dispositivos:

- Sensor de vibración.
- Motor rotatorio.
- Interruptor.
- Relé.
- Luces LED.

Diseño de Interfaz de Usuario

Para facilitar el control del sistema en el hogar, se ha diseñado una interfaz simple dividida en dos secciones: Sistema Control y Sistema Estado, además de un botón de salida. Cada una de estas secciones permite gestionar distintos aspectos del hogar de manera organizada y accesible. A continuación, se muestra la disposición de estas secciones en la interfaz.



Figura 11. Interfaz de Usuario Control

Sección Sistema Control: En esta sección se muestra el estado actual de todas las luces de emergencia instaladas en el hogar, junto con las puertas internas. Cada luz se identifica según el área donde se encuentra y el estado en el que se encuentra (Encendida o apagada), igualmente, con las puertas del hogar, sucede lo mismo, donde se encuentra el estado de estas (Abierta o cerrada). Por lo que el usuario puede controlar tanto las luces de emergencia como las puertas.



Figura 12. Interfaz de Usuario Estado

Sección Sistema Estado: En esta sección se muestra el estado actual de los sensores instalados en el hogar, como la ubicación en la que se encuentran, indicando si están activados o desactivados. Cada sensor cuenta con un indicador de operatividad, lo cual permite verificar su efectividad y en caso de fallo o incidente, facilitar su reparación o reemplazo.

Botón salir: Este botón se encarga de la funcionalidad de salir de la aplicación al accionarlo, de manera que el usuario pueda abandonar la aplicación.

Planificación de procesos de soporte

Planificación de documentación

- **Manual de usuario:** Este documento es creado para el usuario, en lo que se implementa la instalación, la utilización del software entregado y la solución de los errores que puedan ocurrir o cómo evitarlos.
- **Wiki del proyecto:** Este es un blog, dedicado para el desarrollo del proyecto, encargado de documentar el desarrollo del proyecto y el propósito que cumple.
- **Documentación del código:** Este documento se encarga de explicar las funcionalidades a detalle que han sido implementadas en el proyecto y software.

Problemas encontrados

A lo largo del proyecto, se encontraron diversas dificultades, desde el ámbito de funcionalidad como implementación de los diferentes dispositivos utilizados para el correcto desarrollo del proyecto, hasta el hardware utilizado para dicha implementación.

Problema con librería de GPIO

Este problema surgió debido a la poca estabilidad que portaba esta librería, debido a que algunas de las funcionalidades que se utilizaban no estaban soportadas por esta misma.

Solución

Para solucionar este problema, se utilizó la librería RPi.GPIO para el uso de las funciones más actualizadas que podía ofrecer la librería GPIO.

Problema con la conexión remota

Este problema surgió debido a la inestabilidad que poseía el Raspberry Pi, debido a su lentitud en la instalación y sistema en general.

Solución

Se utilizó un ambiente virtual para una mejor aplicación del sistema, y de esta forma se pudo utilizar de forma correcta las funcionalidades que posee el Raspberry.

Problemas con la conexión de relé

Se trabajaba con un relé no soldado, pero al soldar la placa no se encontraba con conexiones posibles por lo que no se pudo seguir trabajando con ello.

Solución

Se conectó directamente a la Raspberry, provocando otro problema debido a la falta de pines para el uso del sensor de vibración necesario para el cumplimiento del objetivo del proyecto, en lo que además, se incluyó una fuente de alimentación externa, en la que se utilizó un Arduino.

Problema con conexiones de pines

Al poder hacer la unión correcta del relé sobre el Raspberry Pi, se hizo uso de la totalidad de los pines disponibles del Raspberry, por lo que no se podía hacer conexiones directas a este.

Solución

Se implementó un dispositivo externo, conectado directamente a la Raspberry, un Arduino, el cual, además de servir como una fuente de alimentación para el relé, se utilizó como una extensión de pines, pero teniendo que recurrir a las funciones que proporcionaba Arduino, algo que no provocó problemas en la implementación.

Implementación

La implementación consiste en la integración de los módulos realizados para cumplir los requerimientos postulados por la problemática que ha sido establecida para la realización del proyecto, realizando una especificación técnica y de códigos, hechas en el lenguaje de programación python, que se han puesto en práctica en el proyecto.

Plan de integración

La aplicación móvil presentada tiene como objetivo, además de ser simple a su uso, prevenir dificultades al momento de que ocurran movimientos sísmicos de cierto grado, entornado en Chile, el cual, es un país que se encuentra en una ubicación geográfica, ser uno de los países más sísmicos. Por lo que se pretende lograr la aplicación es que al momento que el sensor esté implementado un sensor capaz de detectar las vibraciones y mande una señal al dispositivo para así, alertar sobre el movimiento, el cual tiene denominado un rango por el cual se activa y en el que no. De esta manera, se activarán automáticamente las puertas internas para abrirse y las luces del hogar se apagaran, mientras que las luces de emergencia se encenderán de manera automática. Se podrán utilizar las diferentes funcionalidades que posee la aplicación, de manera que se posea control sobre las puertas internas y las luces de emergencia que se posean en el hogar.

Módulo de Interfaz

En este módulo se inicializa todo la interfaz de la aplicación.

```
def control():
    # Contenedor de Luces
    with ui.row().classes("card flex justify-center items-center").style(
        "width: 100%; margin: 5px auto; background-color: #FFFFFF; border-radius:
        20px; padding: 20px;"):
        imgLuces = ui.image(source="Apagado.png").style("flex: 1 1 2rem;")

        with ui.column().style("flex: 1 1 2rem;"):
            ui.label("Luces").classes("text-lg font-bold text-center")
            btnLuces = ui.button("APAGAR", on_click=lambda:
                ControlLuces(btnLuces, imgLuces)).style(
                "color: white; background-color: #D32F2F !important;
                width: 100%; padding: 10px;")

    # Contenedor de Puertas
    with ui.row().classes("card flex justify-center
        items-center").style("width: 100%; margin: 5px auto; background-color:
        #FFFFFF; border-radius: 20px; padding: 20px; "):
        imgPuerta = ui.image(source="puerta_abierta.png").style("flex: 1 1
        2rem;")

        with ui.column().style("flex: 1 1 2rem;"):
            ui.label("Puertas").classes(
                "text-lg font-bold text-center")
            btnPuertas = ui.button("CERRAR", on_click=lambda:
                ControlPuertas(btnPuertas, imgPuerta)).style(
                "color: white; background-color: #D32F2F !important;
                width: 100%; padding: 10px;")
```

Figura 13. Módulo Interfaz 1

```
def estado():
    # Contenedor de Sistema
    with ui.row().classes("card flex justify-center
items-center").style("width: 100%; margin: 10px auto; background-color:
#FFFFFF; border-radius: 20px; padding: 30px;"):
        ui.label("Sistema").classes("text-lg font-bold text-center")

        with ui.row().classes("card flex justify-center
items-center").style("width: 100%;"):
            imgHouse =
ui.image(source="https://img.freepik.com/psd-gratis/ilustracion-casa-propi
edad-3d_23-2151682338.jpg?t=st=1733184632~exp=1733188232~hmac=6d605301309f
c5d3cdea8ddb2787a26d21f2a9132d02a25084bf335494c31d21&w=1380")
            imgHouse.style("width:100%; height: auto;")

            labelEstado = ui.label("Activo" if estado_sistema else
"Inactivo").classes("font-extrabold text-ms")

            btnSistema = ui.button("APAGAR", on_click=lambda:
ControlSistema(btnSistema, labelEstado))
            btnSistema.style("color: white; background-color: #D32F2F
!important; width: 80%; padding: 10px;")
```

Figura 14. Módulo Interfaz 2

```
with ui.row().classes("flex justify-between items-center p-4 w-full"):
    # Logo e Identidad
    with ui.row().classes("flex items-center gap-2"):
        ui.image('terremoto.png').style("border-radius: 50%; width: 50px;
height: 50px;")
        ui.label("Quaksafe").classes("text-lg font-extrabold
text-[#D32F2F]").style('font-family: "Unbounded", sans-serif;')

    # Botón de Salida
    ui.button("Salir", icon="exit_to_app",
on_click=lambda:quit_program()).classes("rounded-full px-4 py-2
text-white").style("background-color: #D32F2F !important; font-size:
12px;")

    # Contenedor general
    with ui.column().classes("card flex justify-center
items-center").style("width: 100%; max-width: 600px; height: calc(100vh -
130px); background: #D32F2F; border-radius: 20px; padding: 20px;"):
        ui.label("Sistema").classes("text-white text-3xl font-bold")
        # Control y Estado
        with ui.tabs().classes('w-full bg-gray-100
rounded-full').style("width: 100%; height: 50px;") as tabs:
            opcionControl = ui.tab('Control')
            opcionEstado = ui.tab('Estado')

        with ui.tab_panels(tabs, value=opcionControl).classes('w-full
bg-transparent p-0'):
            with ui.tab_panel(opcionControl).classes('p-0 items-center'):
                control()
            with ui.tab_panel(opcionEstado).classes('p-0 items-center'):
                estado()

ui.timer(interval=2, callback=check_vibracion)
ui.run()
```

Figura 15. Interfaz de Usuario 3

Módulo de Sensor de vibración

Este módulo consiste en la funcionalidad del sensor de vibración, donde, se necesita el pin del cual se encuentra conectado el sensor, y con ello, el sensor lee si se ha sentido alguna vibración, en lo que debido a lo básico que es el sensor, no tiene un valor específico, pero gracias a su arquitectura se introdujo manualmente la sensibilidad de este.

Por ello, tenemos sólo la comprobación con la variable de vibracionEstado, en el que si esta vibración es alta, HIGH, se hace la activación en cadena por medio de una señal al resto del sistema.

```
const int sensorVibracionPin = 2;

void setup() {
  pinMode(sensorVibracionPin, INPUT);
}

void loop() {
  int vibracionEstado = digitalRead(sensorVibracionPin);

  if (vibracionEstado == HIGH) { // Si el sensor detecta vibración
    Serial.println("VIBRACION"); // Enviar "VIBRACION" a la Raspberry
    Pi
  }
}
```

Figura 16. Módulo Sensor de vibración

Módulo de Relé

Este módulo consiste en la activación del interruptor del relé por medio de un Arduino, de forma que este active y desactive los sensores y los servomotores que se encuentren conectados.

Para ello se necesita el estado del sistema, con el que es una variable booleana, en la cual, dependiendo del estado que se encuentre el sistema, sea en emergencia (en la función `on_mode_emergencia`), el cual es el estado de alerta donde los sensores y servomotores se encuentran activados automáticamente y el estado del sistema se encuentre con el valor `TRUE`, o no se encuentre en emergencia (en la función `off_mode_emergencia`), en el que se encuentra el estado del sistema en `FALSE`.

```
# Funciones para activar/desactivar modos
def on_mode_emergencia():
    global estado_sistema
    GPIO.output(RELE_SUMINISTRO, GPIO.LOW)
    active_led()
    open_door()
    estado_sistema = True

def off_mode_emergencia():
    global estado_sistema
    GPIO.output(RELE_SUMINISTRO, GPIO.HIGH)
    deactivate_led()
    close_door()
    estado_sistema = False
```

Figura 17. Módulo Relé

Módulo de Luces de emergencia

Este módulo consiste en el encendido y apagado de las luces de emergencia, con un estado de las luces led, `luces_activas`, en la tiene un valor booleano , las cuales se encuentran en las funciones `active_led`, en el que el estado de `luces_activas` se cambia al valor `TRUE` y las luces se encienden, y `deactivate_led`, el estado de `luces_activas` cambia a `FALSE` y las luces se apagan en el sistema.

```
# Encender LED de emergencia

def active_led():

    global luces_activas

    GPIO.output(RELE_EMERGENCIA, GPIO.HIGH)

    luces_activas = True

# Apagar LED de emergencia

def deactivate_led():

    global luces_activas

    GPIO.output(RELE_EMERGENCIA, GPIO.LOW)

    luces_activas = False
```

Figura 18. Módulo Luces de emergencia

Módulo de ServoMotores

Este código controla un servomotor utilizando Arduino, moviéndolo a diferentes posiciones según comandos recibidos por la comunicación serial. La última posición del servomotor se guarda en la memoria EEPROM, permitiendo que el motor retome su última posición incluso después de un reinicio. Los comandos 'E' y 'N' permiten mover el servomotor a 90 y 0 grados, respectivamente. La posición del motor se actualiza y guarda en la EEPROM cada vez que se recibe un comando.

```
#include <Servo.h>
#include <EEPROM.h>

Servo servoMotor;
const int servoPin = 13;
int currentPosition = 0; // Posición inicial del servomotor

void setup() {
  Serial.begin(9600);
  servoMotor.attach(servoPin);
  // Recupera la última posición almacenada en la EEPROM
  currentPosition = EEPROM.read(0);
  if (currentPosition < 0 || currentPosition > 180) {
    currentPosition = 0; // Valor por defecto si la EEPROM tiene datos corruptos
  }
  servoMotor.write(currentPosition); // Coloca el servomotor en la última posición conocida
}

void loop() {
  if (Serial.available()) {
    char command = Serial.read();
    if (command == 'E') {
      currentPosition = 90;
    } else if (command == 'N') {
      currentPosition = 0;
    }
    // Guarda la nueva posición en la EEPROM
    EEPROM.write(0, currentPosition);
    servoMotor.write(currentPosition);
  }
  // Mantiene la posición actual del servo
  servoMotor.write(currentPosition);
}
```

Figura 19. Módulo ServoMotors

Conclusiones

En síntesis, se logró planificar un proyecto integral para desarrollar un sistema que sea capaz de monitorear, controlar y alertar a los usuarios sobre situaciones de emergencia, en este caso, terremotos, con el objetivo de resguardar el bienestar de la ciudadanía.

Este sistema, denominado **QuakeSafe**, integra tecnologías avanzadas como IoT y Raspberry Pi para ofrecer una solución innovadora que detecta movimientos telúricos y activa respuestas de seguridad en tiempo real, tales como el corte de energía, la apertura de puertas y el encendido de luces de emergencia.

El funcionamiento del proyecto se basará en una aplicación intuitiva, a través de la cual los usuarios podrán gestionar las luces de emergencia para pruebas del sistema, además de supervisar y controlar el sistema completo. Esta herramienta permitirá verificar el correcto funcionamiento de los dispositivos conectados y garantizará la capacidad de respuesta ante un evento real.

Para alcanzar esta meta, se realizó un análisis exhaustivo de los incidentes sísmicos ocurridos en Chile, lo que permitió identificar los principales desafíos que enfrentan las comunidades ante estos desastres naturales. A partir de esta investigación, se determinó la problemática y se diseñó una solución práctica que optimiza la reacción ante situaciones de emergencia.

El proyecto también definió restricciones y supuestos que aportaron claridad sobre los desafíos y expectativas del desarrollo. Mediante el modelado de procesos, se estructuró de forma eficiente la organización de los componentes y flujos de interacción. Este enfoque permitió una comprensión profunda de las necesidades del usuario, esenciales para definir los requisitos funcionales y no funcionales.

Además, se estableció una arquitectura que separa las responsabilidades entre el cliente y el servidor, optimizando la comunicación, la respuesta del sistema, y garantizando escalabilidad y mantenimiento a largo plazo. Esto asegura que el proyecto sea flexible y capaz de adaptarse a futuras actualizaciones.

En conjunto, estos enfoques proporcionan una base sólida para el desarrollo técnico detallado, incrementando la eficiencia y claridad en cada fase del proyecto. Todo esto se traduce en una experiencia de usuario grata, intuitiva y confiable, reforzando el objetivo principal de **QuakeSafe**: proteger vidas y facilitar la seguridad en momentos críticos.

Trabajo futuro

Optimización de Sensores

- Incorporar sensores de vibración más avanzados o módulos adicionales para mejorar la sensibilidad y precisión en la detección de sismos.
- Explorar el uso de sensores sísmicos especializados que incluyan más parámetros de análisis como frecuencia y duración del movimiento.

Ampliación del Sistema

- Integrar más dispositivos inteligentes, como detectores de gas o cámaras de seguridad, para ofrecer una solución de seguridad integral durante emergencias.
- Expandir la funcionalidad para incluir alertas personalizadas a usuarios a través de notificaciones móviles en tiempo real.

Mejoras en la Interfaz de Usuario (UI/UX)

- Diseñar una interfaz más intuitiva y accesible para usuarios con conocimientos básicos de tecnología.
- Incorporar opciones de accesibilidad para personas con discapacidades visuales o auditivas.

Compatibilidad Multiplataforma

- Ampliar la compatibilidad del sistema a múltiples dispositivos y sistemas operativos (iOS, Android, etc.) para maximizar el alcance del proyecto.

Estudio y Optimización Energética

- Incorporar fuentes de energía renovables para garantizar el funcionamiento continuo del sistema incluso en apagones prolongados.
- Diseñar mecanismos de bajo consumo energético para prolongar la vida útil del hardware conectado.

Escalabilidad del Proyecto

- Diseñar versiones del sistema adaptables a edificios residenciales, oficinas o espacios públicos.
- Implementar infraestructura de red que permita la comunicación y gestión de múltiples dispositivos en áreas amplias.

Documentación y Capacitación

- Crear programas de formación para usuarios y técnicos que administrarán el sistema.
- Actualizar la documentación técnica con cada nueva iteración o mejora del proyecto.

Referencias

Recurso Hardware:

- Raspberry Pi 4: <https://www.amazon.com/GeeekPi-Raspberry-Kit-inicio>
- Arduino R1: <https://arduino.cl/producto/arduino-giga-r1-wifi/>
- Kit claves de conexión: <https://es.aliexpress.com/item/1005002570398150>
- Luz LED: <https://es.aliexpress.com/item/1005004771217994>
- Motor de Rotación: <https://es.aliexpress.com/item/1005006203034102>
- Sensor de Vibración: <https://articulo.mercadolibre.cl/MLC-548831537-sensor-vibracion>
- Relé 220v: <https://articulo.mercadolibre.cl/MLC-566042359-rele-220v>
- Kit resistores: <https://articulo.mercadolibre.cl/MLC-990404596-resistencia>

Recurso Software:

- Licencia Office: <https://www.microsoft.com/es-cl/microsoft-365>
- Canva: https://www.canva.com/es_es/precios
- Sketchup: <https://www.sketchup.com/es/plans-and-pricing>

Leyes y normativas:

- Norma Regulación de los Sistemas Lumínicos en Chile:
https://obtienearchivo.bcn.cl/obtienearchivo?id=repositorio/10221/26691/2/BCN_Regulacion_luminica_en_Chile_DEF.pdf
- Cuadro normativo y tabla de espacios y usos mínimos para el mobiliario:
https://www.minvu.gob.cl/wp-content/uploads/2019/05/Res_7712-16062017-Cuadro-Normativo.pdf
- Eventos extremos y desastres:
<https://sinia.mma.gob.cl/wp-content/uploads/2021/04/17-eventos-extremos-y-desastres.pdf>
- Sueldos Profesionales en Chile:
<https://cl.talent.com/salary>

Recursos Documentación:

- Sensor de Vibración:
 - <https://raspberrysshake.org/>
 - <https://forums.raspberrypi.com/viewtopic.php?t=213463>
 - https://docs.sunfounder.com/projects/ultimate-sensor-kit/en/latest/components_basic/04-component_vibration.html
 - https://joy-it.net/files/files/Produkte/SEN-VIB01/SEN-VIB01_Manual_2024-05-07.pdf