

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática



Plan de proyecto: Wave Unlock

Integrantes:

Gustavo Rios
Kevin Arancibia
Gonzalo De Miguel
Kevin Rojas

Profesor:

Diego Aracena

Asignatura:

Proyecto II

Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
12/09/2023	1.0	Versión preliminar del formato	Kevin Arancibia Gonzalo De Miguel Gustavo Rios Kevin Rojas
25/09/2023	1.1	Mejora del informe en cuanto a recursos que se utilizaran.	Kevin Arancibia Gonzalo De Miguel Gustavo Rios Kevin Rojas
26/09/2023	1.2	Corrección de los Recursos Humanos	Kevin Arancibia Kevin Rojas
20/11/2023	1.3	Mejora del informe	Kevin Arancibia Gonzalo De Miguel Gustavo Rios Kevin Rojas
27/11/2023	1.3.1	Actualización de la Carta Gantt	Kevin Arancibia
13/12/2023	1.4	Mejora de informe 2	Kevin Arancibia Gonzalo De Miguel Gustavo Rios Kevin Rojas
30/12/2023	1.5	Finalizar Inform 3	Kevin Arancibia Gonzalo De Miguel Gustavo Rios Kevin Rojas

Tabla de contenidos

1. Panorama General
 - 1.1. Introducción
 - 1.2. Escenario del Problema
 - 1.3. Escenario de la Solución
 - 1.4. Resumen del Proyecto
 - 1.4.1. Propósito
 - 1.4.2. Alcance
 - 1.5. Objetivos del Proyecto
 - 1.5.1. Objetivo general
 - 1.5.2. Objetivos específicos
 - 1.6. Suposiciones
 - 1.7. Restricciones
 - 1.8. Entregables del Proyecto
2. Organización del Proyecto
 - 2.1. Personal y entidades internas
 - 2.2. Roles y responsabilidades
 - 2.3. Mecanismos de Comunicación
3. Planificación de los procesos de gestión
 - 3.1. Planificación inicial del proyecto
 - 3.1.1. Planificación de estimaciones
 - 3.1.2. Planificación de Recursos Humanos
 - 3.2. Lista de actividades (carta Gantt)
 - 3.2.1. Actividades de trabajo
 - 3.2.2. Asignación de tiempo
 - 3.3. Planificación de la gestión de riesgos
4. Planificación de los procesos técnicos
 - 4.1. Modelo de proceso
 - 4.1.1. Modelo de Diseño(casos de uso)
 - 4.1.2. Modelo de Diseño (diagrama de clases)
 - 4.1.3. Modelo de Interacción (diagrama de secuencia)
 - 4.1.4. Plan de Integración
 - 4.2. Herramientas y técnicas
5. Conclusión
6. Referencias

1. Panorama General

1.1. Introducción

En esta segunda fase del proyecto nos enfocamos en poder implementar lo pensado en la fase anterior para así mejorar la accesibilidad de las personas con movilidad reducida o discapacidad que representa un paso significativo hacia la construcción de entornos inclusivos y equitativos, encontrando soluciones innovadoras, integrando tecnologías avanzadas para facilitar la vida diaria de quienes enfrentan desafíos de movilidad.

Se tiene conciencia sobre las barreras físicas que deben enfrentar las personas con discapacidades a menudo los cuales se traducen en desafíos diarios, como puede ser, abrir una puerta cuya tarea se sabe que requiere un esfuerzo considerable para algunos, y en algunos casos, es simplemente imposible. Nuestra solución, impulsada por la versátil Raspberry Pi, aborda este problema fundamental al aprovechar la tecnología para hacer que las puertas de los hogares sean accesibles de manera fácil y segura.

A diferencia de la fase anterior, se realizaron unos cambios haciendo que el proyecto se vuelva algo más económico, se reemplazó la detección facial por un lector de códigos QR, el cuál tendrá la misma función como se tenía pensado y los sensores ultrasónicos para crear un sistema inteligente que reconoce a las personas que se acercan a la puerta.

El proyecto en cuestión es un testimonio de cómo la tecnología puede ser una herramienta poderosa para superar barreras y mejorar la calidad de vida de las personas con discapacidades. Juntos, daremos un paso más hacia la creación de un mundo más inclusivo, donde la tecnología esté al servicio de todos.

1.2. Escenario del Problema

Actualmente en la sociedad, las personas con discapacidades o movilidad reducida enfrentan desafíos diarios al realizar tareas aparentemente simples, como abrir las puertas. Este problema se agrava cuando se trata de puertas que requieren de mucho esfuerzo. Abrir una puerta puede convertirse en una tarea difícil, lo que limita la independencia y la movilidad de las personas dentro o fuera de su hogar. Las barreras físicas se traducen en una disminución de la calidad de vida y la necesidad de depender de la ayuda de terceros para realizar las actividades cotidianas.

1.3. Escenario de la solución

Para abordar este desafío, hemos concebido el proyecto “Hands Free Home”, que utiliza tecnología innovadora para proporcionar accesibilidad y movilidad mejoradas a personas con discapacidades o movilidad reducida.

El sistema “Hands Free Home” utiliza un lector de código QR y los sensores ultrasónicos para identificar de manera automática a las personas que se acercan a la puerta.

Este proceso no solo facilita la accesibilidad, sino que también garantiza la seguridad e independencia de las personas con discapacidades.

En resumen, “Hands Free Home” promete no solo abrir las puertas dentro del hogar, sino también las puertas exteriores de manera independiente al detectar a su usuario.

1.4. Resumen del proyecto

1.4.1. Propósito

El propósito de este proyecto busca abordar el desafío constante de accesibilidad y autonomía por parte de las personas con discapacidades o movilidad reducida dentro de sus hogares. Por ello, se busca una implementación de un sistema basado en Raspberry Pi que proporcione una solución de integración de apertura de puertas automáticas y seguras. Contribuyendo al usuario una mejora significativa a la calidad de vida, brindando independencia y facilidad de movilidad en el entorno doméstico.

1.4.2. Alcance

El alcance que se busca con “Hands Free Home” trata de enfocar el diseño y desarrollo de un sistema de apertura de puertas basado en Raspberry Pi, que incorporará una cámara capaz de detectar códigos QR y sensores ultrasónicos para facilitar un acceso sin contacto físico y seguro en hogares, está especialmente destinado a usuarios con discapacidades o movilidad reducida, pero podría ser utilizado por más que solo este círculo ya que es fácil de utilizar por su autonomía.

1.5. Objetivos del proyecto

1.5.1. Objetivo General

Desarrollar un sistema de monitoreo y control de acceso residencial, denominado “Wave Unlock”, que emplee tecnologías como Raspberry Pi, sensores ultrasónicos y un lector de código QR.

1.5.2. Objetivos Específicos

- I. Identificar y analizar la problemática de la discapacidad en Chile, según datos de la Encuesta Nacional de Discapacidad (ENDISC) 2022.
- II. Desarrollar una solución técnica basada en Raspberry Pi que incluye sensores de ultrasonido y una cámara con lector de código QR.
- III. Evaluar los beneficios de las soluciones propuestas, que ayuden en la movilidad y la identificación segura de las personas.
- IV. Diseñar el hardware necesario, incluyendo la Raspberry Pi, la cámara y los sensores ultrasónicos, y ensamblarlos en un sistema funcional.
- V. Desarrollar el sistema para identificar los usuarios para abrir o cerrar la puerta si corresponde.
- VI. Realizar pruebas y análisis de resultados.

1.6. Suposiciones

- I. **Condiciones Ambientales Estándar:** Se supone que el sistema funcionará en condiciones ambientales estándar, como temperaturas dentro de rangos normales.. Se podrían requerir ajustes adicionales para entornos extremos.
- II. **Usuarios:** Se asume que los usuarios que utilicen mayoritariamente este producto serán aquellos que padecen alguna discapacidad o movilidad reducida.
- III. **Acceso a Electricidad y Conectividad:** Se supone que el sistema tendrá acceso constante a la electricidad y a una red de conectividad para su funcionamiento y actualizaciones.

1.7. Restricciones

- I. **Maquetación:** Hacer una presentación física con gastos limitados en \$10.000 y utilizar materiales reciclables o modelado virtual que demuestre la funcionalidad del producto.
- II. **Raspberry PI:** Utilizar de forma obligatoria esta micro computadora.
- III. **Usuarios:** Debe ser funcional para usuarios que tengan alguna discapacidad física o móvil.

1.8. Entregables

- I. Informe n°1 y presentación de la planificación del proyecto.
- II. Informe n°2 y presentación del avance del proyecto.
- III. Informe Final y presentación final del proyecto.
- IV. Maqueta o modelo virtual.
- V. Bitácora semanal.
- VI. Wiki (Redmine)
- VII. Manual de usuario.
- VIII. Producto final.

2. Organización del Proyecto

2.1. Personal y entidades internas

Este proyecto cuenta con un equipo de expertos en informática altamente capacitados, además de colaboradores en áreas complementarias. A continuación, se detalla el personal y las entidades internas involucradas:

Jefe de proyecto: Responsable de la planificación, coordinación y supervisión general del proyecto. Debe asegurarse de que se cumplan los plazos, el presupuesto y los objetivos, además de gestionar el equipo y las comunicaciones.

Ingeniero informático (especializado en Hardware): Encargado del diseño, desarrollo y ensamblaje del hardware necesario, incluyendo la Raspberry Pi, la cámara y los sensores ultrasónicos. Debe asegurar que los componentes sean compatibles y funcionen correctamente.

Ingeniero informático (especializado en Software): Responsable de desarrollar el software de detección de código QR, programar la interacción entre los sensores ultrasónicos y la Raspberry Pi, y crear la interfaz de usuario para la configuración del sistema.

Especialista en Accesibilidad: El especialista en accesibilidad es fundamental para garantizar que el sistema sea usable por personas con discapacidades. Colabora con el equipo para recopilar retroalimentación de usuarios con discapacidades, realiza pruebas de usabilidad y asegura que el sistema cumpla con estándares de accesibilidad.

2.2. Roles y responsabilidades

Es fundamental definir claramente las responsabilidades de cada miembro del equipo:

Rol	Responsable	Involucrados
Jefe de Proyecto	Kevin Arancibia	Gustavo Rios
Ingeniero informático (Hardware)	Gustavo Rios	Kevin Arancibia, Kevin Rojas, Gonzalo De Miguel
Ingeniero informático (Software)	Gonzalo De Miguel	Kevin Arancibia, Gustavo Rios, Kevin Rojas
Especialista en Accesibilidad	Kevin Rojas	Gonzalo De Miguel

2.3. Mecanismos de Comunicación

Para garantizar un flujo de trabajo eficiente y eficaz, se han establecido los siguientes mecanismos de comunicación:

- **Reuniones Semanales:** Cada miércoles y jueves a las 16:20 PM, el equipo se reunirá tanto física como virtualmente para revisar avances, plantear desafíos y ajustar la dirección del proyecto si es necesario
- **Comunicación Directa:** Cada miembro del equipo podrá comunicarse de manera rápida y efectiva a través de Discord y WhatsApp.
- **Avance del proyecto:** Se irá actualizando tanto la wiki del proyecto como la carta gantt y se estarán subiendo bitácoras semanales a través de Redmine.

3. Planificación de los procesos de gestión

3.1. Planificación inicial del proyecto

3.1.1. Planificación de estimaciones:

Costos de Software	Estimación
Licencia de MicroSoft 365 Empresa	\$22.000 CLP
Python	\$0 CLP

Total de Software: \$22.000

Costos de Hardware	Estimación (por unidad)
Raspberry PI 3 (modelo B)	\$100.000 CLP
Notebook	\$500.000 CLP
Sensor Ultrasónico	\$3.500 CLP
Cámara Raspberry Pi	\$17.000 CLP

Total de Hardware: \$620.000

3.1.2. Planificación de Recursos Humanos:

Integrante	Rol Responsable e Involucrado	Valor (por hora)	Hora Mensual (48 horas)	Costo Mensual
Kevin Arancibia	Jefe de Proyecto	\$9.200 CLP	10	92.000
	Ingeniero en Software	\$8.300 CLP	25	207.500
	Ingeniero en Hardware	\$6.760 CLP	13	87.880
Gustavo Rios	Jefe de Proyecto (parc)	\$9.200 CLP	5	46.000
	Ingeniero en Software	\$8.300 CLP	15	124.500
	Ingeniero en Hardware	\$6.760 CLP	28	189.280
Kevin Rojas	Ingeniero en Software	\$8.300 CLP	20	166.000
	Ingeniero en Hardware	\$6.760 CLP	13	87.880
	Esp. de Accesibilidad	\$6.500 CLP	15	97.500
Gonzalo De Miguel	Ingeniero en Software	\$8.300 CLP	20	166.000
	Ingeniero en Hardware	\$6.760 CLP	13	87.880
	Esp. de Accesibilidad	\$6.500 CLP	15	97.500
			Total (4 meses):	\$1.449.920 \$5.799.680

Costos Total	Estimación
Software	\$22.000 CLP
Hardware	\$653.500 CLP
Recursos Humanos	\$5.799.680 CLP
Total Proyecto	\$6.475.180

3.2. Lista de actividades

3.2.1. Actividades de trabajo:

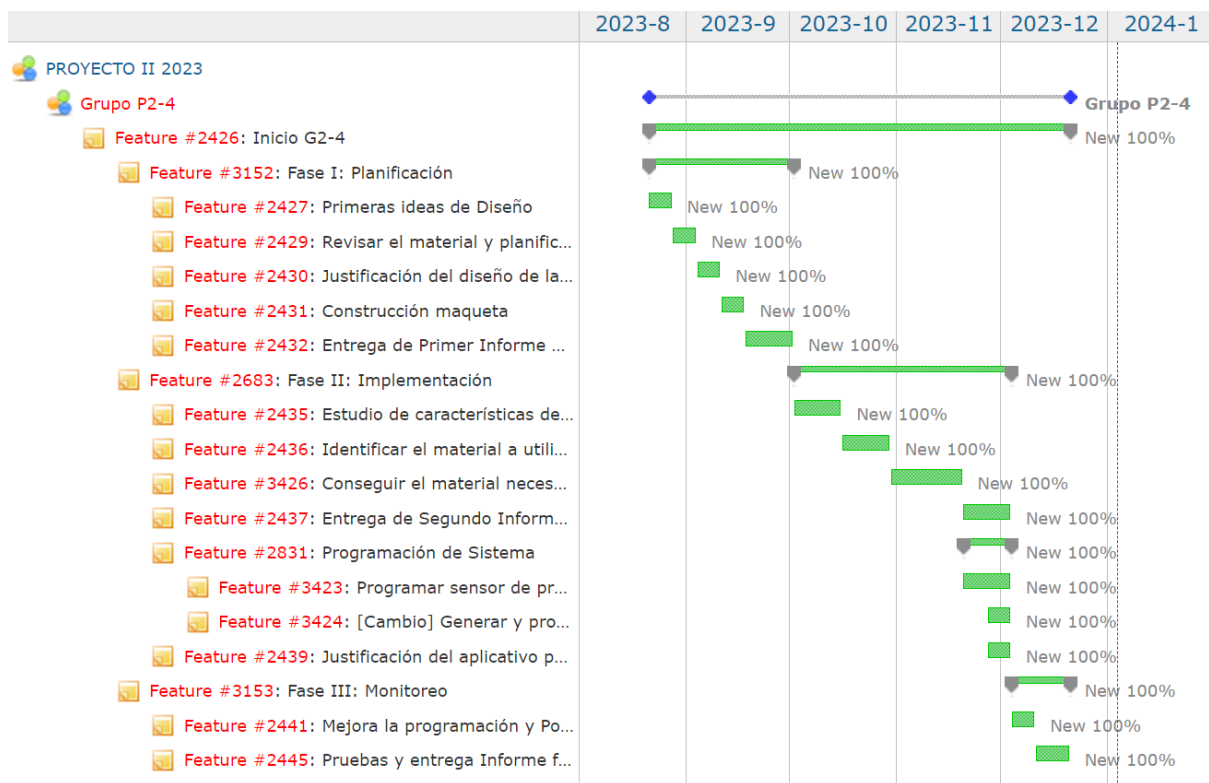
Diseñamos varias tareas para comenzar a realizar este informe:

- Primeras ideas de Diseño
- Revisar el material y planificar una maqueta
- Justificación del diseño de la maqueta
- Construcción maqueta
- Estudio de características de Raspberry PI
- Justificación de la arquitectura de comunicación.
- Justificación del aplicativo para la comunicación con el dispositivo móvil
- Programar la interfaz y realizar el Póster promocional

Además designamos encargados para hacer las Bitácoras o otras tareas más detalladas, como la programación e implementación del programa.

3.2.2. Asignación de tiempo(actualizada):

Creamos la carta gantt para poder definir en qué lapsos de tiempo debemos realizar tales labores.



3.3. Planificación de la gestión de riesgos

Niveles de impacto

Mientras que para los riesgos latentes que se podrían presentar en el proyecto, se clasificaron en 4 niveles de impacto:

1. Catastrófico
2. Crítico
3. Marginal
4. Despreciable

Riesgos	Probabilidad de Ocurrencia	Nivel de impacto	Acción remedial
Fallos de hardware de Raspberry Pi.	10%	2	Realizar pruebas exhaustivas de hardware y software antes de la implementación.
Riesgos de seguridad en el código QR	50%	3	Implementar una validación del contenido del código QR mediante filtros de seguridad y actualizar regularmente los algoritmos de escaneo para mitigar posibles vulnerabilidades y riesgos de manipulación
Fallos en el suministro eléctrico.	40%	3	Implementar sistemas de energía ininterrumpida (UPS).
Mantenimiento inadecuado.	20%	4	Establecer un plan de mantenimiento preventivo.
Fallo de precisión y/o reconocimiento facial	40%	3	Mantener actualizado el software de seguridad y capacitar al personal en prácticas seguras.
Acceso no autorizado.	30%	2	Informar a los usuarios sobre la recopilación y uso de datos de reconocimiento facial
Posibles litigios por problemas de seguridad.	30%	3	Establecer una política de retención de datos.
Inasistencia de personal	50%	3	Mantener un orden de turnos para asistencia completa

Falta de adaptación al entorno inteligente	60%	3	Capacitación para el uso correcto del entorno inteligente
--	-----	---	---

4. Planificación de Procesos Técnicos

4.1. Modelo de Proceso

4.1.1. Modelo de Diseño (Casos de uso)

a) **Caso de Uso:** Acceder al hogar por la puerta principal

Nombre CUS: Acceder al hogar por la puerta principal
Autor/Fecha: Kevin Arancibia 30-12-23
Resumen: El usuario se acerca y muestra la tarjeta con el código QR para ingresar a su hogar.
Actor: Usuario
Condición previa: Tener el equipo instalado y los datos del código QR ingresados en la BD.
<p>Flujo Principal: Usuario</p> <ol style="list-style-type: none"> 1.- El <u>usuario</u> se acerca a la puerta principal y muestra el código QR. 2.- El <u>cámara</u> escanea el código QR a través de la cámara y revisa si está registrado en la BD. 3.- El sistema identifica el código QR proporcionado siendo válido el rut, encendiéndose la Luz verde del Led. 4.- El <u>sensor de proximidad</u> determina el rango de distancia. 5.- El <u>usuario</u> se acerca a la puerta de entrada. 6. El <u>sensor de proximidad</u> detecta la presencia del <u>usuario</u> y el <u>sistema</u> envía la señal al servomotor conectado a la perilla de entrada y al servomotor conectado con el brazo retráctil. 7.- El <u>usuario</u> ingresa correctamente a su hogar. 8.- El <u>sensor de proximidad</u> deja de detectar la presencia del <u>usuario</u> en la entrada a través de sus ondas ultrasónicas y procede a cerrar la puerta.
<p>Flujo Alternativo: Usuario</p> <ol style="list-style-type: none"> 3.1.- El <u>sistema</u> identifica el código QR proporcionado, siendo inválido ya que no está en la BD el rut correspondiente al usuario o invitado, manteniéndose la Luz Roja del Led.
Postcondiciones: ---
Valor medible: Una mayor accesibilidad del usuario que presenta algún problema de movilidad.

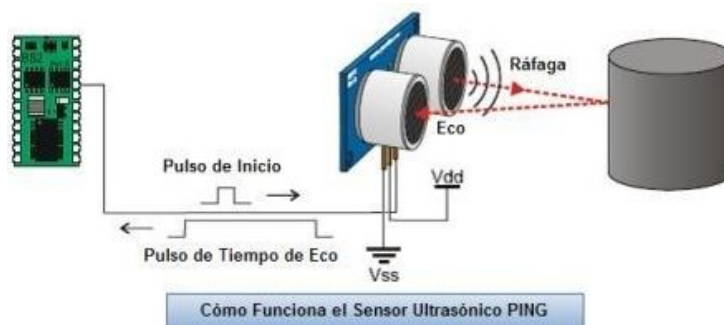
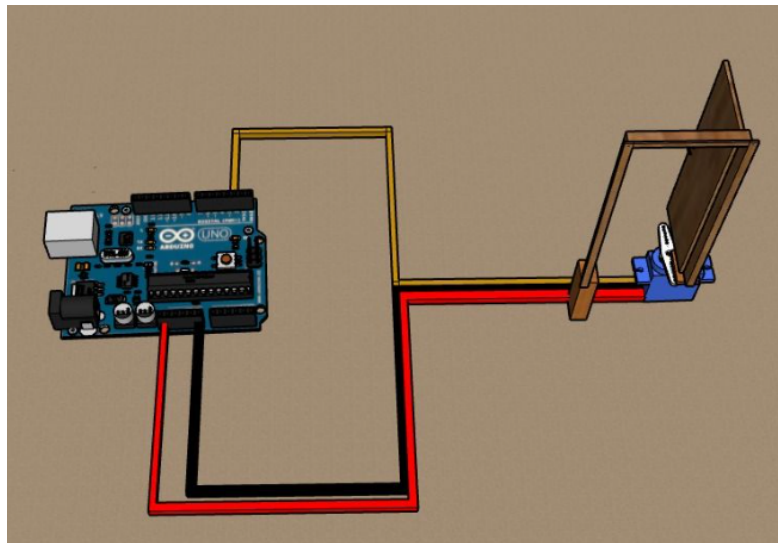
b) Caso de Uso: Abrir y cerrar puertas interiores

Nombre CUS: Abrir y Cerrar puertas interiores
Autor/Fecha: Gonzalo De Miguel 30-12-23
Resumen: El usuario se acerca a una puerta interior siendo detectado por el sensor ultrasónico, abriendo la misma.
Actor: Usuario
Condición previa: ---
Flujo Principal: Usuario 1.- El <u>usuario</u> se acerca a la puerta interior. 2.- El <u>sensor ultrasonido</u> detecta la presencia en su rango de alcance y envía una señal al <u>sistema</u> . 3.- El <u>sistema</u> envía la señal al servomotor conectado a la perilla de entrada y al <u>servomotor</u> conectado con el brazo retráctil. 4.- El <u>servomotor</u> da un giro de 180° permitiendo abrir la puerta con el brazo retráctil. 4.- El <u>usuario</u> ingresa a la habitación correspondiente. 5.- El <u>sensor ultrasónico</u> no detecta a nadie en el rango de detección. 6.- El <u>sistema</u> espera 7 segundos para enviar la señal a los <u>servomotores</u> para el cierre la puerta.
Flujo Alternativo: Usuario 5.- El sensor ultrasónico detecta aún la presencia en su rango de detección, envía una señal al sistema para que el servomotor no cierre la puerta.
Postcondiciones: ---
Valor medible: Una mayor accesibilidad, comodidad y fiabilidad dentro del hogar.

4.1.1.1. Descripción de la Arquitectura

Conexión: Raspberry - Sensor Ultrasónico - Servo Motor/Brazo Retráctil

En el caso de la puerta interior, se visualiza cómo será la conexión entre los componentes, en este caso como la fuente principal utilizará conexiones para llegar al sensor ultrasónico, tras al detectar una presencia que interfiera en sus ondas ultrasónicas enviaría una señal al servomotor para efectuar la apertura de la puerta con el brazo retráctil, en cambio, la puerta exterior requeriría una aprobación tras leer el código QR para así efectuar la apertura.

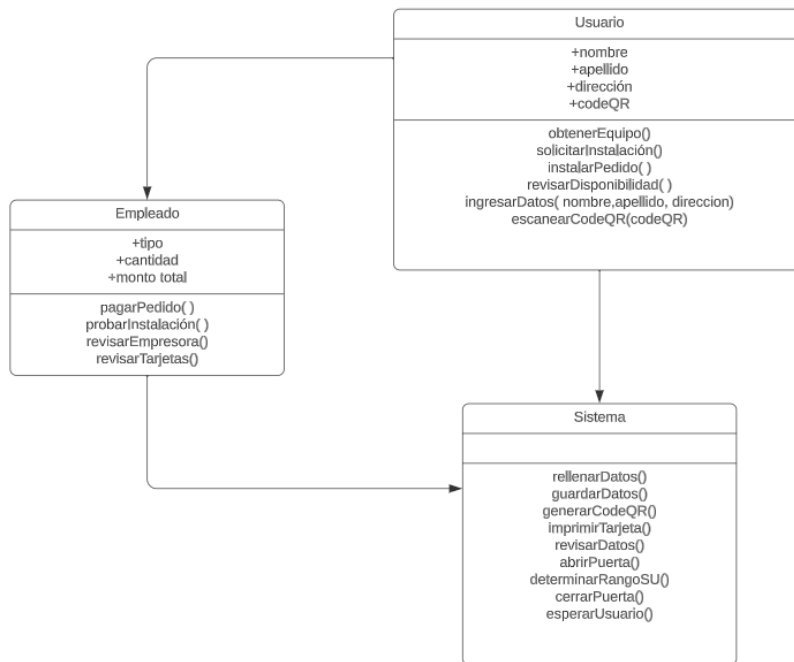


4.1.1.2. Especificación de requerimientos

Para comodidad del usuario, se necesitaran los siguientes requerimientos:

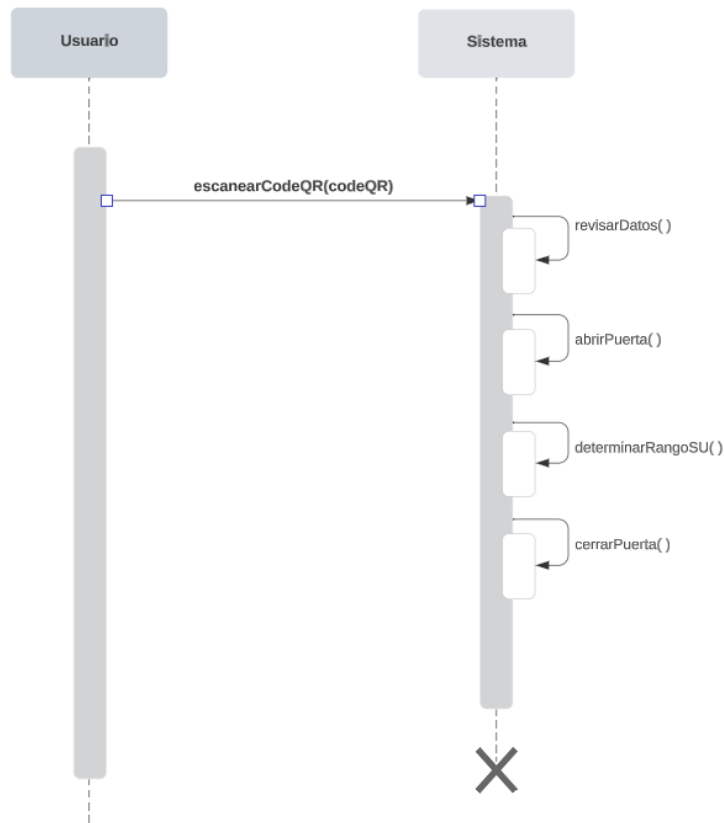
- a) Establecer un rango de detección ajustable para adaptarse a diferentes entornos.
- b) Lograr la apertura automática de puertas cuando se detecta una persona con movilidad reducida.
- c) Incorporar una cámara de seguridad con capacidad para leer códigos QR.
- d) Capturar imágenes de alta resolución para garantizar la precisión del reconocimiento QR.
- e) Implementar e interpretar códigos QR específicos para validar la autorización de acceso a ciertos usuarios.
- f) Utilizar una Raspberry Pi como unidad central de procesamiento y control.
- g) Desarrollar software para coordinar la interacción entre sensores, cámara y mecanismos de apertura de puertas.

4.1.2. Modelo de Diseño (Diagrama de Clases)

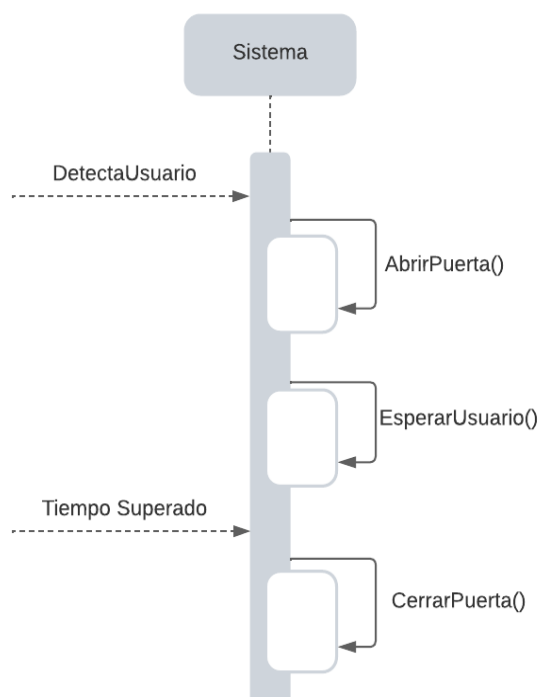


4.1.3. Modelo de Interacción (Diagrama de Secuencia)

a) Diagrama de Secuencia: Acceder al hogar por la puerta principal



b) Diagrama de Secuencia: Abrir y cerrar puertas interiores



4.1.4. Plan de Integración

1. Identificación de Stakeholders:

- Identificar y contactar a organizaciones y comunidades relacionadas con personas con discapacidades o movilidad reducida.

2. Desarrollo del prototipo:

- Configuración de Raspberry Pi con los componentes necesarios: sensores ultrasónicos, cámara para detección facial y lector de código QR.
- Desarrollo y prueba del software que permite la detección y reconocimiento de usuarios, así como la activación automática de las puertas.

3. Prueba piloto:

- Identificar hogares de prueba con personas que tienen discapacidades o movilidad reducida.
- Instalar y probar el sistema "Hands Free Home" en estos hogares para evaluar su efectividad y realizar ajustes según sea necesario.

4. Recolección de retroalimentación:

- Obtener comentarios de los usuarios piloto sobre la facilidad de uso, eficacia y cualquier mejora sugerida.
- Realizar ajustes en el sistema en función de la retroalimentación recibida.

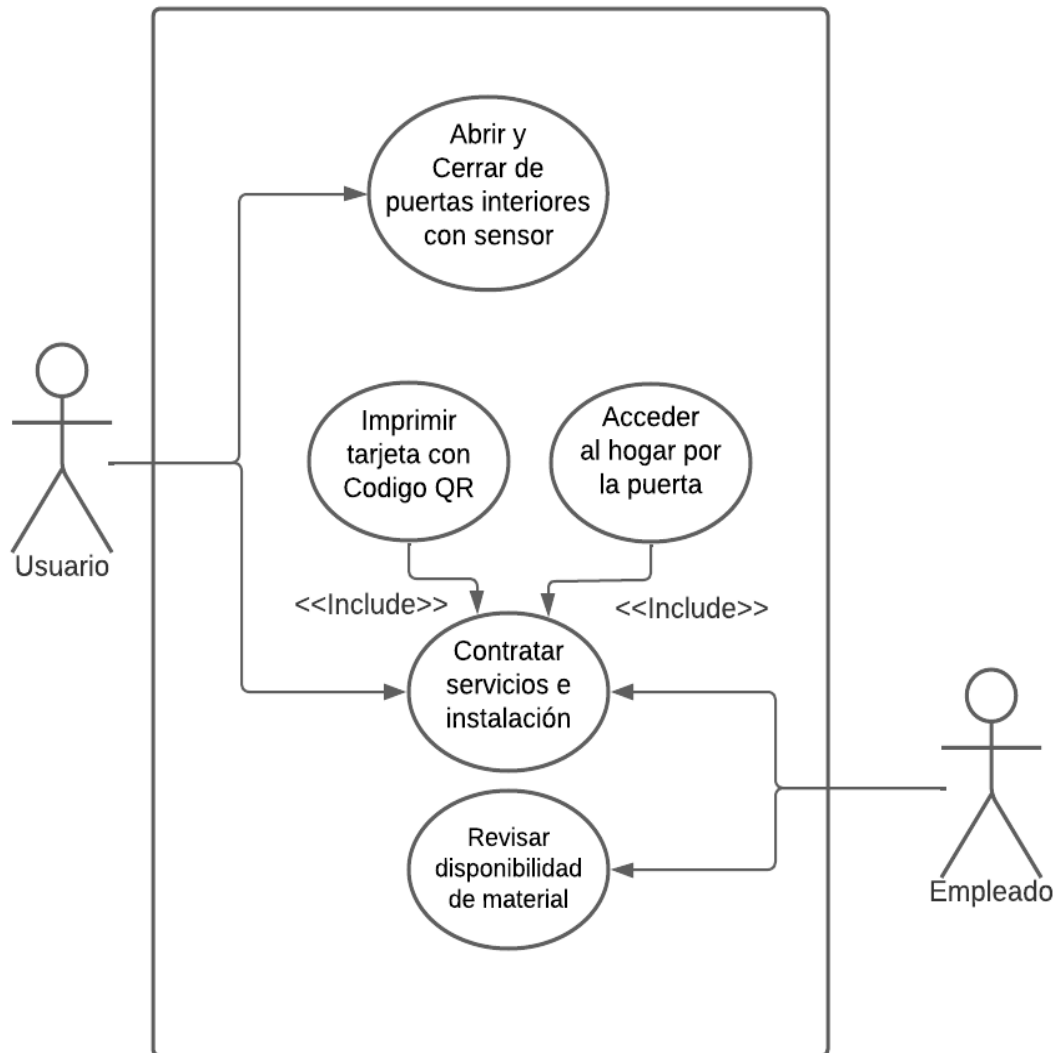
5. Implementación Completa:

- Desarrollar manuales de usuario y materiales de capacitación.
- Implementar el sistema en hogares adicionales, teniendo en cuenta diferentes configuraciones y necesidades específicas.

6. Mejora continua y evaluaciones a largo plazo:

- Explorar oportunidades para expandir la accesibilidad a otros entornos, como oficinas, instituciones públicas y comercios
- Realizar evaluaciones regulares para asegurar que el sistema cumple con las expectativas y necesidades cambiantes de la comunidad.

4.1.4.1 Descripción de Arquitectura (vista desde los módulos del caso de uso)



4.1.4.2 Modelo de implementación

El proyecto "Wave Unlock" busca proporcionar accesibilidad y movilidad mejoradas a personas con discapacidades o movilidad reducida dentro de su hogar. Este sistema revolucionario se basa en la combinación de la versátil Raspberry Pi, sensores ultrasónicos y un lector de código QR para crear un entorno hogareño más inclusivo.

4.1.4.3 Módulos implementados

1. Sensores Ultrasónico:

Descripción: Integración de sensores ultrasónicos para medir la distancia entre la persona y la puerta.

Tecnologías utilizadas: Sensores ultrasónicos compatibles con Raspberry Pi.

2. Lector de Código QR:

Descripción: Utiliza un lector de código QR para una identificación adicional o como opción de respaldo.

Tecnologías utilizadas: Bibliotecas de lectura de códigos QR.

3. Automatización de Puertas:

Descripción: Actúa sobre la base de la detección facial o proximidad, permitiendo la apertura automática de puertas.

Tecnologías utilizadas: Motores de puertas eléctricos controlados por la Raspberry Pi.

4. Interfaz de Usuario (Opcional):

Descripción: Desarrollo de una interfaz para la configuración y gestión del sistema, aunque se prioriza la automatización sin intervención del usuario.

Tecnologías utilizadas: Interfaz web o aplicación móvil.

4.2. Herramientas y técnicas

Herramientas: Geany, Consola de comandos, Visual Studio Code, redmine

Sensores:

- Sensor ultrasónico de rango: permite medir la distancia por medio de retorno de ondas de ultrasonido.
- Cámara raspberry 1.3: se implementa para leer un código QR entregado por el usuario para la apertura de la puerta.
- Servo Motor: permite mover los mecanismos de apertura de puerta y del cierre de la cerradura.
- OS Raspberry: Sistema operativo para Raspberry
- Python3: lenguaje que se usará para la programación en general.
- Técnicas:
 - Dividir para conquistar: dividiendo el problema de la programación en partes manejables como son el manejo de los distintos sensores por separado para luego poder usar los datos para el funcionamiento completo.
- Implementación: consta de una programación sencilla para poder extraer los datos de los sensores a una interfaz gráfica la cual mostrarán los resultados de estos obtenidos en pantalla. De esta forma se procede analizar y encontrar las medidas promedio para el funcionamiento del sistema general.

4.3. Código

```
import cv2
from PIL import Image, ImageTk
from pyzbar import pyzbar
import tkinter as tk
from tkinter import Label, Entry, Button
import RPi.GPIO as GPIO
import time

# Configurar los pines del sensor ultrasónico, el servo motor y los pines del LED RGB
TRIG = 26
ECHO = 16
SERVO_PIN = 12
LED_R = 27 # Pin para el color rojo del LED RGB
LED_G = 17 # Pin para el color verde del LED RGB
LED_B = 22 # Pin para el color azul del LED RGB (no se usa en este caso)

# Inicializar GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
GPIO.setup(SERVO_PIN, GPIO.OUT)
GPIO.setup(LED_R, GPIO.OUT)
GPIO.setup(LED_G, GPIO.OUT)
GPIO.setup(LED_B, GPIO.OUT)

# Función para cambiar el color del LED RGB
def set_rgb_color(red, green, blue):
    GPIO.output(LED_R, red)
    GPIO.output(LED_G, green)
    GPIO.output(LED_B, blue)

class QRCodeReaderApp:
    def __init__(self, root, camera_index=0):
        self.root = root
        self.root.title("Lector de Código QR")
        self.camera = cv2.VideoCapture(camera_index)
        self.camera.set(cv2.CAP_PROP_FPS, 30)
        self.camera_width = 640
        self.camera_height = 480
        self.camera.set(cv2.CAP_PROP_FRAME_WIDTH, self.camera_width)
        self.camera.set(cv2.CAP_PROP_FRAME_HEIGHT, self.camera_height)

        self.label = Label(root)
        self.label.pack(padx=10, pady=10)
```



```

self.qr_content_label = Label(root, text="Contenido del QR:")
self.qr_content_label.pack(padx=10, pady=10)

self.distancia_label = Label(root, text="Distancia: -- cm")
self.distancia_label.pack(padx=10, pady=10)

self.quit_button = Button(root, text="Cerrar", command=self.close_app)
self.quit_button.pack(pady=10)

self.new_qr_entry = Entry(root)
self.new_qr_entry.pack(pady=10)

self.confirm_button = Button(root, text="Confirmar QR", command=self.confirm_qr)
self.confirm_button.pack(pady=10)

self.QR_VALIDOS = ["20.216.163-4"]
self.distance_measurement_active = False
self.ultrasonic_sensor_active = False

set_rgb_color(GPIO.HIGH, GPIO.LOW, GPIO.LOW) # Inicializar LED en rojo

self.read_qr_code()

def read_qr_code(self):
    _, frame = self.camera.read()
    decoded_qr_codes = pyzbar.decode(frame)
    for qr_code in decoded_qr_codes:
        qr_data = qr_code.data.decode('utf-8')
        self.show_qr_result(qr_data)

frame = self.resize_frame(frame, self.camera_width, self.camera_height)
img = self.convert_frame_to_image(frame)
self.label.imgtk = img
self.label.configure(image=img)

if self.distance_measurement_active and self.ultrasonic_sensor_active:
    distancia = self.medir_distancia()
    self.distancia_label.config(text=f"Distancia: {distancia:.2f} cm")
    if distancia < 10:
        self.mover_servo(90)
        time.sleep(3)
        self.mover_servo(0)
        set_rgb_color(GPIO.HIGH, GPIO.LOW, GPIO.LOW) # Cambiar LED a rojo
después de mover el servo
        self.ultrasonic_sensor_active = False
self.root.after(10, self.read_qr_code)

def show_qr_result(self, qr_data):

```

```

if qr_data in self.QR_VALIDOS:
    set_rgb_color(GPIO.LOW, GPIO.HIGH, GPIO.LOW) # Cambiar LED a verde
    self.distance_measurement_active = True
    self.ultrasonic_sensor_active = True
else:
    set_rgb_color(GPIO.HIGH, GPIO.LOW, GPIO.LOW) # Mantener LED en rojo
    self.qr_content_label.config(text=f"Contenido del QR: {qr_data}")

def convert_frame_to_image(self, frame):
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    img = Image.fromarray(frame_rgb)
    img = ImageTk.PhotoImage(img)
    return img

def resize_frame(self, frame, new_width, new_height):
    frame = cv2.resize(frame, (new_width, new_height))
    return frame

def close_app(self):
    self.camera.release()
    GPIO.cleanup()
    self.root.destroy()

def confirm_qr(self):
    nuevo_qr = self.new_qr_entry.get().strip()
    if nuevo_qr:
        self.QR_VALIDOS.append(nuevo_qr)
        set_rgb_color(GPIO.HIGH, GPIO.LOW, GPIO.LOW) # Asegurarse de que el LED
esté en rojo
        self.ultrasonic_sensor_active = True
    else:
        messagebox.showerror("Error", "Por favor, ingrese un código QR válido.")

def medir_distancia(self):
    GPIO.output(TRIG, True)
    time.sleep(0.00001)
    GPIO.output(TRIG, False)

    start_time = time.time()
    end_time = time.time()

    while GPIO.input(ECHO) == 0:
        start_time = time.time()

    while GPIO.input(ECHO) == 1:
        end_time = time.time()

    duration = end_time - start_time

```

```
distancia = (duration * 34300) / 2  
return distancia
```

```
def mover_servo(self, angulo):  
    pwm = GPIO.PWM(SERVO_PIN, 35)  
    pwm.start(0)  
    duty = angulo / 18 + 2  
    GPIO.output(SERVO_PIN, True)  
    pwm.ChangeDutyCycle(duty)  
    time.sleep(1)  
    GPIO.output(SERVO_PIN, False)  
    pwm.ChangeDutyCycle(0)  
    pwm.stop()
```

```
def main():  
    root = tk.Tk()  
    app = QRCodeReaderApp(root)  
    root.mainloop()
```

```
if __name__ == "__main__":  
    main()
```

5. Conclusión

En conclusión, el proyecto de accesibilidad basado en Raspberry Pi representa no solo una solución tecnológica eficaz, sino también un paso significativo hacia la construcción de una sociedad más inclusiva en Chile. La combinación de sensores de ultrasonido y lectores de códigos QR no solo mejora la accesibilidad física de las personas con discapacidad, sino que también garantiza una identificación segura y eficiente de los usuarios.

La planificación cuidadosa de la gestión de riesgos y la implementación exitosa del proyecto subrayan la viabilidad y la sostenibilidad de esta iniciativa. Al mejorar la calidad de vida y la autonomía de las personas con discapacidad, el proyecto no solo abre puertas físicas, sino que también desempeña un papel crucial en la apertura de oportunidades y la promoción de la participación plena en la sociedad. En última instancia, este proyecto no solo transforma la forma en que enfrentamos los desafíos de accesibilidad, sino que también contribuye significativamente a la construcción de un entorno más equitativo donde cada individuo, independientemente de sus capacidades, pueda disfrutar de una igualdad de oportunidades.

6.Referencias

[1] ENDISC, 2022:

<https://fundacioncontrabajo.cl/blog/cultura-inclusiva/discapacidad-en-chile/>

[2] RaspBerry PI:

<https://www.ionos.es/digitalguide/servidores/know-how/un-vistazo-a-proyectos-basados-en-raspberry-pi/>

[3] Tutorial de arduino y sensor ultrasonico

https://naylampmechatronics.com/blog/10_tutorial-de-arduino-y-sensor-ultrasonico-hc-sr04.html

[4] Wiki Grove Sensor Ultrasónico

https://wiki.seeedstudio.com/Grove-Ultrasonic_Ranger/

[5] Pines GPIO

<https://solectroshop.com/es/content/60-5-pines-gpio-y-su-programacion#:~:text=Un%20pin%20GPIO%20designado%20como%20pin%20de%20entrada%20se%20puede.puede%20configurar%20en%20el%20software.>

[6] Material ejemplo casos de uso

<https://lsi2.ugr.es/~mvega/docis/casos%20de%20uso.pdf>

[7] Posibles riesgos en proyectos

<https://es.linkedin.com/pulse/los-100-riesgos-en-la-gesti%C3%B3n-de-proyectos-luciano-m-oreira-da-cruz>

[8] Características de Raspberry

<https://raspberrypi.cl/raspberry/>

[9] Creación de CUS

https://www.google.com/aclk?sa=l&ai=DChcSEwiYh6zBg-eCAxWzQkgAHa01DrQYABAAGgJjZQ&ae=2&gclid=CjwKCAiAvJarBhA1EiwAGgZl0Dv7c80r-2R4FPdywt7CfKuvMdnQg6DExWuMWBZeZi20JvoLzuDWDdBoCuAQQAvD_BwE&sig=AOD64_3GEgbMZTxwJgNcNDbCnQvmlDceEA&q&adurl&ved=2ahUKEwiL56bBg-eCAxWrqJUCHfY0BPwQ0Qx6BAgFEAE