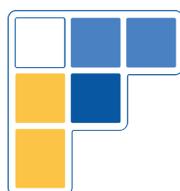


# UNIVERSIDAD DE TARAPACÁ



## FACULTAD DE INGENIERÍA



### DEPARTAMENTO DE INGENIERÍA EN COMPUTACIÓN E INFORMÁTICA



#### Plan de Proyecto:

Solución Domótica Inclusiva: Comunicación y Monitoreo para  
Personas con Capacidades Diferentes mediante Raspberry Pi  
**COSMIC**

Alumnos: Karen Correa.  
Cristina Cortez.  
Jhosep Marca.  
Fernanda Ventura.

Asignatura: Proyecto II.

Profesor: Diego Aracena.

02 DE DICIEMBRE - 2023  
ARICA - CHILE

# 1. Tabla de Contenidos

<b>1. Tabla de Contenidos</b>	<b>2</b>
<b>2. Resumen</b>	<b>4</b>
<b>3. Índice de tablas</b>	<b>6</b>
<b>4. Índice de figuras</b>	<b>7</b>
<b>5. Introducción</b>	<b>8</b>
<b>6. Planteamiento del problema</b>	<b>9</b>
<b>6.1. Identificación del problema</b>	<b>9</b>
6.2. Solución del problema	9
<b>7. Panorama general</b>	<b>10</b>
7.1. Resumen del proyecto	10
7.1.1. Propósito	10
7.1.2. Alcance	10
7.1.3. Objetivos	10
7.1.4. Suposiciones y restricciones	11
7.1.5. Entregables.	11
<b>8. Historial de versiones</b>	<b>12</b>
<b>9. Organización del proyecto</b>	<b>13</b>
9.1. Personal y entidades internas	13
9.2. Roles y responsabilidades	14
9.3. Mecanismos de comunicación	15
<b>10. Planificación de los procesos de gestión</b>	<b>17</b>
10.1. Planificación inicial del proyecto	17
10.1.1. Planificación de estimaciones	17
10.1.2. Planificación de Recursos Humanos	18
10.2. Lista de actividades	19
10.2.1. Actividades de trabajo	19
10.2.2. Actividades de tiempo	20
10.3. Planificación de gestión de riesgos	21
10.4. Aspectos éticos	23
<b>11. Planificación de los procesos técnicos</b>	<b>24</b>
11.1. Modelo de proceso	25
11.1.1. Diagrama de caso de uso general	25
11.1.2. Casos de uso y sus diagramas	26
11.1.3. Diagrama de clases	31
11.1.4. Diagrama de secuencias	31
11.1.4.1. Nivel 1	31
11.1.4.2. Nivel 2	32
11.2. Arquitectura de comunicación con respecto a diseños	33
11.3. Diseño de interfaz de usuario	34
11.4. Especificación de requerimientos	36
<b>11.5. Herramientas y técnicas</b>	<b>37</b>
<b>12. Planificación de procesos de soporte</b>	<b>38</b>
12.1. Plan de integración	38

12.2. Modelo de Implementación:	38
12.3. Módulos Implementados:	38
12.3.1. Módulo de monitoreo (generar imagen):	38
12.3.2. Módulo de monitoreo (capturar imagen):	40
12.3.3. Módulo de detección de movimiento:	41
12.3.4. Módulo de generación de alerta de movimiento:	41
13. Planificación de la documentación	42
<b>14. Conclusión</b>	<b>43</b>
<b>15. Trabajos futuros</b>	<b>44</b>
<b>16. Referencias</b>	<b>45</b>

## **2. Resumen**

El presente informe muestra todo el desarrollo del proyecto COSMIC, un proyecto de domótica con un sistema de comunicación y monitoreo del hogar. Este proyecto está enfocado principalmente en personas con situación de dependencia -pueden ser, por ejemplo, adultos mayores, personas en situación de discapacidad ya sea mental o física, entre otras situaciones- en donde, el enfoque principal, es facilitar la relación entre la persona en situación de dependencia y su cuidador. Se hará implementando un sistema de comunicación integral, fácil de usar, eficiente y adaptable, donde la persona en situación de dependencia podrá ingresar un mensaje -ya sea de voz o de texto- y se reproduzca por los distintos altavoces instalados en el hogar. Paralelamente, el cuidador podrá tener acceso a un sistema de monitoreo en tiempo real, donde podrá observar a la persona asistida en todo momento, para así facilitar y fomentar que el cuidador haga otras tareas en vez de monitorear presencialmente a la persona en situación de dependencia.

En base a todo lo anterior, la denominación "COSMIC" surge de la combinación de las iniciales de seis elementos clave que definen su funcionalidad: Conexión, Observación, Seguridad, Monitoreo, Inteligencia y Comunicación en el Hogar. Esta designación refleja la esencia integral del proyecto, que se centra en mejorar la calidad de vida de las personas con discapacidad y proporcionar tranquilidad a sus cuidadores.

Para poder realizar este proyecto, se hará uso de una computadora SBC compacta llamada Raspberry Pi, modelo 4. Ésta será el corazón del funcionamiento de la estructura del proyecto, además de otros sensores como cámaras, altavoces y micrófonos.

El grupo encargado de realizar este proyecto está conformado por cuatro estudiantes de la carrera Ingeniería civil en computación e informática, de la Universidad de Tarapacá. Estas personas fueron juntadas estratégicamente por las distintas habilidades que poseen cada uno, las cuales se complementan y tiene como resultado un grupo completo en las distintas áreas del desarrollo de este proyecto.

La motivación que se tiene como grupo para la realización del proyecto es que la forma de vida de las personas en situación de dependencia y sus cuidadores mejoren su calidad de vida, facilitando, con este proyecto, el problema de la comunicación entre estos dos individuos debido a las limitaciones con las que viven. De esta manera, el cuidador podrá dedicarse a distintas tareas y el asistido podrá tener más tranquilidad que podrá comunicarse con su cuidador en cualquier momento ante cualquier necesidad que presente.

Finalmente, este proyecto tendría un gran impacto, en caso de realizarse correctamente, pues tanto los usuarios como el asistido y el cuidador, como los estudiantes que están desarrollando este proyecto. Éste, facilita un poco la forma de

vida de los usuarios y ayuda a desarrollar la experiencia en este tipo de proyectos. Sin mencionar, que se aprende de los distintos implementos empleados en el proyecto; se aprendió a trabajar en equipo, identificando y fomentando las habilidades y fortalezas de cada integrante del equipo, además de aprender a organizarnos dependiendo de cada tipo de persona que representa cada integrante.

En la fase del proyecto actual sólo se ha desarrollado una planificación previa a la implementación del proyecto.

### **3. Índice de tablas**

Tabla 1 Historial de versiones del informe	2
Tabla 2 Asignación y designación de los roles del proyecto	9
Tabla 3 Desglose de recursos necesarios para el proyecto	12
Tabla 4 Desglose de los softwares necesarios para el proyecto	12
Tabla 5 Desglose de roles y costos por rol de cada integrante del proyecto	13
Tabla 6 Numeración de riesgos, su probabilidad y nivel de impacto en el proyecto	15
Tabla 7 Acciones remediales para cada riesgo encontrado	16

#### **4. Índice de figuras**

Figura 1 Ilustración de la Carta Gantt del proyecto.

14

## **5. Introducción**

De acuerdo a una encuesta nacional respecto de situaciones de discapacidad, dependencia y cuidados en la población general, el 17,6% de las personas adultas del país presenta algún grado de discapacidad, lo que equivale a 2.703.893 personas. De ellas, un 11,4% presenta discapacidad severa y un 6,2% presenta discapacidad leve o moderada.

Según lo anterior, se encontró una solución parcial a la problemática mediante un proyecto el cual presenta un sistema domótico que busca proporcionar seguridad y eficiencia en el hogar mediante la implementación de una tarjeta Raspberry-Pi, donde la persona dependiente podrá comunicarse en cualquier momento con su cuidador, y el cuidador podrá monitorear constantemente a la persona dependiente. Como propósito se pretende no solo facilitar la comunicación sino también permitir un monitoreo constante para mejorar la calidad de vida de las personas con capacidades distintas y proporcionar tranquilidad a sus cuidadores.

Este informe proporciona un panorama detallado de la planificación y el desarrollo del proyecto, lo cual permitirá mantener una visión fija de la finalidad de este trabajo, y así progresar correctamente para poder brindar una solución efectiva y tecnológica a la problemática planteada.

## **6. Planteamiento del problema**

### **6.1. Identificación del problema**

Las personas con capacidades distintas, en mayor parte, son personas dependientes, por lo que necesitan de un cuidador. Existe una problemática en cuanto a la comunicación entre la persona dependiente y el cuidador, restringiendo así la comunicación directa debido a que el cuidador debe hacer otro tipo de tareas aparte de estar con la persona dependiente. Además, al estar haciendo otras tareas, el cuidador no puede estar pendiente todo el tiempo del asistido, por lo que existen momentos de descuido donde pueden ocurrir problemas debido a la falta de atención.

### **6.2. Solución del problema**

Nuestro equipo encontró esta problemática y, luego de una lluvia de ideas, se llegó a la idea de crear una aplicación donde se pueda tener una comunicación más directa entre el cuidador y el asistido, usando una cámara para tener un monitoreo constante de la persona asistida y con parlantes para que el asistido pueda comunicarse en cualquier momento con el cuidador donde sea que esté.

## **7. Panorama general**

### **7.1. Resumen del proyecto**

#### **7.1.1. Propósito**

El proyecto COSMIC permitirá desarrollar de mejor manera la comunicación entre las personas en situación de dependencia y sus cuidadores, mediante parlantes ubicados en el hogar; además, los cuidadores podrán monitorear constantemente a la persona asistida mediante una cámara ubicada estratégicamente para que tenga la mejor vista posible.

#### **7.1.2. Alcance**

El proyecto cuenta con módulos con vistas distintas dependiendo del tipo de usuario que esté utilizando la aplicación móvil, permitiendo monitorear o comunicarse entre personas ubicadas en el mismo hogar. Este software estará codificado en lenguaje Python y permitirá desarrollar las funcionalidades mediante una tarjeta Raspberry Pi y sensores como altavoces, cámaras, entre otros.

#### **7.1.3. Objetivos**

##### **a) Objetivo General**

Crear un sistema domótico que permita mantener un monitoreo constante y una comunicación fluida, eficaz y accesible entre personas en situación de discapacidad y/o en situación de dependencia con sus cuidadores que se encuentren en el mismo hogar.

##### **b) Objetivos Específicos**

- Estudiar e investigar sobre los problemas de comunicación que afrontan las personas con capacidades distintas con sus cuidadores.
- Instalar y configurar la tarjeta Raspberry Pi para que los altavoces y la cámara funcionen correctamente.
- Diseñar y asegurarse de que la interfaz de la aplicación móvil sea fácil de usar, con características de accesibilidad que se adapten a las distintas necesidades.
- Realizar pruebas de usabilidad para que la aplicación móvil funcione correctamente, poniéndose en la situación donde una persona en situación de dependencia y el cuidador la usen.
- Documentar todo el proceso de desarrollo del proyecto.

#### 7.1.4. Suposiciones y restricciones

##### a) Suposiciones

- El modelo 3D plantea correctamente los escenarios en los que se utilizará el proyecto.
- Las actividades a realizar se llevarán a cabo en los plazos establecidos previamente, de acuerdo a la carta gantt.
- Los sensores son adecuados y compatibles con el objetivo del proyecto.
- Existe una conexión wi-fi estable.
- Existe al menos un cuidador y una persona en situación de discapacidad viviendo bajo el mismo techo.
- Los implementos como Raspberry Pi, sensores, entre otros, se encuentran disponibles y en buen estado para su correcto funcionamiento.

##### b) Restricciones

- Orientado a personas con distintas capacidades y/o en situación de dependencia.
- Trabajar con Raspberry-Pi.
- El proyecto debe efectuarse para el contexto de un hogar.
- El proyecto debe ser orientado al funcionamiento con smartphones Android.

#### 7.1.5. Entregables.

- **Maqueta del proyecto:** En ella se modela de forma 3D la instalación y los escenarios contemplados para el proyecto, representa cómo se organizará físicamente el proyecto en su entorno final.
- **Informes del Proyecto:** Estos informes contienen detalles sobre los objetivos, resultados, desafíos y conclusiones de cada fase (primera fase, segunda fase y la fase final), ayudan a tener una visión completa del avance del proyecto.
- **Presentaciones:** Permiten presentar el avance de los proyectos a los demás compañeros, y así transmitir de forma clara y concisa la información relevante de la evolución del proyecto.
- **Software COSMIC:** Es la aplicación a través de la cual los usuarios pueden tener acceso a las funciones del proyecto.
- **Bitácoras Semanales:** Registran el progreso semanal del proyecto y define las actividades a realizar en las próximas sesiones de trabajo.
- **Wiki:** Contiene datos importantes sobre el proyecto. Además sirve como una fuente centralizada de información a la que los interesados en el proyecto pueden consultar.

- **Manual de Usuario:** Contiene las instrucciones de uso para que cualquier usuario pueda utilizar correctamente el proyecto.
- **Carta Gantt:** Contiene la planificación de todas las actividades del proyecto y las fechas importantes.
- **Descripción de la arquitectura con respecto a los modelos:** Son las conexiones físicas, eléctricas o inalámbricas que se establecen entre los sensores utilizados en el proyecto y la Raspberry Pi.
- **Diseño de interfaz de usuario:** es una imagen referencial de la interfaz que usará el usuario, incluyendo las distintas vistas dependiendo del tipo de usuario.
- **Especificación de requerimientos:** listado de los distintos tipos de requerimientos que presenta el proyecto.
- **Modelo de diseño:** descripción de los casos de uso que representan el proyecto actual y los diagramas de clases.
- **Modelo de interacción:** representación y descripción de diagramas de secuencia.

## 8. Historial de versiones

Fecha	Versión	Descripción	Autor(es)
22-09-2023	1.0	Se redactan los puntos de Panorama General, Referencias, Organización del Proyecto y hasta el punto 4.3 de la Planificación de los Procesos de Gestión.	Karen Correa, Cristina Cortez, Jhosep Marca, Fernanda Ventura.
	2.0	Se corrigen los errores y correcciones señalados en la versión anterior por el docente. Además, se agrega el punto de planificación de procesos técnicos.	Karen Correa, Cristina Cortez, Jhosep Marca, Fernanda Ventura.

Tabla 1 Historial de versiones del informe

## 9. Organización del proyecto

### 9.1. Personal y entidades internas

A continuación, se presenta el personal que participa en el proyecto, junto con las entidades internas que representan dentro de la organización del proyecto COSMIC.

#### a) Personal en el Proyecto

- Jefe de Proyecto:
  - Jhosep Marca.
- Diseñador de Experiencia de Usuario (UX):
  - Fernanda Ventura.
- Desarrolladores de Software:
  - Karen Correa.
  - Jhosep Marca.
  - Cristina Cortez.
  - Fernanda Ventura.
- Documentalistas de Proyecto:
  - Cristina Cortez.
  - Jhosep Marca.
  - Karen Correa.
  - Fernanda Ventura.
- Técnico de Configuración de Hardware:
  - Jhosep Marca.

#### b) Entidades Internas en el Proyecto

- **Líder y Coordinador del Proyecto:** Jhosep Marca.
- **Responsable del Diseño de Interfaz y Experiencia de Usuario:** Fernanda Ventura.
- **Equipo de Desarrollo de Software:** Karen Correa (Encargada de Desarrollo de Software), Jhosep Marca, Cristina Cortez, Fernanda Ventura.
- **Equipo de Documentación de Proyecto:** Cristina Cortez (Encargada de Documentación de Proyecto), Jhosep Marca, Karen Correa, Fernanda Ventura.
- **Encargado de Configuración de Hardware:** Jhosep Marca.

## 9.2. Roles y responsabilidades

Para la realización del proyecto se asignó a cada integrante distintos roles y responsabilidades:

Rol	Función	Responsabilidad
<b>Jefe de proyecto</b>	Encargado de supervisar el progreso del proyecto, tomar decisiones estratégicas, gestionar el presupuesto y los recursos, y garantizar que se cumplan los plazos.	<ul style="list-style-type: none"> <li>- Responsable de monitorear y evaluar constantemente el progreso del proyecto para asegurarse de que se estén alcanzando los objetivos y cumpliendo los plazos.</li> <li>- Administrar el presupuesto del proyecto, asegurándose de que los gastos se mantengan dentro de los límites establecidos y que se utilicen de manera eficiente.</li> </ul>
<b>Diseñador de Experiencia de Usuario (UX)</b>	Encargado de crear la interfaz de usuario de la aplicación móvil, asegurándose de que sea accesible y atractiva.	<ul style="list-style-type: none"> <li>- Responsable de trabajar estrechamente con el Desarrollador de Software para asegurarse de que el diseño se implemente correctamente y que la interfaz funcione según lo previsto.</li> </ul>
<b>Desarrollador de Software</b>	Encargado de escribir, desarrollar y mantener el código de la aplicación móvil.	<ul style="list-style-type: none"> <li>- Responsable de escribir código eficiente y funcional.</li> <li>- Identificar y solucionar errores y problemas en el código. Realizando pruebas para asegurarse de que el software funcione correctamente.</li> <li>- Mejora el rendimiento de las aplicaciones y sistemas, optimizando el código.</li> </ul>
<b>Documentalista de</b>	Encargado de registrar	<ul style="list-style-type: none"> <li>- Responsable de mantener</li> </ul>

<b>Proyecto</b>	y documentar meticulosamente el progreso del proyecto, incluyendo tareas realizadas y decisiones tomadas. Además, mantiene los informes actualizados y disponibles para el equipo de proyecto y las partes interesadas.	bitácoras detalladas y cronológicas del proyecto, registrando actividades diarias, reuniones, comunicaciones y cualquier evento relevante.  - Colaborar con los miembros del equipo para recopilar información y asegurarse de que todas las actividades estén debidamente registradas.  - Responsable de redactar, editar y presentar documentos de manera clara y precisa.
<b>Técnico de Configuración de Hardware</b>	Encargado de configurar y operar el hardware, incluyendo sensores y dispositivos como la Raspberry Pi. Este rol implicaría la instalación, conexión, configuración de componentes hardware, asegurando que funcionen correctamente y se integren adecuadamente en el sistema.	- Configurar y operar el hardware, incluyendo sensores y dispositivos como Raspberry Pi.  - Realizar la instalación, conexión y configuración de componentes hardware.  - Asegurar el correcto funcionamiento de todos los componentes hardware.

Tabla 2 Asignación y designación de los roles del proyecto

### 9.3. Mecanismos de comunicación

El proyecto utiliza varios mecanismos de comunicación para facilitar la colaboración en equipo. Estas herramientas desempeñan un papel importante en la gestión y la coordinación de las actividades del proyecto. A continuación, se describen los principales mecanismos de comunicación y sus respectivos horarios de uso:

- **Discord:** Discord es una plataforma de comunicación en línea que nos ofrece una variedad de funciones, incluyendo chat de texto y voz, lo que lo convierte en una herramienta versátil para nuestras reuniones en línea.
- **Horario de Uso:**
  - **Días:** Jueves y sábado.
  - **Horario:** De 20.00 a 22.00 hrs.
- **WhatsApp:** Permite una comunicación rápida a través de mensajes de texto, voz y multimedia. Es adecuado para coordinar tareas y resolver problemas de manera ágil. También, se puede utilizar para enviar recordatorios y notificaciones importantes relacionadas con el proyecto, como plazos de entrega o fechas de reuniones.
- **Horario de Uso:** Constante.

De esta forma, queda claro que WhatsApp se utiliza de manera continua, sin restricciones de días u horas específicas.

- **Correo electrónico:** El correo electrónico es un sistema de comunicación electrónica que nos permite enviar y recibir mensajes y archivos a través de Internet.
- **Horario de Uso:** Para casos especiales.

Se utiliza únicamente en situaciones específicas cuando sea necesario, sin un horario fijo o regular.

- **Google Drive:** Permite almacenar, sincronizar y compartir archivos y documentos en línea. También ofrece herramientas de colaboración para crear y editar documentos en tiempo real.
- **Horario de Uso:** Durante las sesiones de trabajo.
- **Redmine:** Redmine es una plataforma de gestión de proyectos y seguimiento. Permite herramientas para planificar, colaborar y llevar un registro de tareas y proyectos en un entorno en línea.
- **Horario de Uso:** Durante las sesiones de trabajo.

De esta forma, queda claro que Google Drive y Redmine se utilizan exclusivamente durante las sesiones de trabajo.

## 10. Planificación de los procesos de gestión

### 10.1. Planificación inicial del proyecto

#### 10.1.1. Planificación de estimaciones

Recursos			
Nombre	Precio Unitario	Cantidad	Precio Total
Mini Cámara Espía A9 HD WiFi	\$6.000.-	1	\$6.000.-
Kit Audio Completo 2X10W Bluetooth USB SD MP3 RADIO Control R Fonestar.	\$150.000.-	2	\$300.000.-
Microfono Mini Alta Calidad Clip Profesional Unidireccional	\$4.990.-	1	\$4.990.-
Tarjeta Raspberry-pi modelo B 8GB.	\$85.000.-	4	\$340.000.-
<b>Total</b>	<b>\$395.990.-</b>		

Tabla 3 Desglose de recursos necesarios para el proyecto

Software	
Nombre	Costo
IDE Visual Studio Code.	\$0.-
Python	\$0.-
Raspberry Pi OS.	\$0.-
WhatsApp.	\$0.-
Discord.	\$0.-
Servicios de Google.	\$0.-

Plataforma Redmine.	\$0.-
<b>Total</b>	<b>\$0.-</b>

Tabla 4 Desglose de los softwares necesarios para el proyecto

### 10.1.2. Planificación de Recursos Humanos

Considerando la siguiente información:

- El jefe de proyecto dedicará 7 horas por semana durante 4 meses, lo que resulta en un total de 112 horas en el proyecto.
- El Desarrollador de Software trabajará 10 horas por semana durante un total de 4 meses y 2 semanas, lo que suma un total de 60 horas en el proyecto.
- El Diseñador de Experiencia de Usuario (UX) dedicará 10 horas por semana durante 3 semanas, totalizando 30 horas en el proyecto.
- El Documentalista de Proyecto trabajará 4 horas por semana durante un total de 4 meses, sumando un total de 64 horas en el proyecto.
- El Técnico de Configuración de Hardware se encargará de la configuración y operación del hardware durante 2 meses, trabajando 2 horas por semana y sumando un total de 16 horas en el proyecto.

En base a lo anterior, cada encargado principal deberá cumplir con las horas asignadas, en cambio a los roles secundarios, que cumplirán menos horas de las asignadas.

A continuación se muestra de forma organizada los roles de cada integrante del proyecto, su rol principal destacado en rojo y sus roles secundarios en negro. Se muestran los valores por hora de cada rol y, según las horas de trabajo, un costo total por rol.

Nombre del trabajador	Roles	Valor por hora	Horas trabajado	Costo total por rol
Fernanda Ventura	Desarrollador de software	\$7.385	30	\$220.740
	Diseñador de Experiencia de Usuario	\$3.692	30	\$110.760
	Documentalista de Proyecto	\$3.692	30	\$110.760
Cristina Cortez	Desarrollador de software	\$7.385	60	\$443.100
	Documentalista de Proyecto	\$3.692	64	\$236.288
Jhosep Marca	Desarrollador de software	\$7.385	30	\$221.550
	Jefe de proyecto	\$9.231	112	\$1.033.872
	Técnico de Configuración de Hardware	\$2.338	16	\$37.408
	Documentalista de Proyecto	\$3.692	25	\$92.300
Karen Correa	Desarrollador de software	\$7.385	60	\$443.100
	Documentalista de Proyecto	\$3.692	45	\$166.140
<b>Total</b>				<b>\$3.116.018</b>

Tabla 5 Desglose de roles y costos por rol de cada integrante del proyecto

Estimación de costos totales del proyecto, incluyendo recursos, softwares y horas de trabajo: **\$3.512.008.-**

## 10.2. Lista de actividades

### 10.2.1. Actividades de trabajo

- Planificación del proyecto.
  - Conformación de grupos y asignación de roles.
  - Planeación idea borrador del proyecto.
  - Presentación esbozo de la idea.
  - Determinación de la idea y nombre del proyecto.
  - Formulación y construcción de la maqueta.
  - Redacción y entrega del informe 1.

- Ejecución del proyecto:
  - Estudio características Raspberry-Pi.
  - Justificación de la arquitectura de comunicación.
  - Redacción y entrega informe 2.
  - Justificación del aplicativo para la comunicación del dispositivo móvil.
  - Programación de la interfaz.
  
- Cierre del proyecto:
  - Pruebas.
  - Redacción y entrega del informe final.

### 10.2.2. Actividades de tiempo

Para la planificación de las actividades de tiempo se realizó una carta gantt:

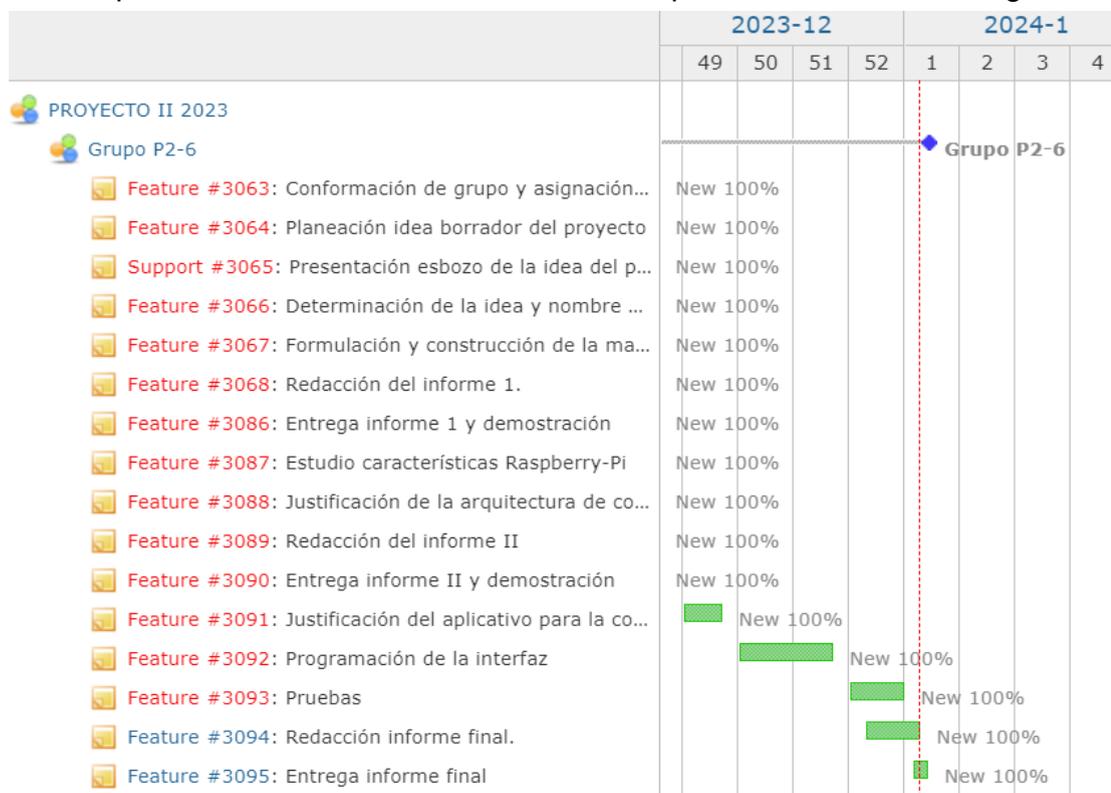


Figura 1 Ilustración de la Carta Gantt del proyecto

- Planificación del proyecto: 7 semanas.
  - Conformación de grupos y asignación de roles: 2 días.
  - Planeación idea borrador del proyecto: 3 días.
  - Presentación esbozo de la idea: 1 día.
  - Determinación de la idea y nombre del proyecto: 3 días.
  - Formulación y construcción de la maqueta: 10 días.
  - Redacción y entrega del informe 1: 15 días.

- Desarrollo del proyecto: 5 semanas.
  - Estudio características Raspberry-Pi: 10 días.
  - Justificación de la arquitectura de comunicación: 2 días.
  - Redacción y entrega informe 2: 10 días.
  - Justificación del aplicativo para la comunicación del dispositivo móvil: 11 días.
  - Programación de la interfaz: 5 días.
- Finalización del proyecto: 2 semanas.
  - Pruebas: 1 día.
  - Redacción y entrega del informe final: 2 días.

### 10.3. Planificación de gestión de riesgos

Durante la ejecución del proyecto, pueden surgir diversas dificultades que potencialmente afecten su desarrollo. Por lo tanto, es fundamental considerar los posibles riesgos junto con sus respectivas soluciones, con el objetivo de prevenir problemas. A continuación, se presenta una tabla que contiene esta información, además de un nivel de dificultad que representa el nivel de impacto que puede tener un riesgo, este nivel se clasifica en una escala del 1 al 5, con los siguientes niveles de dificultad:

- 1) Mínimo.
- 2) Bajo.
- 3) Moderado.
- 4) Alto.
- 5) Crítico.

N°	Riesgo	Probabilidad de ocurrencia	Nivel de impacto
1	Perder el progreso del Raspberry Pi	20%	5
2	No tener dispositivos de trabajo	10%	5
3	Cambio de requisitos por parte del cliente	60%	4
4	Falta de formación respecto a las herramientas	80%	4
5	Cambio de personal	10%	4
6	Falta de disposición de los integrantes	20%	3
7	Suspensión de clases	50%	3

<b>8</b>	Perder información de los informes	40%	3
<b>9</b>	Indisposición de los integrantes	40%	3
<b>10</b>	Exceder los gastos establecidos	50%	3
<b>11</b>	Herramientas defectuosas	20%	2
<b>12</b>	No cumplir con las fechas establecidas	65%	2
<b>13</b>	Falta de organización	40%	2

**Tabla 6** Numeración de riesgos, su probabilidad y nivel de impacto en el proyecto

A continuación, se expondrá una serie de acciones remediales para cada riesgo explicado anteriormente.

<b>Riesgo</b>	<b>Acciones remediales</b>
Perder el progreso del Raspberry Pi.	Se le dará total prioridad a desarrollar nuevamente el progreso de la Raspberry Pi.
No tener dispositivos de trabajo.	Comunicarse con el cliente para adquirir dispositivos de trabajo.
Cambio de requisitos por parte del cliente.	Llegar a un acuerdo con el cliente y hacer constantemente consultas al cliente, de modo que tenga conocimientos de los avances.
Falta de formación respecto a las herramientas.	Adquirir información en equipo, para avanzar de forma conjunta y progresiva.
Cambio de personal.	El equipo redistribuye el trabajo faltante del personal faltante, de modo que no retrase el proyecto.
Falta de disposición de los integrantes.	Hacer un llamado de atención a los integrantes que tengan poca disposición con el proyecto.
Suspensión de las sesiones de trabajos.	Realizar reuniones fuera del horario del trabajo.
Perder el contenido de los informes.	El equipo se enfocará en desarrollar nuevamente el contenido de los informes hasta el punto perdido, para que no retrase el desarrollo del proyecto.
Indisposición de los	Repartir el trabajo del integrante al personal, de modo

integrantes.	que no retrase el desarrollo del proyecto.
Exceder los gastos establecidos.	Dar aviso al cliente del posible aumento de gastos del proyecto para coordinar un nuevo acuerdo de gastos o cambiar las herramientas del proyecto.

Tabla 7 Acciones remediales para cada riesgo encontrado

#### 10.4. Aspectos éticos

- **Accesibilidad y Diseño Universal:** Asegurar que la interfaz y las funcionalidades del sistema sean accesibles para usuarios con diversas capacidades, garantizando que todos puedan beneficiarse de la tecnología sin importar sus habilidades físicas o cognitivas.
- **Consentimiento Informado sobre Funcionalidades:** Informar claramente a los usuarios y cuidadores sobre las funcionalidades del sistema, cómo se utilizarán y qué impacto tendrán en su experiencia diaria.
- **Seguridad de la Comunicación:** Aunque no se recopilen datos sensibles, asegurarse de que las comunicaciones entre el usuario y el sistema, así como entre el usuario y el cuidador, estén protegidas para garantizar la privacidad y la seguridad.
- **Transparencia y Claridad en el Uso del Sistema:** Proporcionar información transparente sobre cómo el sistema opera y cómo puede beneficiar a las personas con discapacidad y a sus cuidadores.
- **Soporte y Capacitación:** Ofrecer soporte y capacitación adecuados para los usuarios y cuidadores, asegurando que comprendan completamente cómo utilizar el sistema y aprovechar al máximo sus capacidades.

## 11. Planificación de los procesos técnicos

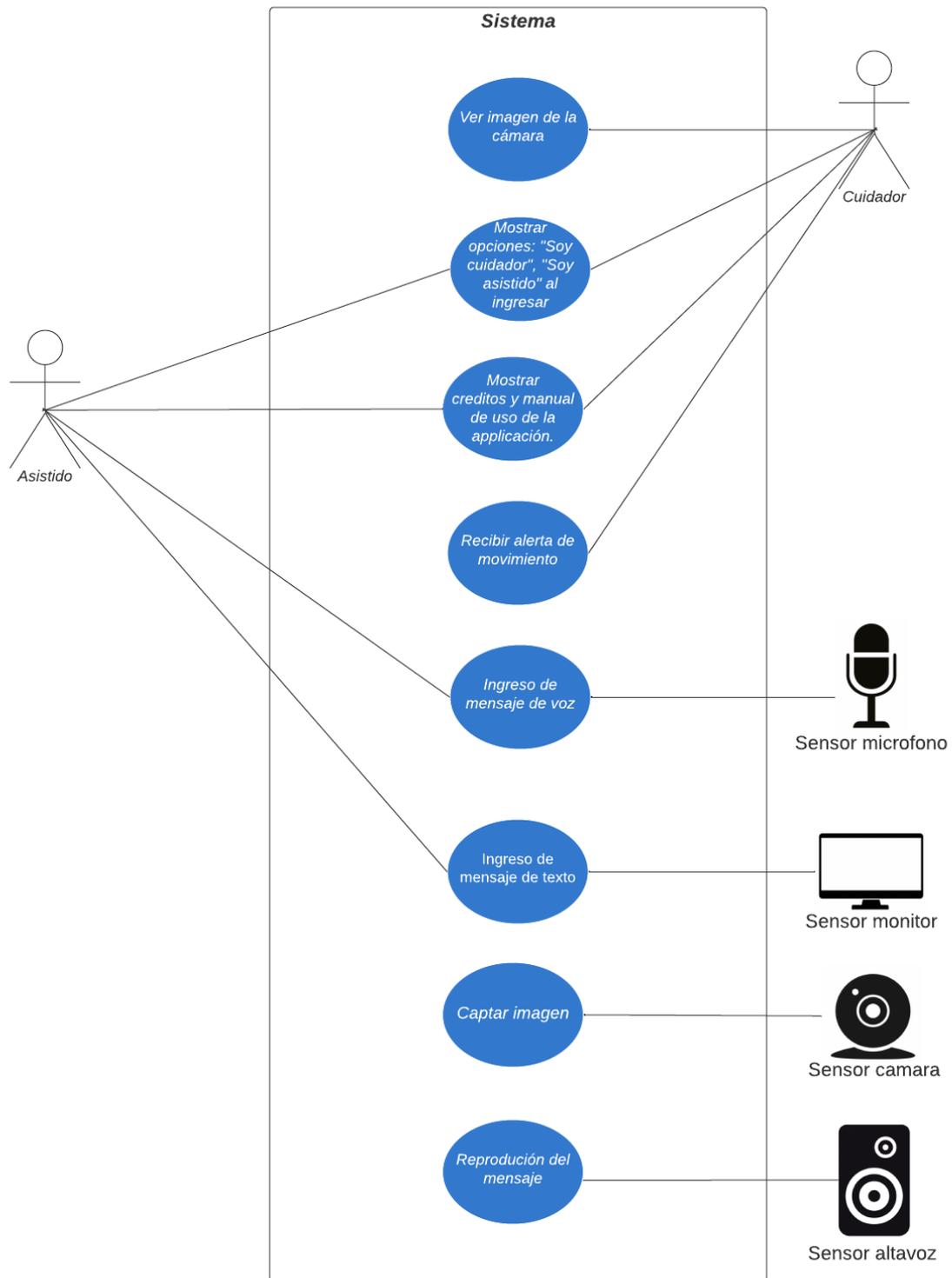
Para una mejor visualización del funcionamiento e interacción entre la aplicación del proyecto a desarrollar y los distintos actores involucrados en el mismo, se realizan una serie de casos de uso y sus respectivos diagramas, para, de esta manera, describir y graficar las distintas funcionalidades que tendrá la aplicación móvil y su interacción con los actores y sensores.

Los actores involucrados en el proyecto son los siguientes:

- a) **Usuario “asistido”**: este actor es la persona que se encuentra en situación de discapacidad, de dependencia o con capacidades diferentes, la cual necesita de un cuidador para poder desenvolverse. El usuario “asistido” tiene un acceso limitado a la aplicación, pudiendo solamente ingresar información, ya sea por mensaje de texto o mensaje de voz, el cual será transmitido por los parlantes previamente instalados y conectados por la casa.
- b) **Usuario “cuidador”**: este actor es la persona que ayuda a la persona asistida en las tareas cotidianas, por lo que necesita tener su atención casi en un 100% en ese actor. El usuario “cuidador” también tiene un acceso limitado de la aplicación, pudiendo solamente tener una vista de la cámara instalada en la casa y posicionada en la habitación de la persona asistida.
- c) **Parlantes**: es una máquina traductora electroacústica la cual se encarga de transmitir el mensaje del asistido por toda la habitación para que el cuidador se entere de lo que quiere o necesita el asistido. Deben existir 3 parlantes alrededor de la casa, colocados estratégicamente para que el mensaje se transmita de la mejor forma posible.
- d) **Micrófono**: dispositivo de entrada encargado de recepcionar el mensaje del asistido para después, transmitirlo por los parlantes a través de la casa. Este dispositivo debe estar conectado previamente y ubicado de tal manera que el asistido pueda usarlo de forma sencilla.
- e) **Cámara**: aparato cuya funcionalidad es registrar imágenes de la habitación donde se encuentra el asistido para que, de esta forma, el cuidador pueda tener una imagen en vivo y en directo de todo lo que está sucediendo en dicha habitación. Este aparato debe estar colocado estratégicamente en la habitación para que en la imagen se pueda ver claramente al asistido y se tenga una vista lo más panorámica posible para que no se pierda ningún punto de la habitación.
- f) **Monitor**: dispositivo que consiste en una pantalla en la cual el usuario inserta información y da órdenes. Este dispositivo lo utiliza el asistido para ingresar el mensaje de texto que desea transmitir y, a través de un traspaso texto-audio, los parlantes transmiten el mensaje con una voz artificial.

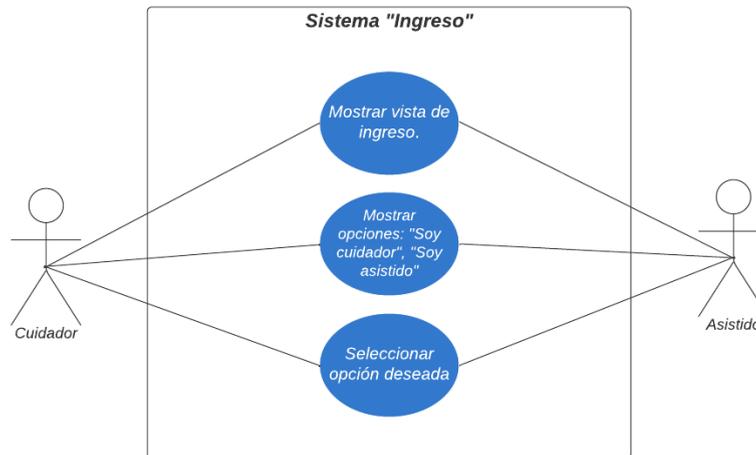
### 11.1. Modelo de proceso

#### 11.1.1. Diagrama de caso de uso general



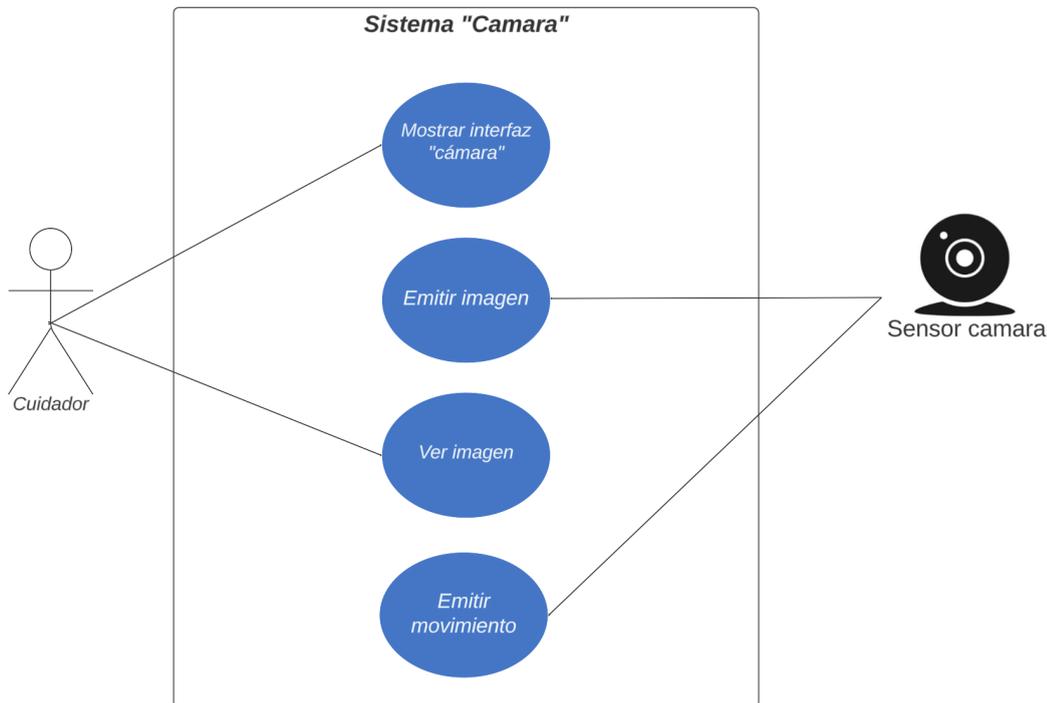
### 11.1.2. Casos de uso y sus diagramas

a) Caso de uso: Mostrar opciones de ingreso.



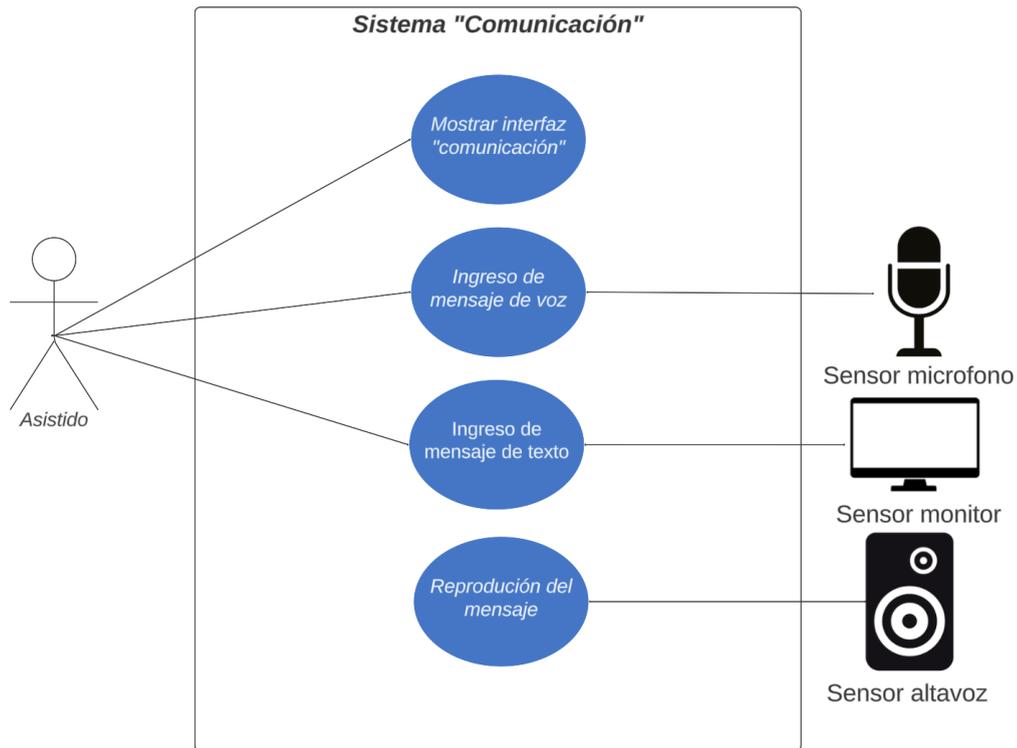
Caso de uso:	Mostrar opciones de ingreso
Descripción:	Se muestra una vista principal para que el usuario ingrese a la aplicación con uno de los 2 roles definidos.
Actor:	Usuario.
Precondición:	Se debe tener una conexión establecida a la red wi-fi. El usuario debe tener cargada la aplicación del sistema en su dispositivo.
Flujo principal:	1.- Se muestran dos opciones para ingresar a la aplicación: "Soy cuidador", "Soy asistido". 2.- El usuario interactúa con el sistema seleccionando la opción deseada. 3.- Se muestra la vista principal dependiendo del rol de usuario seleccionado.
Flujo Alternativo:	
Postcondiciones:	El sistema queda configurado para mostrar la vista correspondiente a su rol.

b) Caso de uso: mostrar interfaz “Cámara”.



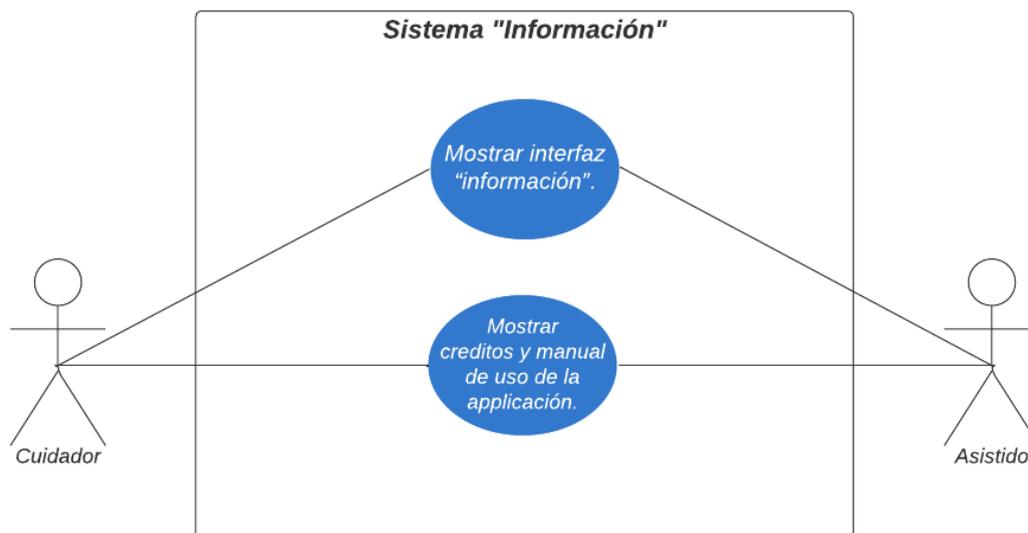
Caso de uso:	Mostrar interfaz “Cámara”.
Descripción:	Se muestra un apartado donde el cliente puede ver la vista de la cámara de la habitación donde se encuentra la persona asistida.
Actor:	Cuidador, Cámara.
Precondición:	La aplicación requiere una conexión estable entre la cámara y la aplicación. Los usuarios deben tener roles designados, ya sea como 'cuidador' o 'asistido'.
Flujo principal:	1.- <<Incluye>> caso de uso “Ingreso”. 2.- El sistema muestra la vista de la imagen de la cámara en grande, junto con el botón “Información” y “Salir”. 3.- El usuario presiona el botón “Salir”. 4.- El sistema vuelve a la interfaz “opciones de Ingreso”.
Flujo Alternativo:	3.1.- Presiona el botón “Información”. 4.1.- Muestra interfaz “Información”.
Postcondiciones:	El cuidador puede ver una vista de la cámara instalada en su residencia.

c) Caso de uso: mostrar interfaz “Comunicación”.



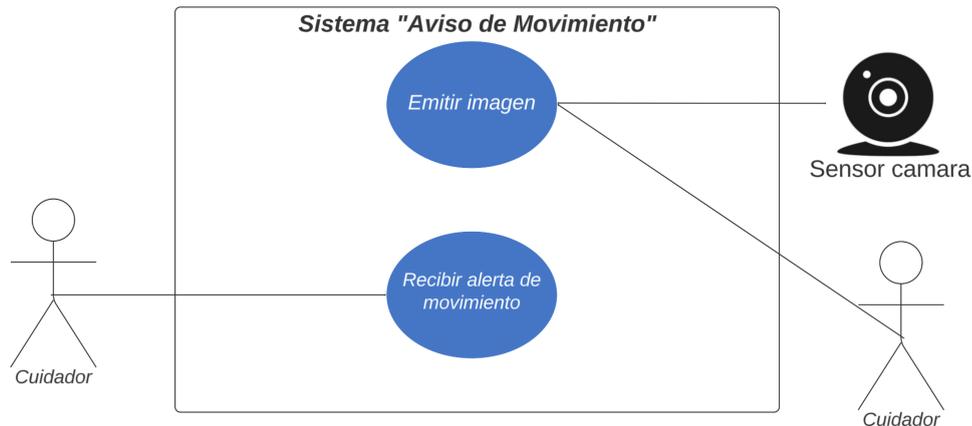
Caso de uso:	Mostrar interfaz “Comunicación”.
Descripción:	Se muestra un apartado donde se encuentra vista principal
Actor:	Asistido, Micrófono, Monitor, Altavoz.
Precondición:	Se debe tener una conexión establecida entre la aplicación y el parlante.
Flujo principal:	<ol style="list-style-type: none"> <li>1.- &lt;&lt;Incluye&gt;&gt; caso de uso “Ingreso”.</li> <li>2.- Selecciona la opción de “Pulsar para hablar” para reproducir el mensaje por el altavoz.</li> <li>3.- Ingresa mensaje.</li> <li>4.- Se reproduce el mensaje de voz por todos los altavoces.</li> </ol>
Flujo Alternativo:	2.1.- Selecciona la opción de “Escribir para hablar” para reproducir el mensaje por el altavoz.
Postcondiciones:	El cliente puede comunicarse con su cuidador desde la aplicación, sin tener que moverse de su ubicación.

d) Caso de uso: mostrar interfaz "Información".



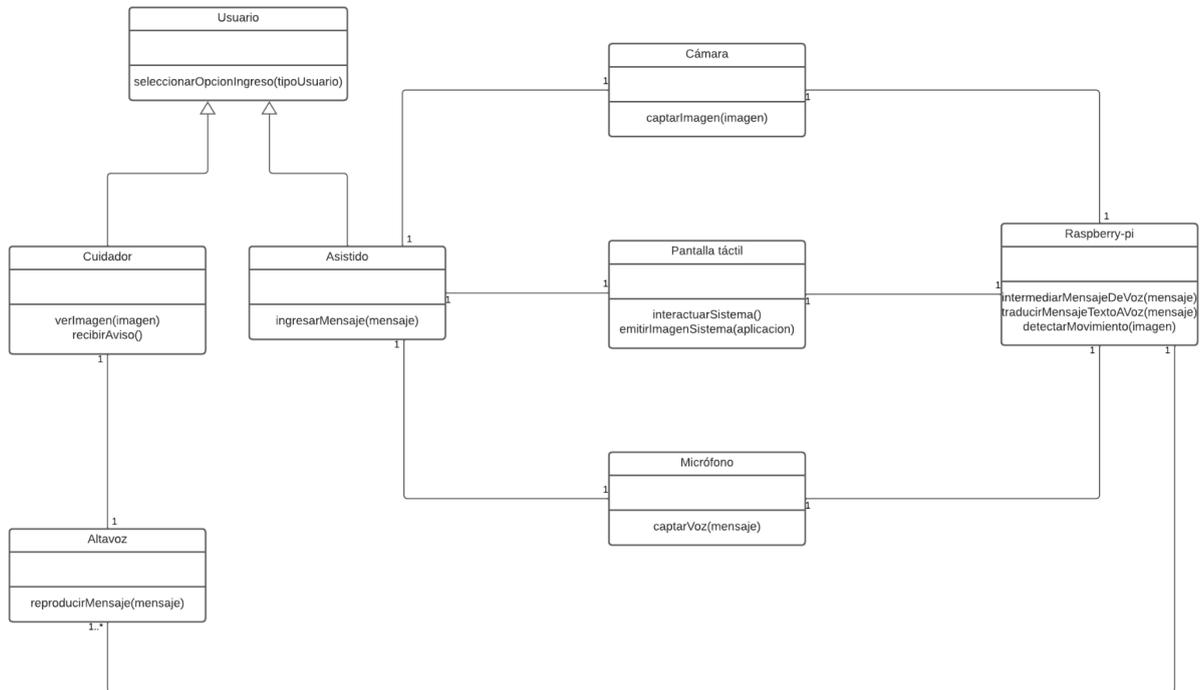
Caso de uso:	Mostrar interfaz "Información".
Descripción:	Muestra información de los creadores de la aplicación que el usuario está utilizando y un pequeño manual de uso dependiendo de las funcionalidades que el mismo tenga.
Actor:	Asistido, Cuidador.
Precondición:	La información debe estar redactada.
Flujo principal:	1.- El usuario selecciona el botón de "información". 2.- Se muestran en pantalla los créditos de la aplicación COSMIC y se encontraría un manual de usuario para visualizar.
Flujo Alternativo:	2.1.- Selecciona el manual de usuario. 2.2.- Se muestran las distintas funcionalidades de la aplicación.
Postcondiciones:	El usuario tiene acceso a la información y al manual de usuario de la aplicación.

e) Caso de uso: generar alerta de movimiento.



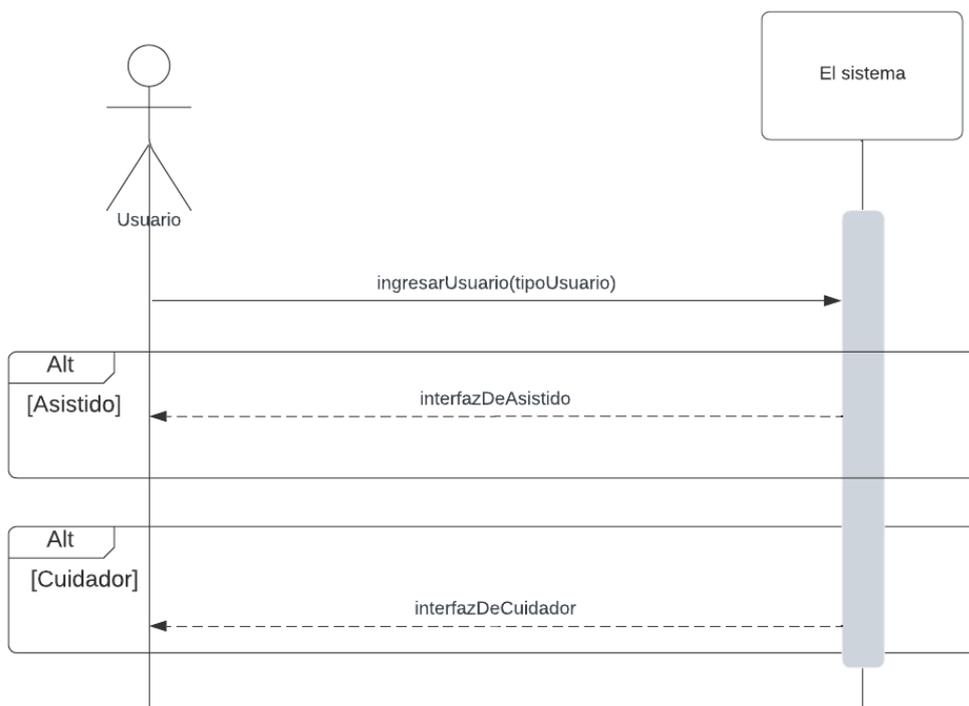
Caso de uso:	Generar alerta de movimiento
Descripción:	Se emite una alerta en el dispositivo del cuidador al detectarse movimiento del asistido.
Actor:	Cámara, Asistido, Cuidador.
Precondición:	La aplicación requiere una conexión estable entre la cámara y la aplicación. El usuario debe tener cargada la aplicación del sistema en su dispositivo, y además, se requiere que tenga instalada la aplicación Telegram para recibir notificaciones. El usuario también debe tener el rol de 'Cuidador' para acceder a las funciones y recibir alertas.
Flujo principal:	1.- <<Incluye>> caso de uso "Ingreso". 2.- La cámara capta la imagen del asistido. 4.- El sistema detecta movimiento del asistido. 3.- El sistema envía una alerta al dispositivo del cuidador.
Flujo Alternativo:	
Postcondiciones:	El usuario recibe un aviso en caso de movimiento.

### 11.1.3. Diagrama de clases

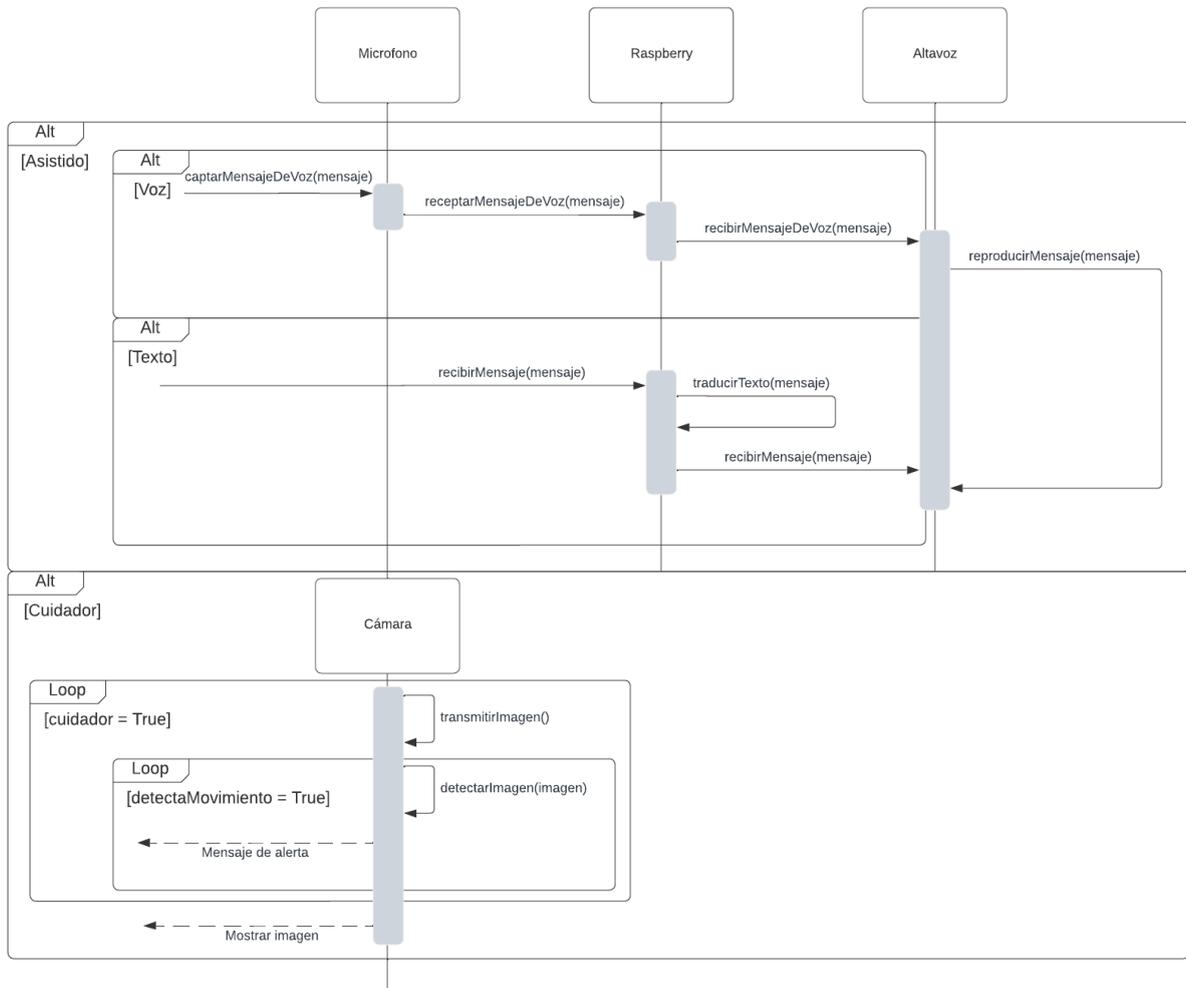


### 11.1.4. Diagrama de secuencias

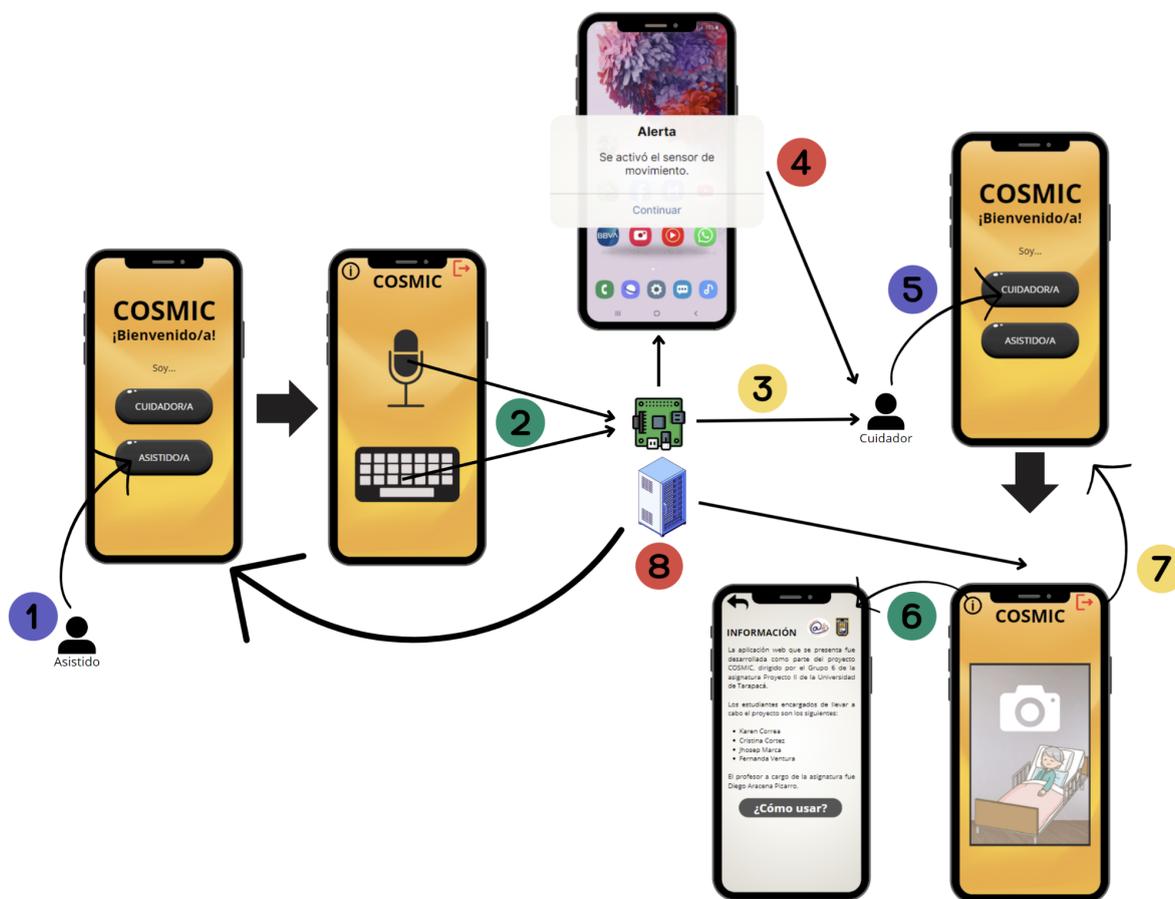
#### 11.1.4.1. Nivel 1



### 11.1.4.2. Nivel 2



## 11.2. Arquitectura de comunicación con respecto a diseños



- 1) El asistido selecciona su opción para entrar a su interfaz correspondiente.
- 2) El asistido tendrá dos opciones a seleccionar, en donde podrá elegir entre “Pulsar para hablar” y “Escribir para hablar”.
- 3) La tarjeta Raspberry-Pi procesa el mensaje del asistido. Si es en teclado, lo transforma en mensaje de audio y lo manda al cuidador por medio de los altavoces; si es en voz, envía directamente el mensaje al cuidador por los altavoces instalados en el hogar.
- 4) La tarjeta Raspberry-Pi detecta el movimiento a través de la cámara, y le manda una notificación por telegram al cuidador.
- 5) El cuidador selecciona su opción para entrar a su interfaz correspondiente.
- 6) El usuario (ya sea asistido o cuidador) selecciona el ícono de información, caracterizado por un símbolo “i” y muestra una vista con toda la información de la aplicación como información del proyecto realizado y un botón que al presionar muestra el manual de usuario general de la aplicación móvil.
- 7) El asistido y el cuidador tienen la opción de “cerrar sesión” para volver a la interfaz del inicio.
- 8) El servidor actúa como el núcleo central que coordina la comunicación y la funcionalidad entre diferentes partes de la aplicación, asegurando una experiencia de usuario coherente y segura tanto para el cuidador como para el asistido.

### 11.3. Diseño de interfaz de usuario



#### Interfaz de inicio:

En la pantalla de bienvenida COSMIC, extendemos una cálida bienvenida al usuario. En el centro de la interfaz, se destacan dos botones principales: "CUIDADOR/A" y "ASISTIDO/A". Aquí, el usuario tiene la libertad de elegir la opción que mejor se adapte. Una vez realizada la selección, la aplicación revela las funcionalidades específicas asociadas cambiando a la interfaz que corresponda.

#### Interfaz para el asistido:

En la pantalla destinada al asistido, al igual que en la interfaz del cuidador, se ubicarán los botones "Información" y "Salir". En el centro de la pantalla, se destaca un botón designado para el "Micrófono". Aquí, el asistido puede activar la función presionando antes de hablar, permitiendo la transmisión de mensajes a través de los altavoces. Justo debajo de este botón, se dispondrá un teclado, brindando la opción al asistido de escribir mensajes, los cuales serán transmitidos a través de los altavoces cuando así lo prefiera. Esta disposición facilita tanto la comunicación oral como escrita, ofreciendo flexibilidad y comodidad al asistido.



#### Interfaz para el cuidador:

En la pantalla destinada al cuidador, se dispondrá de un botón de "Información" en la esquina superior izquierda y, en la esquina superior derecha un botón de "Salir" que al ser activado retorna de inmediato a la pantalla de bienvenida. El centro de la pantalla albergará un recuadro que mostrará la imagen transmitida en tiempo real, brindando así una visualización centralizada y accesible de la información relevante.

### Interfaz de Información:

En esta pantalla, se encontrará un botón con la etiqueta "Retroceder", el cual al ser seleccionado te llevará de vuelta a la pantalla anterior. Además, se incluirán los créditos de la aplicación COSMIC junto con un archivo adjunto que contiene el manual de usuario, permitiendo una comprensión detallada y completa de todas las funcionalidades disponibles.



### Notificación de alerta:

Esta notificación se presentará al cuidador al detectar movimiento en la cámara, ya sea que esté activamente utilizando la aplicación COSMIC. Además, para garantizar una comunicación instantánea, la notificación se enviará tanto dentro de la aplicación como a través de Telegram.

## 11.4. Especificación de requerimientos

### a) Requerimientos funcionales:

- El sistema captará correctamente el texto escrito por el asistido en el monitor.
- El sistema traducirá el texto ingresado por el asistido a voz.
- El sistema captará correctamente el mensaje de voz por el asistido mediante el micrófono.
- El sistema reproducirá en el altavoz correctamente el mensaje transmitido por el asistido.
- El sistema detectará adecuadamente el movimiento del asistido mediante la cámara y enviará un mensaje de alerta al cuidador, notificándose tanto dentro de la aplicación COSMIC como a través de Telegram.

### b) Requerimientos no funcionales:

- Se contará con una vista para cuidadores y otra para los asistidos.
- El raspberry-pi del asistido y el dispositivo del cuidador deben estar conectados a la misma red wi-fi.
- El sistema debe funcionar 24 horas al día.
- La interfaz debe ser fácil de usar: no debe tener información sobrante, debe tener botones y elementos precisos y debe ser lo más intuitiva y ordenada posible. Además, debe ser amigable con el usuario: no tener colores confundibles, preferiblemente colores contrastantes para una mejor claridad, botones y elementos grandes y, finalmente, debe tener un diseño simple y sin muchas decoraciones ni distractores.
- El sistema debe tener un apartado al momento de iniciar la aplicación por primera vez, permitiendo al usuario escoger el tipo de usuario que será durante el uso de la aplicación; siendo "Cuidador" o "Asistido" las opciones disponibles.
- El sistema tendrá un apartado de información y ayuda, donde se encontrará toda la información del proyecto y un manual de usuario general.
- El sistema debe tener ingresado un manual de usuario simple, conciso y sencillo de leer. Con características amigables para el usuario como colores fuertes y contrastantes, explicación de los pasos con dibujos y flechas y fuentes de letras grandes y simples.
- El sistema debe generar una notificación en el teléfono del cuidador, ya sea dentro de la aplicación COSMIC o a través de Telegram, informando que el sistema detectó un movimiento.

## 11.5. Herramientas y técnicas

El proyecto necesitará de distintas herramientas para el desarrollo y realización del mismo. Las herramientas necesitadas son las siguientes.

Herramientas para la implementación:

- **Visual Studio Code:** Un entorno de desarrollo integrado (IDE) que facilita la escritura, depuración y administración de código. Utilizaremos Visual Studio Code para desarrollar y editar nuestro código Python y para gestionar el proyecto en general.
- **Python:** Un lenguaje de programación versátil y de alto nivel. Utilizaremos Python como el lenguaje principal para desarrollar la lógica de nuestra aplicación web.
- **Framework Flask:** Un framework web ligero para Python que simplifica el desarrollo de aplicaciones web. Flask nos permitirá construir rápidamente una aplicación web interactiva y escalable para Raspberry Pi.
- **Raspberry OS:** El sistema operativo diseñado específicamente para Raspberry Pi. Proporciona la plataforma sobre la cual ejecutaremos nuestra aplicación web basada en Flask.

Herramientas y dispositivos:

- Micrófono pequeño.
- Cámara pequeña.
- Monitor.

## 12. Planificación de procesos de soporte

### 12.1. Plan de integración

- **Comunicación entre el usuario asistido y el cuidador:** El asistido se puede comunicar con el cuidador a través de dos modos, por medio de transformación de texto a voz o transmitiendo el mensaje de voz que desea enviar el asistido al cuidador en un altavoz conectado por bluetooth.
- **Monitoreo del usuario asistido:** El monitoreo se lleva a cabo a través de una cámara usb conectada a la raspberry, la cual se encuentra captando constantemente la imagen del asistido, esta imagen llega a la vista del cuidador por medio de un servidor utilizando Flask y OpenCV.
- **Sistema de alerta de movimiento:** Se genera una alerta cuando se detecta el movimiento del asistido. Esto debido a que en la aplicación kivy que recibe la imagen de la cámara se comparan fotogramas para detectar algún cambio en dicha imagen, cuando se detecta el movimiento se envía un mensaje a través del bot de telegram al cuidador.

### 12.2. Descripción de arquitectura:

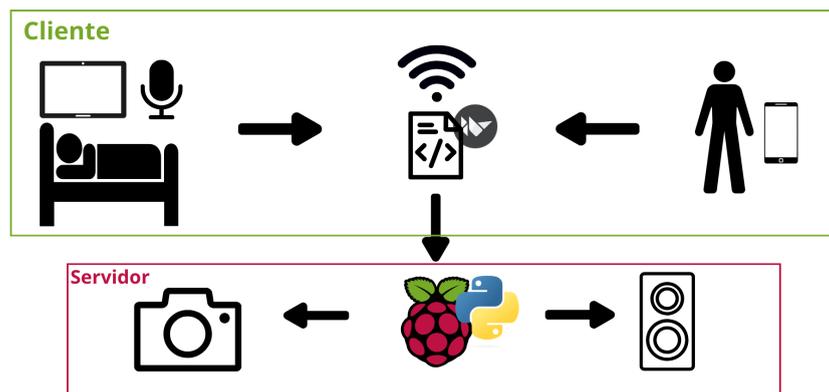
La arquitectura del proyecto se basa en la estructura cliente-servidor ya que se transmite una imagen utilizando Flask y OpenCV, de esta forma el cliente recibe esta imagen hacia la Aplicación Kivy, todo esto gracias a una IP donde los dispositivos deben estar conectados a la misma dirección IP, y se representa de la siguiente manera:

- **Cliente:**

El Cuidador utiliza la aplicación móvil, en donde se mostrará la imagen del asistido, gracias a kivy para el desarrollo de la interfaz, este recibirá la información del servidor para poder mostrar la imagen en directo.

- **Servidor:**

El servidor está formado de información entregada por la raspberry, gracias a Flask se creó el servidor y se muestra la información de la cámara, facilitando la entrega de información hacia el cliente, actuando como un intermediario entre la raspberry y el cliente.



### 12.3. Modelo de Implementación:

Tanto para la aplicación del asistido como para la del cuidador se utiliza Python en un IDE Visual Studio Code, para la aplicación del asistido se suma el framework flask. Algunas librerías y módulos que se utilizaron para el correcto funcionamiento de las aplicaciones, son los siguientes:

- OpenCV
- Numpy
- Telegram-bot
- Pyaudio
- KivyMD

### 12.4. Módulos Implementados:

#### 12.4.1. Módulo de monitoreo (generar imagen):

En este módulo se producen los fotogramas de la imagen de la cámara y se convierten .jpg con “cv2.imencode()”. En “@app.route('/video\_feed')” Define la ruta que devuelve la transmisión de video, utilizando “generate\_frames()”

```
def generate_frames():
    while True:
        success, frame = cap.read()
        if not success:
            break

        _, buffer = cv2.imencode('.jpg', frame)
        frame_bytes = buffer.tobytes()

        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame_bytes + b'\r\n\r\n')

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/video_feed')
def video_feed():
    return Response(generate_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')
```

### 12.4.2. Módulo de comunicación (iniciar y detener grabación de voz):

En este módulo se lleva a cabo la grabación de la voz del asistido para emitir su mensaje al cuidador, con “def iniciar\_grabacion()” y “def detener\_grabacion()” se realiza la grabación si no se está grabando previamente, y al detener la grabación, ésta se reproduce inmediatamente en los parlantes.

```
def iniciar_grabacion(self):
    if not self.recording:
        self.recording = True
        threading.Thread(target=self._grabar_audio).start()

def detener_grabacion(self):
    if self.recording:
        self.recording = False
        threading.Thread(target=self._reproducir_audio).start()
```

### 12.4.3. Módulo de comunicación (grabación de voz):

En este módulo se realiza la grabación del audio del asistido, con “open(self.file\_name, 'wb')” se crea un archivo de audio el cual se irá sobrescribiendo a medida que se grabe un audio nuevo, ocupando “self.frames.append(data)” se van grabando los datos en “self.frames”. Luego con “wave.open(self.file\_name, 'wb')” abre el archivo de audio “.wav” para sobrescribirlo.

```
def _grabar_audio(self):
    CHUNK = 1024
    FORMAT = pyaudio.paInt16
    CHANNELS = 2
    RATE = 44100

    p = pyaudio.PyAudio()

    with open(self.file_name, 'wb'):
        pass

    self.audio_stream = p.open(format=FORMAT,
                               channels=CHANNELS,
                               rate=RATE,
                               input=True,
                               frames_per_buffer=CHUNK)

    while self.recording:
        data = self.audio_stream.read(CHUNK)
        self.frames.append(data)

    self.audio_stream.stop_stream()
    self.audio_stream.close()
    p.terminate()

    with wave.open(self.file_name, 'wb') as wf:
        wf.setnchannels(CHANNELS)
        wf.setsampwidth(p.get_sample_size(FORMAT))
        wf.setframerate(RATE)
        wf.writeframes(b''.join(self.frames))

    self._reproducir_audio()

    self.frames = []
```

#### 12.4.4. Módulo de comunicación (teclado):

En este módulo se crea y muestra el teclado, en "def show\_keyboard(self)" se crea la ventana que contiene el diseño del teclado y con "def \_build\_keyboard\_layout(self)" se construye el teclado que se mostrará al asistido para que lo utilice si es de su preferencia.

```
def show_keyboard(self):
    self.keyboard_popup = Popup(title='Teclado',
                                content=self._build_keyboard_layout(),
                                size_hint=(None, None),
                                size=(400, 300),
                                auto_dismiss=True)
    self.keyboard_popup.open()

def _build_keyboard_layout(self):
    layout = BoxLayout(orientation='vertical', spacing=10, padding=10)

    text_input = TextInput(multiline=False, readonly=True, halign='center', font_size=24)
    layout.add_widget(text_input)

    buttons_layout = GridLayout(cols=10, spacing=5, size_hint_y=None, height=150)

    qwerty_layout = [
        'Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P',
        'A', 'S', 'D', 'F', 'G', 'H', 'J', 'K', 'L',
        'Z', 'X', 'C', 'V', 'B', 'N', 'M',
        'Space', 'Borrar', 'Enter'
    ]

    for button_text in qwerty_layout:
        buttons_layout.add_widget(Button(text=button_text, on_press=self._on_button_press))

    layout.add_widget(buttons_layout)

    return layout
```

En el siguiente método se indica que si se presiona el botón "Enter" se llama al método "self.reproducir\_texto(current\_text\_input.text)" para reproducir el texto.

```
def _on_button_press(self, instance):
    current_text_input = self.find_text_input(self.keyboard_popup.content)

    if instance.text == 'Enter':
        self.reproducir_texto(current_text_input.text)
        current_text_input.text = ''
    elif instance.text == 'Space':
        current_text_input.text += ' '
    elif instance.text == 'Borrar':
        current_text_input.text = ''
    else:
        current_text_input.text += instance.text
```

#### 12.4.5. Módulo de comunicación (teclado a voz):

En este módulo se realiza el texto a voz.

Con "def \_reproducir\_voz(self, texto):" se convierte el texto ingresado por el asistido a voz que será reproducido en el altavoz, en "engine = pyttsx3.init()" se inicializa la función que convertirá la voz a texto, con "engine.say(texto)" se define el texto a reproducir.

```
def _reproducir_voz(self, texto):
    engine = pyttsx3.init()
    engine.say(texto)
    engine.runAndWait()
```

#### 12.4.6. Módulo de monitoreo (capturar imagen):

En este módulo se captura la imagen mostrada en el servidor, y esto lo mostramos en la interfaz con el "super(CuidadorScreen, self)", para obtener el valor de la imagen que se encuentra en el servidor se hace un "self.video = video(source = "[http://172.20.10.4:5000/video\\_feed](http://172.20.10.4:5000/video_feed)")", le entregamos la dirección ip en donde se encuentra conectado la raspberry y en donde ejecuta el servidor, para así mostrar la imagen de la cámara que tiene la raspberry en el servidor y mostrarla al Cuidador, haciendo que este obtenga la imagen desde el servidor.

```
def on_enter(self, *args):
    super(CuidadorScreen, self).on_enter(*args)
    print("Entering CuidadorScreen")

    if not self.camera_initialized:
        print("Initializing camera")
        # Crear el widget Video solo si no se ha inicializado previamente
        self.video = Video(
            source="http://172.20.10.4:5000/video_feed",
            state="play",
            options={"allow_stretch": True},
        )
        self.video.bind(texture=self.texture_callback)
        self.ids.camera_layout.add_widget(self.video)

        # Crear etiqueta para la alerta de movimiento
        self.motion_label = Label(
            text="No se detecta movimiento", font_size="15sp", color=(1, 0, 0, 1)
        )
        self.ids.camera_layout.add_widget(self.motion_label)

        # Iniciar el procesamiento de la cámara en un hilo separado
        threading.Thread(target=self.detect_motion).start()

        # Marcar la cámara como inicializada para evitar la duplicación
        self.camera_initialized = True
```

### 12.4.7. Módulo de detección de movimiento:

En este módulo se captura la imagen del video que está siendo emitida en la ip correspondiente con "cv2.VideoCapture()", en el ciclo while se va monitoreando constantemente los fotogramas del video con "motion\_detected", para así detectar movimiento en la posición de los objetos de cada fotograma. si se detecta se envía una alerta con "show\_alert".

```
def detect_motion(self):
    cap = cv2.VideoCapture("http://172.20.10.4:5000/video_feed")
    _, prev_frame = cap.read()

    while True:
        _, frame = cap.read()
        diff = cv2.absdiff(prev_frame, frame)
        gray = cv2.cvtColor(diff, cv2.COLOR_BGR2GRAY)
        _, thresh = cv2.threshold(gray, 30, 255, cv2.THRESH_BINARY)
        contours, _ = cv2.findContours(
            thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE
        )

        motion_detected = any(cv2.contourArea(cnt) > 1000 for cnt in contours)

        if motion_detected:
            Clock.schedule_once(self.show_alert, 0)

        prev_frame = frame.copy()
```

### 12.4.8. Módulo de generación de alerta de movimiento:

En este módulo se genera la alerta al detectar movimiento en el módulo de detección de movimiento, se envía un mensaje predefinido en el código al cuidador, esto con el ID del telegram del usuario. Luego se resetea la alerta iniciando un nuevo hilo que reinicie la interfaz inicializando en "No se detecta movimiento" y se prepara para generar la próxima alerta, este hilo se genera para que la aplicación pueda seguir ejecutándose sin ser interrumpida.

```
def show_alert(self, *args):
    async def send_telegram_message(token, chat_id, message_text):
        bot = Bot(token=token)
        await bot.send_message(chat_id=chat_id, text=message_text)

    TOKEN = "6879188437:AAFO-HGke40IoUZMGN_EA16_vgcMfDufqhY"
    CHAT_ID = "5081858998"
    MESSAGE_TEXT = "Si se pudo burro"
    asyncio.run(send_telegram_message(TOKEN, CHAT_ID, MESSAGE_TEXT))
    self.motion_label.text = "¡Se detectó movimiento!"
    Clock.schedule_once(self.reset_alert, 5)

def reset_alert(self, *args):
    self.motion_label.text = "No se detecta movimiento"
    threading.Thread(
        target=self.detect_motion
    ).start() # Reiniciar la detección para la próxima alerta
```

### 13. Planificación de la documentación

Problemas encontrados y soluciones propuestas.

Problemas encontrados	Soluciones propuestas
No se pudo obtener una pantalla táctil para que el asistido pudiera utilizar sus diversas funciones.	Se optó por cambiar a un monitor más accesible que cumpliera con las necesidades del usuario.
La alarma originalmente estaba asignada para ser emitida desde el dispositivo, pero se identificó que programarla directamente era complicado, y además, se enfrentaba a restricciones de tiempo para su implementación.	Se modificó la configuración para que la alarma llegará a través de Telegram cuando el usuario está dentro de la aplicación COSMIC, proporcionando una solución más rápida y simplificando así la programación.
Se dificultó trabajar con la Tarjeta Raspberry-pi de forma que, en muchas ocasiones, no permitía avanzar con el proyecto, con diversos errores de importaciones y actualizaciones.	Se optó por formatear la tarjeta microSD la cual contenía el Sistema Operativo de la Tarjeta Raspberry-pi, haciendo un respaldo del código que se encontraba originalmente ahí, para así no perder el progreso realizado.
La escasez de recursos dificultó el desarrollo correcto del proyecto, limitando la creatividad y futuro potencial del proyecto.	Se consiguieron distintos sensores por cada integrante, pudiendo avanzar con el desarrollo del proyecto. Además, se consultó a otros profesores del departamento para así poder conseguir los recursos necesarios según la planificación inicial.
Desorganización simultánea entre los integrantes del grupo debido a la sobrecarga académica de los distintos ramos de alta exigencia.	Se conversaron las distintas problemáticas de los integrantes para así llegar a un consenso que beneficie a todos y puedan mantener un equilibrio entre los ramos. Se acordaron reuniones acorde al tiempo disponible de los integrantes del grupo y se mantuvo una actitud de comprensión hacia los integrantes que no pudieron asistir a las reuniones por el mismo motivo.

## **14. Conclusión**

La dificultad de comunicación en familias con personas en situación de dependencia es una realidad fehaciente en nuestro país, ésta problemática puede deteriorar mucho la calidad de vida tanto de la persona dependiente como de la persona cuidadora de ella. Éste problema fue detectado y analizado en una lluvia de ideas, realidad al principio del proceso de este proyecto. Luego de esto, se realizó nuevamente una lluvia de ideas entre los integrantes del grupo para poder buscar alguna solución para poder ayudar a esta problemática.

El resultado del proyecto ofrece una solución directa a dos problemas específicos encontrados en el ambiente y desenvolvimiento en el hogar de una persona con dependencia y un cuidador de ésta; los cuales son: una falta de monitoreo para la persona dependiente por parte del cuidador, debido a que la persona cuidador no puede estar el 100% de su tiempo vigilando a la persona dependiente; y, por otro lado, una falta de comunicación de las necesidades de la persona dependiente hacia la persona cuidadora, pues la persona cuidadora no estará todo el tiempo en la habitación y en esos momentos donde no hay monitoreo presencial, puede pasar que la persona dependiente tenga alguna necesidad o hasta algún accidente.

Sin duda, este proyecto ha pasado por varias etapas, empezando por la organización de los integrantes del grupo y la conformación del mismo, así como problemáticas en la configuración y programación de los recursos utilizados para llevar a cabo el proyecto. Sin embargo, lo más destacable del proyecto es su resultado, el cual nos entrega un producto funcional casi en su totalidad, pensado con la mejor de las intenciones para ir en ayuda de las personas en situación de dependencia y sus cuidadores, para así poder ayudar a mejorar su calidad de vida.

Como finalización del proyecto, se puede reflexionar sobre todo el proceso por el cual se pasó para poder llegar al resultado final, tanto cosas buenas como malas, pero de todas formas, se pudo llegar a la meta. No se puede evitar destacar el progreso que se tuvo tanto en los como grupo como estudiantes individualmente, en conocimiento y en organización grupal, se adquirieron conocimientos de los recursos que se utilizaron en la realización de este proyecto como la misma Tarjeta Raspberry-Pi, la implementación de los sensores, el sistema operativo nuevo con el que se trabajaba, entre otros; la adquisición de conocimientos es algo muy destacable del proyecto, ya que no se tenía conocimiento alguno sobre la forma de trabajar con la que tuvimos que emplear en este proceso.

Finalmente, la reflexión más grande que da este proyecto es “Nunca rendirse”, pues a pesar de todas las dificultades técnicas, problemas de tiempo y organización se pudo llegar a un resultado casi completamente satisfactorio tanto para el docente como para las expectativas iniciales que se tenían como grupo.

## **15. Trabajos futuros**

### **15.1. Implementación de una Base de Datos:**

Se podría mejorar considerablemente la experiencia del usuario con la implementación de una Base de Datos para que el usuario en caso de ser cuidador pueda registrarse con su ID de telegram y que éste quede registrado como cuidador y las siguientes veces que entre a la aplicación sea redirigido a su vista correspondiente de acuerdo a un token. De la misma forma el usuario asistido podría registrarse una vez y las próximas veces que entre a la aplicación sea redirigido a su vista designada.

### **15.2. Transmisión del monitoreo desde redes distintas:**

Se podría implementar un servicio el cual permita que se pueda acceder a la transmisión de la imagen desde cualquier dispositivo sin la necesidad de estar en la misma red de internet, esto podría proporcionar más seguridad, ya que en caso de que el cuidador tenga que salir del hogar pueda seguir monitoreando al asistido.

## 16. Referencias

- [1] Planner 5D. (s.f.). Diseño de Interiores en Línea. Recuperado de <https://www.planner5d.com/>
- [2] Emb.cl. (s.f.). Domótica en el hogar: Comodidad y eficiencia energética. Recuperado de <https://www.emb.cl/electroindustria/articulo.mvc?xid=3114&ni=domotica-en-el-hogar-comodidad-y-eficiencia-energetica>
- [2] Organización Panamericana de la Salud. (s.f.). Discapacidad. Recuperado de <https://www.paho.org/es/temas/discapacidad#:~:text=Entre%20las%20barreras%20a%20las.profesionales%20y%20las%20barreras%20financieras>
- [3] Talent.com. (s.f.). Salario de Diseñador Gráfico en Chile. Recuperado de <https://cl.talent.com/salary?job=dise%C3%B1ador+gr%C3%A1fico>
- [4] Talent.com. (s.f.). Salario de Jefe de Proyecto en Chile. Recuperado de <https://cl.talent.com/salary?job=jefe+de+proyecto>
- [5] Talent.com. (s.f.). Salario de Programador en Chile. Recuperado de <https://cl.talent.com/salary?job=programador#:~:text=Descubre%20cu%C3%A1l%20es%20el%20salario%20medio%20para%20Programador&text=%C2%BFcu%C3%A1nto%20gana%20un%20Programador%20en%20Chile%3F&text=El%20salario%20programador%20promedio%20en%20Chile%20es%20de%20%2410.200.>
- [6] Talent.com. (s.f.). Salario de Secretaria en Chile. Recuperado de <https://cl.talent.com/salary?job=secretaria#:~:text=%C2%BFcu%C3%A1nto%20gana%20un%20Secretaria%20en%20Chile%3F&text=El%20salario%20secretaria%20promedio%20en,con%20un%20ingreso%20de%20%244.560.>
- [6] Talent.com. (s.f.). Salario de Documentalista en Chile. Recuperado de <https://cl.talent.com/salary?job=documentalista>
- [7] MCI Electronics. (s.f.). Tarjetas Raspberry Pi. Recuperado de <https://mcielectronics.cl/shop/category/raspberry-pi/tarjetas-pi/>
- [10] Falabella. (s.f.). Kit de Audio Completo 2X10W Bluetooth USB SD MP3 RADIO Control R-Fonestar. Recuperado de [https://www.falabella.com/falabella-cl/product/118421268/Kit-Audio-Completo-2X10W-Bluetooth-USB-SD-MP3-RADIO-Control-R-Fonestar/118421269?rid=Recs%21PDP%21CL\\_F.com%21Rec1%21Viewed\\_Together\\_winnerv2%21Viewed\\_together\\_v2\\_ctrl%21119501507%21118421269%216%2113](https://www.falabella.com/falabella-cl/product/118421268/Kit-Audio-Completo-2X10W-Bluetooth-USB-SD-MP3-RADIO-Control-R-Fonestar/118421269?rid=Recs%21PDP%21CL_F.com%21Rec1%21Viewed_Together_winnerv2%21Viewed_together_v2_ctrl%21119501507%21118421269%216%2113)

[11] Xiaomi Online. (s.f.). Xiaomi Smart Camera C200. Recuperado de [https://xiaomionline.cl/xiaomi-smart-camera-c200?gclid=CjwKCAjwu4WoBhBkEiwAojNdXlfa2xnaMAEVo\\_zeUBQNzHKu7yWZHRhoB3YvLM03T2HghvjXaBbB6xoC6JMQAvD\\_BwE](https://xiaomionline.cl/xiaomi-smart-camera-c200?gclid=CjwKCAjwu4WoBhBkEiwAojNdXlfa2xnaMAEVo_zeUBQNzHKu7yWZHRhoB3YvLM03T2HghvjXaBbB6xoC6JMQAvD_BwE)

[12] <https://www.jotablueetooth.cl/mini-camara-espia-a9-hd-wifi>