

UNIVERSIDAD DE TARAPACÁ

FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática



**Sistema de comunicación asistencial para el
cuidado de personas en situación de dependencia
“ALBA”**

Autor(es): Daniel Alday

Tomás Silva

Juan Yampara

Asignatura: Proyecto II

Profesor: Diego Aracena

ARICA, 29 Agosto 2023

Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
29/08/2023	1.0	Nombre de proyecto y panorama general	Juan Yampara
10/09/2023	1.2	Revisión y anotaciones en el informe.	Daniel Alday Tomás Silva Juan Yampara
12/09/2023	1.3	Distribución del informe y avance en el resumen de proyecto.	Daniel Alday Tomás Silva Juan Yampara
18/08/2023	1.4	Edición de la redacción del resumen	Juan Yampara
24/09/2023	1.5	Finalización de tareas para la primera entrega del informe	Daniel Alday Tomás Silva Juan Yampara
30/12/2023	1.6	Finalización el informe	Daniel Alday Tomás Silva Juan Yampara

1. Tabla de contenidos

1. Tabla de contenidos	3
2. Panorama General	5
2.1. Resumen del proyecto	5
2.1.1. Objetivo general	5
2.1.2. Objetivos específicos	5
2.2.3. Suposiciones y Restricciones	6
2.2.4. Entregables del proyecto	6
3. Organización del proyecto	7
3.1. Personal y entidades internas	7
3.2. Roles y Responsabilidades	7
3.3. Mecanismos de comunicación y organización	8
4. Planificación de los procesos de gestión	8
4.1. Planificación de recursos	8
4.1.1. Programación del proyecto	8
4.1.2. Electrónica del proyecto	11
4.1.3. Recursos Humanos	16
4.2. Costos totales	18
4.2.1. Recursos tangibles	18
4.2.2. Recursos intangibles	18
4.2.3. Tabla de costos totales	18
4.3. Distribución de tiempos	19
4.3.1. Nueva Carta Gantt	19
4.3.2. Asignación de tiempos	20
4.4. Planificación de la gestión de riesgos	20
5. Planificación de los procesos técnicos	22
5.1. Modelos de proceso	22
5.1.1. Requerimientos	22
5.1.2. Casos de uso	24
5.1.3. Diagramas de secuencia	31
5.1.4. Boceto Interfaz de usuario	38
5.2. Herramientas y técnicas	41
6. Conclusión	42
7. Referencias	43
7.1. Referencias Adicionales	44

2. Panorama General

2.1. Resumen del proyecto

La discapacidad en Chile es un problema social que ha tenido un antes y después del estudio ENDISC[1] del 2004, ya que provocó un mayor compromiso para la inclusión y participación social de las personas en situación de discapacidad¹. Según el estudio ENDISC[1] del 2022, el 9.8% de la población adulta se encuentra en situación de dependencia, siendo el 3.8% de estas con una dependencia severa. Estos últimos son aquellos que requieren de un tercero para realizar algunas tareas o actividades diarias.

La labor de los cuidadores, por lo general, conlleva a la persona a sufrir de desgaste, sobrecarga y cuadros de estrés agudos, esto se agrava si el cuidador no posee las capacitaciones necesarias para el cuidado.

El proyecto “ALBA” busca mejorar la comunicación existente entre la persona en situación de dependencia y el cuidador, centrándose en el aumento de la independencia de la persona, facilitar el monitoreo de salud y reducir la carga laboral al personal encargado del cuidado de la persona.

2.1.1 Objetivo general

El objetivo de este proyecto es el siguiente:

“Desarrollar un sistema de comunicación asistencial para personas en situación de dependencia, que pueda proveer una mejora en la comunicación entre los usuarios, brindando más independencia a la persona en situación de dependencia y reduciendo la carga laboral para el(los) cuidador(es) respectivamente.”

2.1.2. Objetivos específicos

- Estudiar y definir los recursos necesarios para construir el sistema.
- Diseñar el sistema “ALBA” para su construcción.
- Planificar el desarrollo del proyecto.
- Desarrollar e implementar los conocimientos (desarrollo de software por ejemplo) y los recursos necesarios (uso del raspberry pi, sensores, entre otros) para desarrollar el sistema “ALBA”.
- Realizar pruebas de los instrumentos del sistema para verificar el estado del proyecto.

¹ «Folleto Tercer Estudio Nacional de la Discapacidad - III ENDISC 2022», *Servicio Nacional de la Discapacidad*, 2022. <https://www.senadis.gob.cl/descarga/i/7171/documento>

2.2.3. Suposiciones y Restricciones

Se espera que el proyecto sea un éxito se deberá cumplir con las siguientes cuestiones:

- La persona en situación de dependencia tendrá la capacidad de utilizar el sistema "ALBA".
- El cuidador tendrá la disposición y el tiempo para utilizar el sistema "ALBA".

Los recursos necesarios para construir el sistema estarán disponibles cuando se necesiten.

- El equipo de proyecto tendrá la experiencia y las habilidades necesarias para completar el proyecto.
- El sistema "ALBA" será aceptado por las personas en situación de dependencia y los cuidadores.
- El equipo completará y entregará los trabajos asignados en el tiempo estimado.
- Los instrumentos utilizados cumplirán con las funcionalidades esperadas para el sistema.

Las restricciones para este proyecto son las siguientes:

- **Compatibilidad entre software y hardware:** Versiones compatibles exactamente con otra.
- **Económico:** El costo de los elementos (sensores, cables, entre otros) debe ser de acuerdo a un presupuesto.
- **Usuarios o público definido:** Los usuarios que utilizarán este sistema serán aquellos que estén a cargo de una persona en situación de dependencia y la persona quien está siendo cuidada.

2.2.4. Entregables del proyecto

Los entregables del proyecto son los siguientes:

1. Primera Presentación del proyecto
2. Maqueta del sistema
3. Informe del proyecto (primera parte)
4. Segunda Presentación del proyecto
5. Informe del proyecto (segunda parte)
6. Redmine uta (Wiki, bitácoras y Carta Gantt)
7. Poster Promocional
8. Manual de usuario
9. Sistema "ALBA"

3. Organización del proyecto

3.1. Personal y entidades internas

Tabla 1: Personal y entidades internas

Entidades Internas	Personal encargado	Descripción
Jefe de Proyecto	Daniel Alday	El jefe de proyecto debe ser la persona encargada de gestionar y organizar el trabajo en equipo como también asegurarse de que las tareas se dividan de manera equitativa.
Jefe Analista	Tomás Silva	Un jefe analista tiene como objetivo principal el diseño y planificación del proyecto. Este se enfoca en comprender e investigar acerca de los requerimientos y restricciones relacionadas al "Sistema".
Jefe de Tecnología	Juan Yampara	Un jefe de tecnología es quien investiga los artefactos tecnológicos a utilizar, es decir, se encarga de su implementación, mantenimiento y desarrollo.

3.2. Roles y Responsabilidades

Tabla 2: Roles de los integrantes

Los roles que tomarán cada integrante del equipo serán:

Programador	Se encargará de diseñar la arquitectura de software, desarrollar aplicaciones y sistemas de control, además de realizar pruebas de funcionamiento del software para su depuración.
Técnico de hardware	Encargado de seleccionar los componentes de hardware (como sensores y actuadores) y ensamblar los prototipos de hardware, desarrollar pruebas del mismo y solucionar los problemas que se presenten relacionados con el funcionamiento del hardware.
Diseñador	Encargado de diseñar presentaciones, diagramas e interfaces y transmitir información compleja a través de ellas. Se encarga también de asegurar un mínimo de calidad y atractividad para el material audiovisual que genera.
Técnico de redes	Encargado de investigar sobre las herramientas y conocimientos necesarios para poder implementar una red y diseñar su estructura.

Especialista en documentación	Mantiene la documentación del proyecto actualizada manteniendo un registro constante de la información relevante para un proyecto.
--------------------------------------	--

3.3. Mecanismos de comunicación y organización

Un proyecto necesita coordinación entre los integrantes del equipo, es por ello que para el desarrollo del sistema "ALBA" se requieren de los siguientes medios de comunicación

- **Whatsapp:** Será el principal medio de comunicación por el cual los integrantes poseen mayor disponibilidad.
- **Discord:** Utilizado para reuniones, discord es un excelente medio para las reuniones y presentación en el equipo.
- **Google Drive:** Organización de archivos, carpetas y avances del sistema, google drive nos brindará un espacio en donde se pueda almacenar la información relacionada.
- **Obsidian:** Como medio de organización, sirve para que cada integrante del equipo elabore borradores, bosquejos o modelos de sus tareas asignadas. Este medio está conectado a google drive.
- **Redmine:** Ideal para almacenar la documentación y el seguimiento de los plazos.

4. Planificación de los procesos de gestión

4.1. Planificación de recursos

4.1.1. Programación del proyecto

Dentro de los proyectos relacionados con el Internet de las Cosas (IoT), es indiscutible la centralidad y relevancia que posee el software. Si bien los dispositivos electrónicos y los sensores son el aspecto exterior de cualquier sistema, es el software el que les da vida, inteligencia y la capacidad de trabajar en conjunto de manera programada y lógica.

En esta sección, se analizarán los aspectos financieros esenciales para la implementación del software de este proyecto.

Equipos para la programación

Para el apartado de programación del código necesario para el proyecto, se dispondrán equipos (notebooks) personales para cada integrante del grupo.

Total notebooks: 3

Software externo utilizado

De uso libre

Nombre del Software	Uso dado	Condiciones de uso
Visual Studio Code	Utilizado para programar los programas necesarios	Libre uso bajo los Términos de licencia de Microsoft
Lucidchart	Utilizado para diagramar y esquematizar interacciones con el sistema (usando UML)	Plan gratuito bajo los Términos de uso del Software Lucid
Canva	Utilizado para diseñar diagramas demostrativos	Libre uso bajo los Términos de uso de Canva
Google Workspace	Utilizado para facilitar el guardado en la nube y comunicación de archivos	Proporcionado por la institución UTA
WhatsApp	Utilizado para la comunicación informal del grupo de trabajo	Libre uso bajo los Términos de servicio de WhatsApp
Discord	Utilizado para la realización de reuniones remotas del grupo de trabajo	Libre uso bajo los Términos de servicio de Discord
Obsidian	Utilizado para anotar y organizar ideas del grupo de trabajo	Libre uso <i>personal</i> bajo los Términos de licencia de Obsidian
Redmine	Utilizado para planear y organizar tareas y documentos del proyecto	Proporcionado por la institución UTA
Sweet Home 3D	Utilizado para diagramar y visualizar la aplicación del sistema del proyecto en la vida real	Software de libre disposición. Véase Términos de licencia

Pagados

Nombre del Software	Uso dado	Coste de uso
Microsoft 365	Utilizado para suplir necesidades de ofimática (documentos y presentaciones)	\$34.999 CLP/mes (2-6 personas por licencia)

Sistema operativo Raspberry Pi

Para poder realizar la interconexión del código programado con la propia Raspberry Pi, es necesario contar con el sistema operativo apropiado para asegurar que esta misma funcione adecuadamente.

Para la Raspberry Pi, hay una ingente cantidad de sistemas operativos disponibles que pueden ser instalados en ella. Para propósitos de la programación inicial, se busca un sistema operativo fácil de usar y que provea de compatibilidad con los sensores que se usarán en el armado del sistema. Es por ello que, el sistema utilizado para el desarrollo de este proyecto será Raspberry Pi OS, el sistema operativo oficial secundado por los mismos creadores de Raspberry Pi².

“Raspberry Pi OS es un sistema operativo gratuito basado en Debian, optimizado para el hardware Raspberry Pi y es el sistema operativo recomendado para uso normal en una Raspberry Pi. El sistema operativo viene con más de 35.000 paquetes: software precompilado incluido en un formato agradable para una fácil instalación en su Raspberry Pi.

El sistema operativo Raspberry Pi está en desarrollo activo, con énfasis en mejorar la estabilidad y el rendimiento de tantos paquetes Debian como sea posible en Raspberry Pi.”

Traducido de:

“Raspberry Pi OS is a free operating system based on Debian, optimised for the Raspberry Pi hardware, and is the recommended operating system for normal use on a Raspberry Pi. The OS comes with over 35,000 packages: pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.

Raspberry Pi OS is under active development, with an emphasis on improving the stability and performance of as many Debian packages as possible on Raspberry Pi.”³

² R. P. Ltd, «Raspberry Pi OS – Raspberry Pi», *Raspberry Pi*. <https://www.raspberrypi.com/software/>

³ «Raspberry Pi Documentation - Raspberry Pi OS». <https://www.raspberrypi.com/documentation/computers/os.html>

4.1.2. Electrónica del proyecto

Así también como se necesita software para el desarrollo de este proyecto, se necesita también hardware para complementar su funcionamiento. Se listan a continuación los componentes electrónicos a usar en el transcurso de este proyecto.

Raspberry Pi 4

Es el núcleo fundamental del sistema. Conecta al usuario cuidador con las lecturas provistas por los sensores. Contiene el código y los aparatos necesarios para permitir esta interconexión. Se contempla el uso de una Raspberry Pi 4 para el desarrollo de este proyecto.

“La Raspberry Pi 4 es el último producto de la familia Raspberry Pi, con un procesador quad-core de 64 bits con 1,5 GHz, Red inalámbrica de banda dual 2.4/5 GHz, Bluetooth 5 / BLE, Ethernet más rápido (Gigabit y capacidad PoE a través de un PoE HAT separado).

La LAN inalámbrica de doble banda viene con certificación de cumplimiento modular, lo que permite que la placa se diseñe en productos finales con pruebas de cumplimiento de LAN inalámbrica significativamente reducidas, lo que mejora tanto el costo como el tiempo de comercialización.”⁴

Figura 1: Raspberry Pi 4

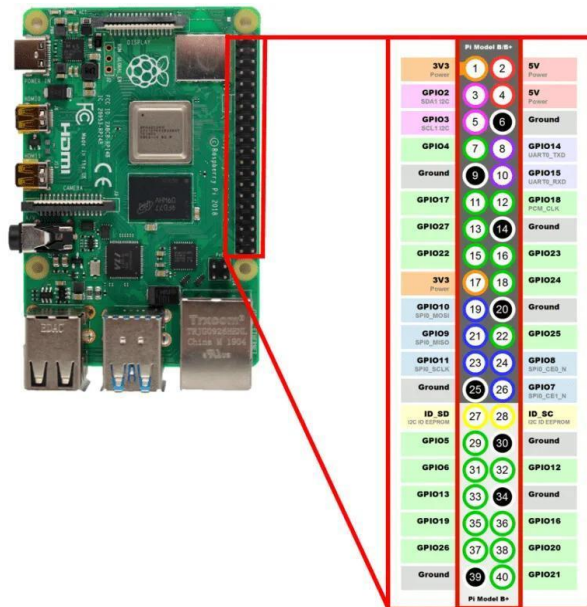


Fuente: Website raspberrypi.cl

⁴ Paguay, «Raspberry Pi 4 - Raspberry Pi», *Raspberry Pi*, sep. 2019, [En línea]. Disponible en: <https://raspberrypi.cl/raspberrypi-pi-4/>

Especificaciones técnicas⁵

Figura 2: GPIO de la raspberrypi



Fuente: Website raspberrypi.cl

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 1GB, 2GB or 4GB LPDDR4-2400 SDRAM (depende del modelo)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- GPIO de 40 pines estándar de Raspberry Pi (tiene total retrocompatibilidad con modelos anteriores)
- 2 puertos micro-HDMI (soporta hasta 4k 60fps)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- Puerto de 4 polos de audio estéreo y vídeo componente
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.0 graphics
- Puerto Micro-SD para el sistema operativo y almacenamiento masivo
- 5V DC por conector USB-C (mínimo 3A*)

⁵ Paguayo, «Raspberry Pi 4 - Raspberry Pi», *Raspberry Pi*, sep. 2019, [En línea]. Disponible en: <https://raspberrypi.cl/raspberry-pi-4/>

- 5V DC por header GPIO (mínimo 3A*)
- Power over Ethernet (PoE) enabled (necesita un PoE HAT aparte)
- Temperatura de trabajo: 0 – 50 grados C ambiente

Se puede usar un cargador de buena calidad de 2.5A si la carga total de los puertos USB es inferior a 500mA

Sensores

Complementando los objetivos principales del proyecto, la finalidad que se tiene con los sensores es proporcionar gran cantidad de datos fisiológicos generales a los usuarios del sistema, considerando también los costos adicionales que estos conllevan, y la incomodidad que estos puedan generar en la PSD (intrusividad).

Lista de sensores:

1. **Considerados**
 - a. Sensor de temperatura (por infrarrojo)
 - b. Sensor de ritmo cardíaco (por fotopleletismografía)
 - c. Cámara con visión nocturna
2. **No considerados**
 - a. Sensor de presión sanguínea
 - i. Esfigmomanómetro (Manual y Permanente)
 - ii. Cánula intravenosa
 - iii. Sensor ultrasónico
 - b. Sensor de presencia

Considerados

Tabla 5: Sensores considerados

Sensor de temperatura	Para proveer la temperatura corporal, se optará por el uso de un sensor térmico infrarrojo. Este sensor brinda lecturas rápidas, efectivas y poco intrusivas, ya que no requiere de contacto ni de un tiempo de actuación prolongado. ⁶
Sensor de ritmo cardíaco	Para proveer el ritmo cardíaco, se optará por el uso de un sensor de ritmo cardíaco por <i>fotopleletismografía</i> . ⁷

⁶ «Termómetro infrarrojo».

<https://mx.omega.com/prodinfo/termometro-infrarrojo.html#:~:text=Introducci%C3%B3n%20a%20los%20term%C3%B3metros%20infrarrojos,variaci%C3%B3n%20en%20la%20temperatura%20ambiente>

⁷ R. Medrán Medrán, «Medición del ritmo cardíaco mediante Fotopleletismografía», *e-reading «Trabajos y proyectos de estudios de la E.T.S.T»*, 2018. <https://biblus.us.es/bibing/proyectos/abreproy/12461> (accedido 24 de septiembre de 2023).

	Para su correcto funcionamiento, este sensor se sitúa inmediatamente sobre la piel de la PSD. Es notable también señalar que para asegurar que el sensor arroje un resultado efectivo, este requiere de una gran cantidad de mediciones. Por tanto, no es tan rápido al arrojar resultados. ^{8 9}
Cámara (con visión nocturna)	Teniendo en cuenta el enfoque propuesto en un principio, consideramos así también con este sensor proporcionar un vistazo rápido al estado general de la PSD, como lo puede ser su posición o postura dentro de su cuarto. Considerando la mayor cantidad de situaciones posibles, se optará por la capacidad de la cámara de ver en entornos con poca o nula luz (visión nocturna).

Fuente: Elaboración propia

No considerados

Los sensores que no cumplen con la visión de la propuesta de este proyecto por cualquier motivo son listados en esta sección. Ya sea por ser muy caros, muy lentos en arrojar resultados, o por ser muy intrusivos a la hora de ser instalados.

Tabla 6: Sensores no considerados

Sensor de presión sanguínea	Se consideró que proporcionar la presión sanguínea como dato es de alta relevancia para el usuario del sistema. Por razones de logística, un aparato como éste no existe en la medida que el proyecto requiere. Se listan las alternativas vistas. ¹⁰	Esfigmomanómetro (Oscilometría) Se considera la mejor de las alternativas en el momento de desarrollo de este proyecto. Tiene la cualidad de que su período de actuación es largo y es muy lento en arrojar resultados, además de ser muy grande a comparación de otros sensores mínimamente intrusivos. Posee dos formas de instalarse, manual y permanente. La forma manual tiene la ventaja de que es más cómoda la mayor parte del tiempo, puesto que con manual se refiere a instalar y retirar el esfigmomanómetro de la PSD a necesidad de la lectura. La forma automática contempla que la PSD tenga puesto el esfigmomanómetro todo el tiempo y que, frente a la necesidad de una lectura, este se active y arroje los resultados. Se considera que la eficacia que brinda esta forma de instalación no compensa la comodidad perdida por las razones
-----------------------------	---	---

⁸ «Aumento en la efectividad de la identificación de cimas y pies en el pulso fotopleletismográfico al reconstruirlo mediante filtrado adaptativo», *ScieELO*, [En línea]. Disponible en:

https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-99402017000100061&lang=pt

⁹ SriTu Hobby, «How to use the heart Pulse Sensor with Arduino | Heart Pulse Monitoring System», *YouTube*. 3 de octubre de 2021. [En línea]. Disponible en: <https://www.youtube.com/watch?v=aKus0FV4deU>

¹⁰ A. Al-Qatatsheh *et al.*, «Blood pressure sensors: materials, fabrication methods, performance evaluations and future perspectives», *Sensors*, vol. 20, n.º 16, p. 4484, ago. 2020, doi: 10.3390/s20164484.

		mencionadas anteriormente.
		<p>Cánula Intravenosa (Oscilometría) Por obvias razones, el sistema no cuenta con las capacidades para soportar el chequeo efectivo y más importante aún, cumpliendo con las regulaciones requeridas, el monitoreo activo del torrente sanguíneo de una persona.¹¹</p>
		<p>Sensor ultrasónico (Efecto Doppler) Se encuentra en desarrollo actualmente. Es una idea prometedora pero por el momento no se han encontrado pruebas físicas de que este sensor pueda funcionar en la actualidad. Se espera que se desarrolle en el futuro para poder implementarse en el sistema.^{12 13 14 15} [OBJ] [OBJ] [OBJ] [OBJ]</p>
Sensor de presencia	Consideramos que para el propósito de este proyecto, este sensor puede servir de gran ayuda. Sin embargo, la instalación de estos sensores es costosa e introduce complejidad innecesaria para los usuarios del sistema.	<p>Dependiendo del desarrollo futuro que pueda tener el proyecto, se considerará nuevamente si estos sensores pueden brindar algún tipo de ayuda extra a los usuarios del sistema.¹⁶</p>

¹¹K. Huang, F. Tan, T. Wang, y Y. Yang, «A highly sensitive Pressure-Sensing array for blood pressure estimation assisted by Machine-Learning techniques», *Sensors*, vol. 19, n.º 4, p. 848, feb. 2019, doi: 10.3390/s19040848.

¹² «Ultrasound: accurate pulse wave velocity and blood pressure | IMEC».

<https://www.imec-int.com/en/press/imecs-ultrasound-sensor-technology-yields-accurate-pulse-wave-velocity-and-blood-pressure>

¹³ M. Meusel *et al.*, «Measurement of blood pressure by Ultrasound—The applicability of devices, algorithms and a view in local hemodynamics», *Diagnostics*, vol. 11, n.º 12, p. 2255, dic. 2021, doi: 10.3390/diagnostics11122255.

¹⁴ N. Tamada, «US20160038117A1 - Ultrasonic blood pressure measuring device and blood pressure measuring Method - Google Patents», 11 de agosto de 2014. <https://patents.google.com/patent/US20160038117A1/en>

¹⁵ Waterloo Engineering, «Ultrasonic blood pressure sensor», *YouTube*. 1 de noviembre de 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=cjEPz9xqII0>

¹⁶ Enreta Domotica, «ADIOS a los SENSORES DE MOVIMIENTO - HOLA SENSOR DE PRESENCIA», *YouTube*. 16 de julio de 2023. [En línea]. Disponible en: <https://www.youtube.com/watch?v=1MXeFdh5080>

Equipos móviles

El sistema presentado por este proyecto cuenta con un subsistema que involucra una aplicación móvil. Con el fin de ir revisando los avances en este subsistema se usará un emulador para facilitar la integración con el proyecto y abaratar costos.

4.1.3. Recursos Humanos

Para el desarrollo del proyecto se consideran 16 semanas de trabajo continuo. Las horas de trabajo de cada rol se organizan y costean de la siguiente manera:

Costos hora por rol

Tabla 7: costos hora por rol

Rol	Costo/hora	Personal que cumple el rol
Programador	\$5.231 CLP/hora ¹⁷	Daniel Alday, Tomás Silva, Juan Yampara
Técnico de hardware	\$3.077 CLP/hora ¹⁸	Juan Yampara, Tomás Silva
Diseñador	\$3.692 CLP/hora ¹⁹	Tomás Silva, Daniel Alday
Técnico de redes	\$3.200 CLP/hora ²⁰	Juan Yampara, Daniel Alday
Especialista en documentación	\$5.538 CLP/hora ²¹	Daniel Alday, Tomás Silva, Juan Yampara

Fuente: Talent.com

Tiempo dedicado a cada rol (por tarea específica)

Las horas semanales mostradas a continuación no necesariamente se cumplen al mismo tiempo. Se considera la totalidad del proyecto al hablar de semanas y horas semanales.

Tabla 8: Tiempo dedicado a cada rol

Tarea específica	Semanas	Horas Semanales	Horas Totales
Programación	9 sem.	6 hrs./sem.	54 hrs.

¹⁷ «Salario para programador en Chile - salario medio», *Talent.com*.

¹⁸ «Salario para técnico informático en Chile - salario medio», *Talent.com*.

¹⁹ «Salario para Diseñador en Chile - Salario Medio», *Talent.com*.

²⁰ «Salario para Técnico Redes en Chile - Salario Medio», *Talent.com*.

²¹ «Salario para Control Documental en Chile - Salario Medio», *Talent.com*.

Proyecto II, Sistema de Comunicación Asistencial para Discapacitados

Preparación Hardware	11 sem.	3 hrs./sem.	33 hrs.
Diseño	7 sem.	1.5 hrs./sem.	10.5 hrs.
Investigación de tecnologías de redes	3 sem.	4.5 hrs./sem.	13.5 hrs.
Documentar el proyecto	16 sem.	1.5 hrs./sem.	24 hrs.
Total horas		~8.4 hrs./sem.	135 hrs.

Fuente: Elaboración propia

Costo de la contratación

Tabla 9: Costos de la contratación

Rol contratado	Horas requeridas p./p.	N° personas encargadas	Total horas requeridas	Costo total contratación
Programador	54 hrs.	3	162 hrs.	\$847.422 CLP
Técnico de hardware	33 hrs.	2	66 hrs.	\$203.082 CLP
Diseñador	10.5 hrs.	2	21 hrs.	\$77.532 CLP
Técnico de redes	13.5 hrs.	2	27 hrs.	\$86.400 CLP
Especialista en documentación	24 hrs.	3	72 hrs.	\$398.736 CLP
Costo total recursos humanos				\$1.613.172 CLP

4.2. Costos totales

4.2.1. Recursos tangibles

Tabla 10: Recursos tangibles

Elemento	Uso	Unidades	Tipo de pago	Coste unitario est.	Coste total est.
Notebooks	Equipos dedicados a la codificación	3	Un solo pago	\$500.000 CLP	\$1.500.000 CLP
Raspberry Pi (4GB RAM)	Ser la pieza principal del sistema	1	Un solo pago	\$79.000 CLP	\$79.000 CLP
Sensores	Detectar, medir y digitalizar informaciones del mundo físico	1 (p/sensor)	Un solo pago	~ \$20.000 CLP	~ \$75.000 CLP ²²
				~ \$5.000 CLP	
				~ \$20.000 CLP	
				~ \$30.000 CLP	

Fuente: Elaboración propia

4.2.2. Recursos intangibles

Tabla 11: Recursos intangibles

Elemento	Uso	Unidades	Tipo de pago	Coste unitario est.	Coste total est.
Software (Licencias pagadas)	Facilitar tareas administrativas y de desarrollo	1 (servicio por suscripción)	Mensual	\$6.990 CLP/mes (5 meses)	\$34.950 CLP

Fuente: Elaboración propia

4.2.3. Tabla de costos totales

Tabla 12: Costos totales

Elemento	Costo
Recursos tangibles	\$1.654.000 CLP

²² Precios obtenidos de distintas páginas chilenas. Véase las referencias.

Recursos intangibles	\$34.950 CLP
Recursos humanos	\$1.613.172 CLP
Costo total proyecto	\$3.302.122 CLP

Fuente: Elaboración propia

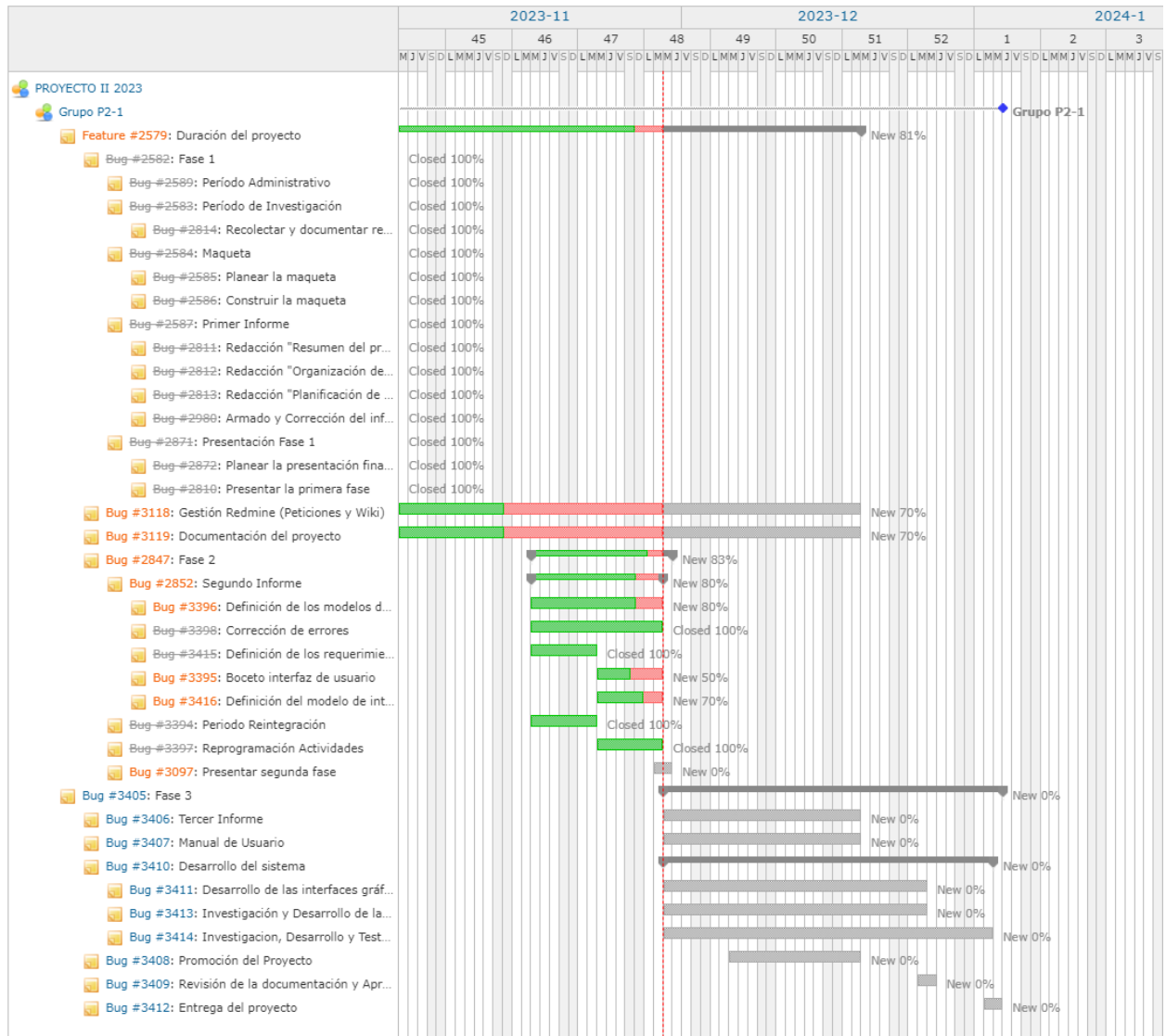
4.3. Distribución de tiempos

4.3.1. Nueva Carta Gantt

Para propósitos de organización de tiempos y actividades, se re-elaboró una carta Gantt que permitirá visualizar y gestionar de manera eficiente el desarrollo del proyecto ALBA.

En esta sección se muestra cómo se planificó las tareas, etapas cruciales y los tiempos requeridos para su concreción.

Proyecto II, Sistema de Comunicación Asistencial para Discapacitados



Para las etapas más próximas el detalle de las tareas es mayor. Según el desarrollo del proyecto vaya avanzando, se irán agregando tareas más y más específicas según corresponda.

4.3.2. Asignación de tiempos

Para el desarrollo del proyecto se contemplan 16 semanas de trabajo, para las cuales, se trabajará una media de 12 horas semanales.

La duración total del proyecto está estimada en 192 horas de trabajo repartidas en 4 meses de desarrollo.

Etapas previstas

El proyecto se dividirá en tres fases, numeradas del 1 al 3.

Tabla 13: Etapas previstas

Fase 1	Semanas comprendidas: 1 a la 5 (5)	Del 16/08 al 26/09
Fase 2	Semanas comprendidas: 6 a la 11 (6)	Del 27/09 al 28/11
Fase 3	Semanas comprendidas: 12 a la 16 (5)	Del 29/11 al 03/01

Fuente: Elaboración

4.4. Planificación de la gestión de riesgos

Los posibles riesgos que se puede enfrentar el equipo se encuentran en la siguiente tabla

Tabla 14: probabilidad, riesgos e impacto

#	Riesgo	Probabilidad	Impacto
1	Funcionamiento deficiente del personal	20%	Marginal (3)
2	Poca investigación acerca de las interfaces y comunicación con raspberry Pi	40%	Crítico(2)
3	Distribución de tareas de manera deficiente	10%	Marginal(3)
4	Ausencia del personal por enfermedad	40%	Crítico(2)
5	Funcionamiento defectuoso de los dispositivos involucrados	20%	Catastrófico (4)
6	Daño del material utilizado para construir el sistema	30%	Catastrófico(4)
7	Pérdida de la documentación del proyecto	10%	Catastrófico(4)
8	Administración del tiempo de forma ineficiente	20%	Marginal(3)

9	Escasa documentación relacionada a los subsistemas del proyecto	40%	Marginal(3)
10	Retraso en las entregas del proyecto	20%	Catastrófico(4)

Fuente: Elaboración propia

Las acciones para remediar estas situaciones son las siguientes:

1. **Funcionamiento deficiente del personal** : Realizar evaluaciones de desempeño periódicas, fomentar una comunicación abierta y retroalimentación para abordar preocupaciones.
2. **Poca investigación acerca de las interfaces y comunicación con Raspberry Pi**: Destinar tiempo adicional para investigar y probar las interfaces con Raspberry Pi antes de integrarlas en el proyecto.
3. **Mala distribución de tareas**: Establecer reuniones regulares de planificación y seguimiento para garantizar una distribución equitativa de las tareas.
4. **Falta por enfermedad por parte del personal**: Establecer un plan de contingencia para distribuir tareas en caso de ausencia.
5. **Funcionamiento defectuoso de los dispositivos involucrados**: Realizar pruebas rigurosas y mantenimiento preventivo de los dispositivos, y tener un plan de contingencia para la rápida sustitución de dispositivos en caso de fallos.
6. **Daño del material utilizado para construir el sistema**: Almacenar adecuadamente los materiales y equipos, evitando condiciones adversas.
7. **Pérdida de la documentación del proyecto**: Implementar un sistema de gestión de documentos sólido, realizar copias de seguridad regulares.
8. **Mala administración de los tiempos**: Utilizar herramientas de organización y planificación para poder estar consciente de los plazos de entrega.
9. **Poca documentación relacionada a los subsistemas del proyecto**: Priorizar la documentación relacionada a los subsistemas y registrar la documentación existente encontrada por parte de algún miembro del equipo.
10. **Retraso en las entregas del proyecto**: Monitorea de cerca el avance del proyecto y toma medidas preventivas para evitar retrasos.

5. Planificación de los procesos técnicos

5.1. Modelos de proceso

5.1.1. Requerimientos

Los requerimientos funcionales y no funcionales son pilares fundamentales en el diseño y desarrollo de sistemas, proporcionando la estructura esencial para la creación de soluciones tecnológicas que cumplen con las necesidades y expectativas de sus usuarios.

Requerimientos funcionales

A continuación presentamos los requerimientos funcionales de nuestro sistema.

1. **Comunicación entre el PSD y el cuidador:** Capacidad para establecer comunicación entre el paciente y el cuidador a través de mensajes texto de voz
2. **Interfaz de usuario intuitiva:** Diseño de una interfaz fácil de usar para el cuidador, permitiendo una comunicación fluida y accesible
3. **Registro de mensajes:** Posibilidad de almacenar mensajes anteriores para referencia futura o seguimiento de conversaciones.
4. **Registro de los datos de los sensores:** El sistema deberá registrar los datos del paciente como la frecuencia cardiaca y la temperatura.
5. **Alerta al cuidador:** El sistema enviará una alerta cuando los datos recopilados de algún sensor estén fuera de algún rango establecido.
6. **Registro de los datos del paciente:** El sistema podrá mantener información personalizable relevante para el cuidado del paciente

Requerimientos no funcionales

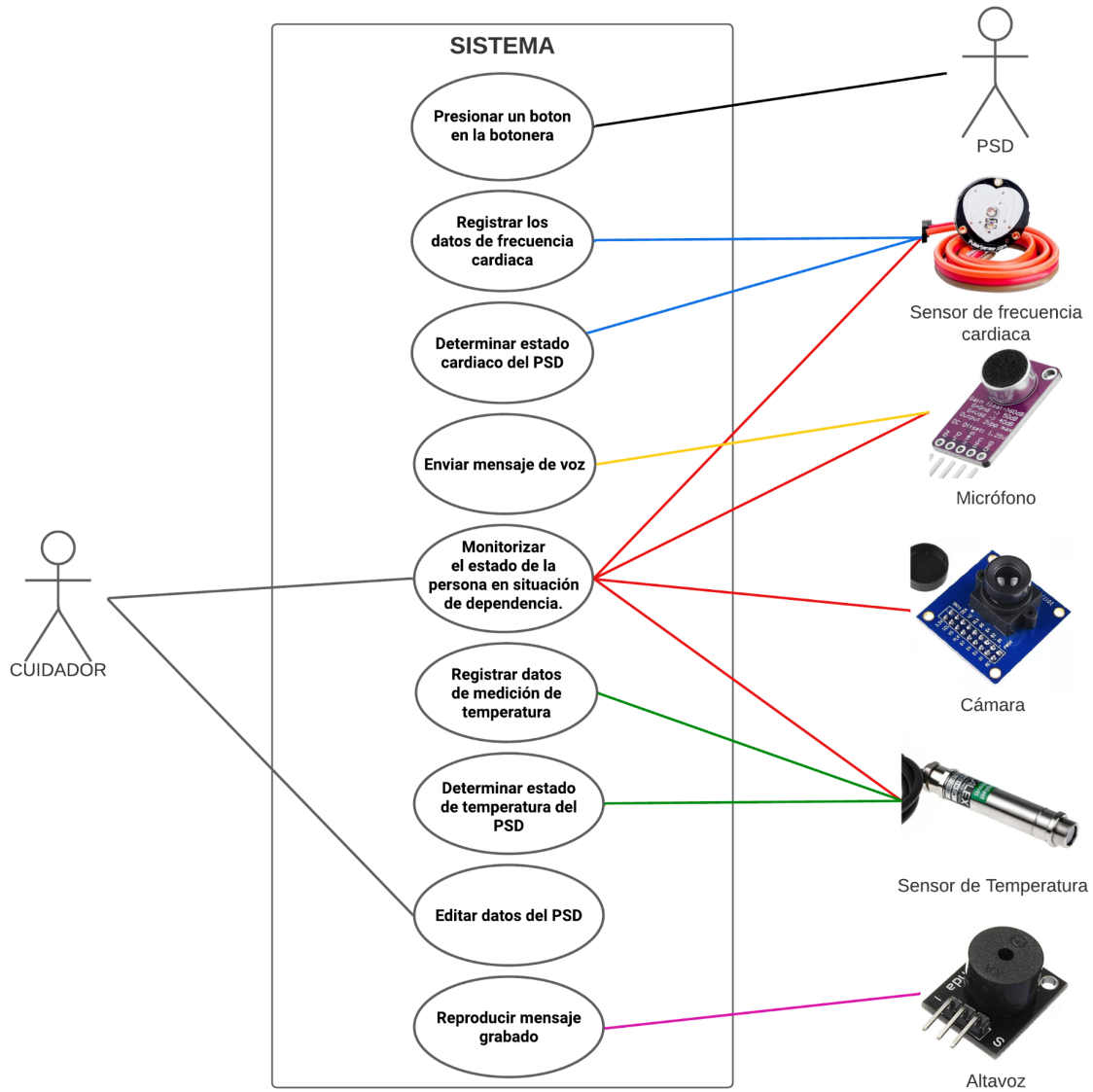
1. **Seguridad de la información:** Garantizar la seguridad y privacidad de las conversaciones y datos sensibles intercambiados entre el paciente y el cuidador.
2. **Disponibilidad y confiabilidad:** Mantener el sistema disponible en todo momento para garantizar una comunicación ininterrumpida.
3. **Baja latencia:** Minimizar el retraso en la comunicación para una interacción más natural y efectiva.
4. **Conexión estable a Wifi:** Garantizar una conexión confiable y estable entre los dispositivos para facilitar la comunicación continua.

5. **Capacidad de personalización:** Permitir ajustes y personalización según las necesidades individuales del paciente y el cuidador.

5.1.2. Casos de uso

Diagrama de casos de uso base

Figura 3: caso de uso base



Fuente: Elaboración propia

Descripción de casos de uso.

Tabla 14: Caso de uso monitorizar el estado de la persona en situación de dependencia

Nombre CUS: Monitorizar el estado de la persona en situación de dependencia.	
Resumen: El cuidador revisa el estado del PSD según los datos de los diferentes sensores.	
Actor: Cuidador, sensor de temp., sensor de frecuencia card, cámara	
Precondición: Debe estar registrado los datos tratados de los sensores en el sistema y el estado del PSD. La cámara deberá estar conectada al sistema.	
Flujo Principal: Cuidador 2. El cuidador selecciona ver las mediciones del sensor de temperatura	Flujo Principal: Sistema 1. El sistema muestra las siguientes opciones: a. Ver las mediciones del sensor de temperatura b. Ver las mediciones del sensor de frecuencia cardiaca c. Ver cámara 3. El sistema debe mostrar los datos tratados recibidos del sensor de temperatura y mostrar el estado del PSD.
Alternativo: 2.1 El cuidador selecciona ver las mediciones del sensor de frecuencia cardiaca.	3.1 El sistema debe mostrar los datos tratados recibidos del sensor de frecuencia cardiaca y mostrar el estado del PSD.
Alternativo: 2.2 El cuidador selecciona ver la camara	3.2 El sistema muestra las imágenes transmitidas por la cámara
Postcondiciones: --	

Fuente: Elaboración propia

Tabla 15: caso de uso de registrar datos de medición de temperatura

Nombre CUS: Registrar datos de medición de temperatura	
Resumen: El Sensor de temperatura al estar conectado al sistema, envía los datos para que estos sean tratados, definiendo el estado del PSD para almacenarlo.	
Actor: Sensor Temperatura	
Precondición: --	
<p>Flujo Principal: Sensor Temperatura</p> <p>2. El sensor mide y envía las mediciones de temperatura en grados Celsius al sistema.</p>	<p>Flujo Principal: Sistema</p> <p>1. El sistema pide que el sensor mida la temperatura del PSD</p> <p>3. El sistema almacena la temperatura corporal del PSD, la fecha y hora del registro.</p>
Postcondiciones: Se registra la temperatura, la fecha y hora del registro.	

Fuente: Elaboración propia

Tabla 16: Registrar los datos de frecuencia cardiaca

Nombre CUS: Registrar los datos de frecuencia cardiaca	
Resumen: El Sensor de frecuencia cardiaca al estar conectado al sistema, envía los datos para que estos sean almacenados.	
Actor: Sensor de frecuencia cardiaca	
Precondición: El PSD debe estar asociado al sistema	
<p>Flujo Principal: frecuencia cardiaca</p> <p>2. El sensor envía las mediciones de la frecuencia cardiaca del PSD</p>	<p>Flujo Principal: Sistema</p> <p>1.El sistema le pide al sensor enviar datos de frecuencia cardiaca cada un periodo de tiempo determinado.</p>

	3.El sistema almacena la frecuencia cardiaca del PSD y lo asocia a un fecha y hora.
Postcondiciones: Se registra la frecuencia cardiaca y el estado del PSD	

Fuente: Elaboración propia

Tabla 17: Caso de uso presionar un botón de la botonera

Nombre CUS: Presionar un botón de la botonera	
Resumen: El P.S.D envía un mensaje preestablecido a su cuidador.	
Actor: PSD	
Precondición: Deben estar registrados los mensajes en el sistema	
Flujo Principal: PSD 1. El PSD presiona uno de los botones de la botonera.	Flujo Principal: Sistema 2. El sistema envía una notificación al cuidador con el mensaje asociado al botón.
Postcondiciones:	

Fuente: Elaboración propia

Tabla 18: Caso de uso enviar mensaje de voz

Nombre CUS: Enviar Mensaje de voz	
Resumen: El P.S.D quiere enviar un mensaje de voz a su cuidador.	
Actor: PSD	
Precondición: --	
Flujo Principal: PSD 1.El PSD mantiene el botón para poder grabar un mensaje de voz.	Flujo Principal: Sistema 2. El sistema envía este mensaje de voz al cuidador.
Postcondiciones: El mensaje de voz queda registrado en la app del cuidador.	

Fuente: Elaboración propia

Tabla 19: Caso de uso determinar estado cardíaco del PSD

Nombre CUS: Determinar estado cardíaco del PSD	
Resumen: El raspberry pi determina el estado cardíaco del PSD dada la frecuencia cardiaca registrada.	
Actor: Raspberry pi	
Precondición: Los datos del sensor de frecuencia cardiaca deben estar registrados en el sistema.	
<p>Flujo Principal: Raspberry pi</p> <p>1. La Raspberry Pi pide los datos de frecuencia cardiaca al sistema.</p> <p>3. La Raspberry pi recibe los datos y determina el estado del PSD dependiendo de:</p> <ul style="list-style-type: none"> a. Si la frecuencia cardiaca es mayor que la frecuencia máxima, se marcará como "Posible Taquicardia" b. Si la frecuencia cardiaca es menor que la frecuencia mínima se marcará como: "Posible Bradicardia" c. Si la frecuencia cardiaca es "normal", se marcará como: "normal" 	<p>Flujo Principal: Sistema</p> <p>2. El sistema envía los datos de frecuencia cardiaca.</p> <p>4. El sistema determina el estado cardíaco del PSD.</p>
Postcondiciones: El estado cardíaco del PSD queda registrado en el sistema.	

Fuente: Elaboración propia

Tabla 20: Caso de uso determinar estado de temperatura corporal del PSD.

Nombre CUS: Determinar estado de temperatura corporal del PSD	
Resumen: El raspberry pi determina el estado de temperatura corporal del PSD dada la temperatura corporal registrada.	
Actor: Raspberry pi	
Precondición: Los datos del sensor de temperatura corporal deben estar registrados en el sistema.	
<p>Flujo Principal: Raspberry pi</p> <p>1. La Raspberry Pi pide los datos de temperatura corporal al sistema.</p> <p>3. El sistema marca el estado del PSD dependiendo de:</p> <ul style="list-style-type: none"> a. Si la temperatura (C°) es mayor a 38 grados la marcará como: "Posible Fiebre" b. Si la temperatura (C°) es menor a 37 grados lo marcará como : "Posible Hipotermia" c. Si la temperatura (C°) está contenido entre los intervalos de 38° y 37°, se marcará como: "Normal". 	<p>Flujo Principal: Sistema</p> <p>2. El sistema envía los datos de temperatura corporal.</p> <p>4. El sistema registra el estado de la temperatura corporal del PSD.</p>
Postcondiciones: El estado de la temperatura corporal del PSD queda registrado	

Fuente: Elaboración propia

Tabla 21: Caso de uso registro de los datos del PSD

Nombre CUS: Registro de los datos del PSD	
Resumen: El cuidador registra los datos (nombres, apellidos, edad, altura, peso, sexo, enfermedades, alergias y tipo de sangre)	
Actor: Cuidador	
Precondición: --	
Flujo Principal: Cuidador 1. El cuidador registra los datos del PSD (nombres, apellidos, altura, peso, sexo, fecha nacimiento, enfermedades, alergias y tipo de sangre)	Flujo Principal: Sistema 2. El sistema almacena los datos del PSD
Postcondiciones: Los datos del PSD quedan registrados en el sistema.	

Fuente: Elaboración propia

Tabla 22: Caso de uso editar datos del PSD

Nombre CUS: Editar datos del PSD	
Resumen: El cuidador puede editar datos del PSD tales como nombre, apellido, fecha nacimiento, peso, altura, sexo, tipo de sangre, alergias y enfermedades.	
Actor: Cuidador	
Precondición: Debe estar registrado los datos del PSD (nombre y apellido)	
Flujo Principal: Cuidador 1. El cuidador ingresa los datos: Nombre, apellido, fecha nacimiento, altura, peso, sexo, tipo de sangre, alergias y enfermedades.	Flujo Principal: Sistema 3. El sistema almacena los datos, actualizando cada uno.
Postcondiciones: Se actualiza los datos tales como peso, altura, sexo, tipo de sangre, alergias y enfermedades	

Fuente: Elaboración propia

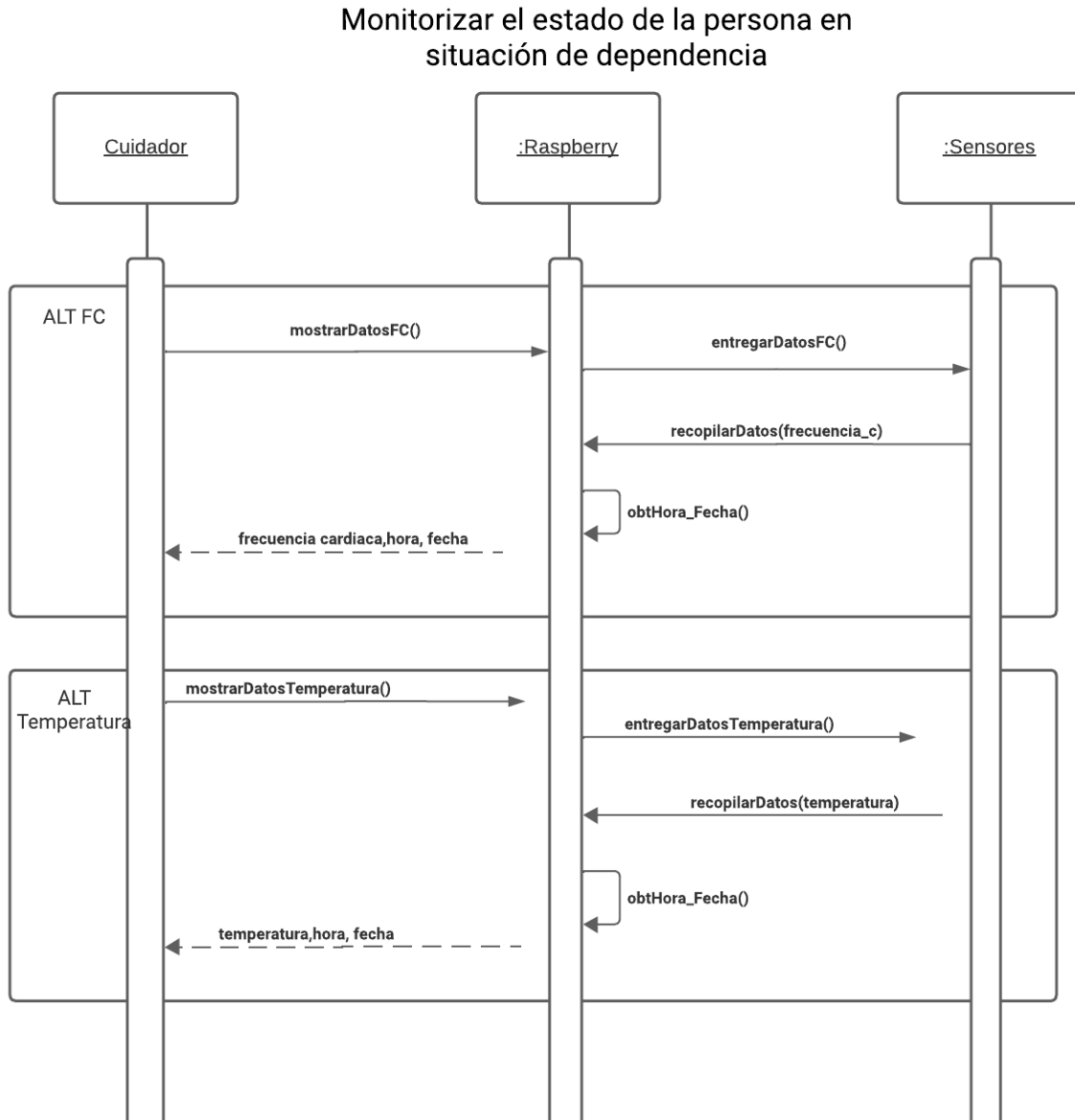
Tabla 23: Caso de uso reproducir mensaje grabado

Nombre CUS: Reproducir mensaje grabado	
Resumen: El altavoz recibe una pista de audio referente a un mensaje de voz y la reproduce.	
Actor: Altavoz	
Precondición: Debe estar registrado algún mensaje de voz	
Flujo Principal: Altavoz 2. El altavoz reproduce la pista de audio.	Flujo Principal: Sistema 1. El sistema envía una pista de audio.
Postcondiciones: –	

Fuente: Elaboración propia

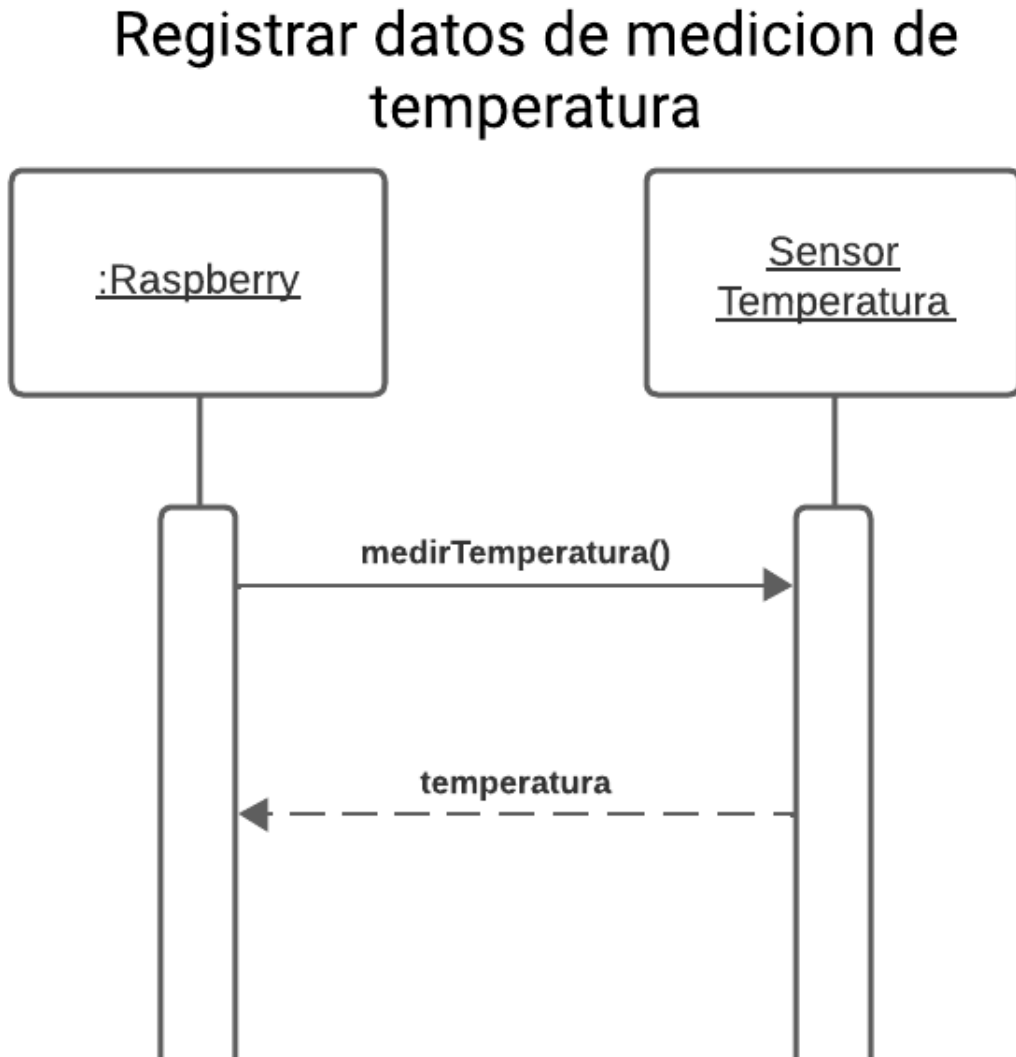
5.1.3. Diagramas de secuencia

Figura 4: Monitorizar el estado de la persona en situación de dependencia



Fuente: Elaboración propia

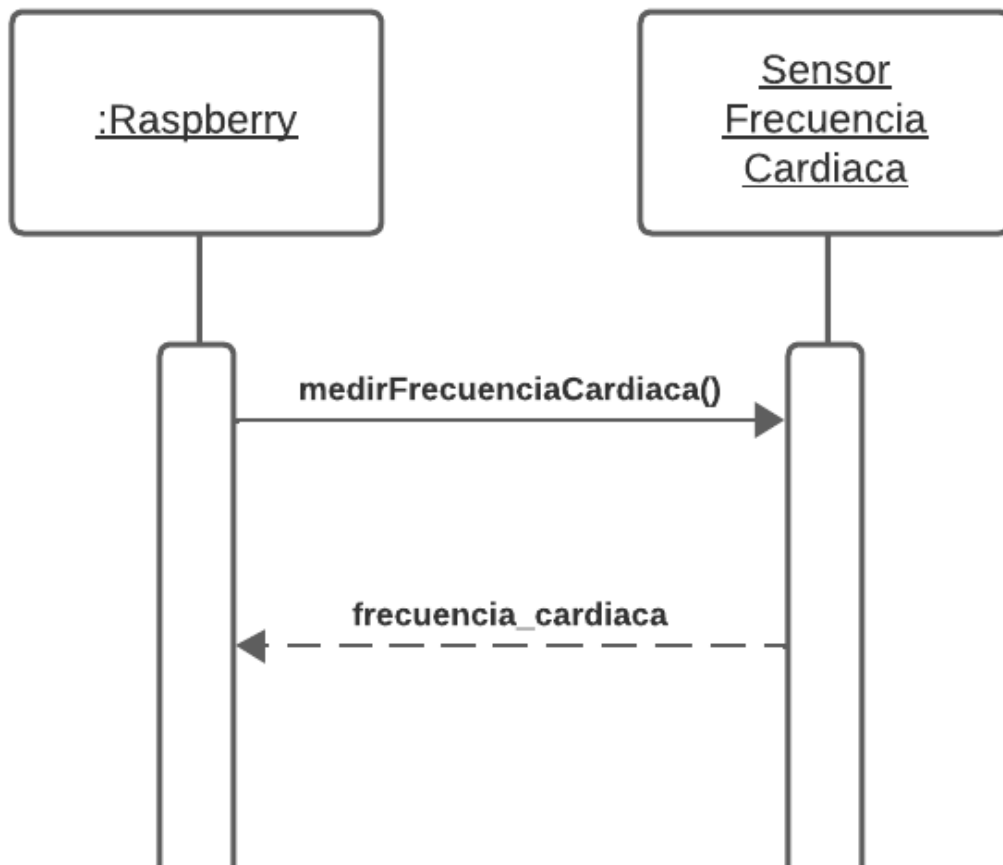
Figura 5: Registrar datos de medición de temperatura



Fuente: Elaboración propia

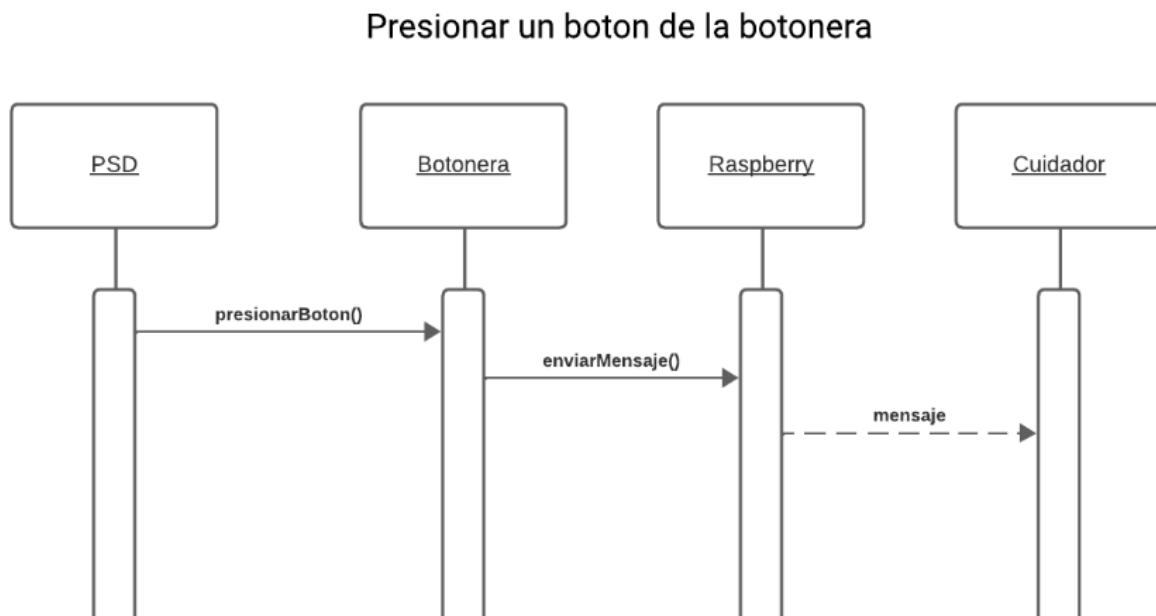
Figura 6: Registro los datos de frecuencia cardiaca

Registrar los datos de frecuencia cardiaca



Fuente: Elaboración propia

Figura 7: Presionar un botón de la botonera



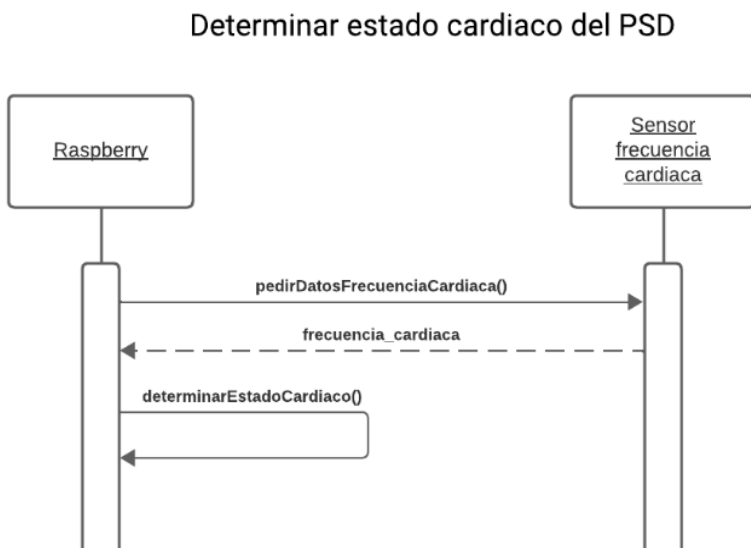
Fuente: Elaboración propia

Figura 8: Enviar mensaje de voz



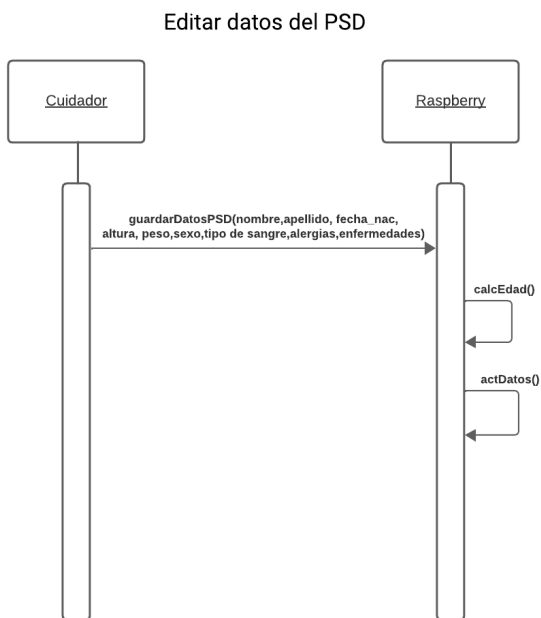
Fuente: Elaboración propia

Figura 9: Determinar estado cardíaco del PSD



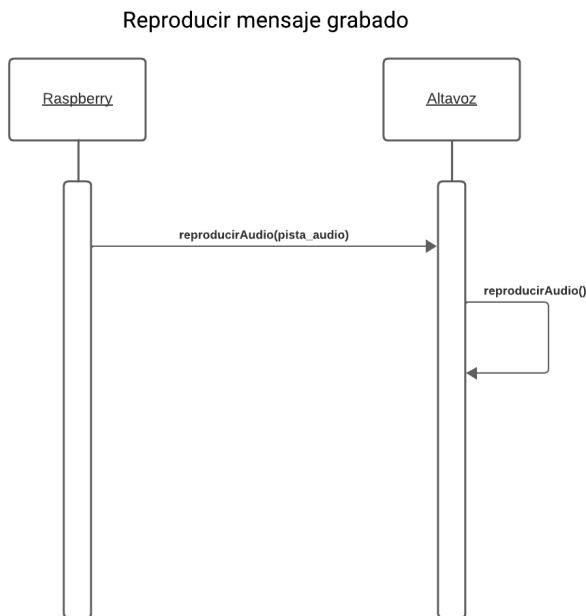
Fuente: Elaboración propia

Figura 10: Editar datos del PSD



Fuente: Elaboración propia

Figura 11 : Reproducir mensaje grabado



Fuente: Elaboración propia

5.1.4. Boceto Interfaz de usuario

Paleta de Colores

Se optó por una paleta de colores la cual su color principal es el azul, debido a que la tonalidades presentes poseen la característica de ser relajantes y/o transmitir confianza, según amplios estudios referentes a la sicología de los colores. Se optó por un color intenso de acento para comunicar aspectos importantes y necesariamente llamativos dada la índole de aplicación que se quiere construir.



Bocetos de pantallas principales

Se optó por un diseño muy sencillo, con pocos botones y pocos menús. Se decidió por este diseño meramente por la facilidad de navegación que otorga. La mayoría de los botones se encuentran cerca del pulgar derecho, facilitando el uso para aquellas personas con teléfonos móviles muy grandes o con dificultades de movimiento.

Los bocetos se encuentran en mayor detalle en las siguientes páginas.

Figura 12: Bocetos aplicación ALBA



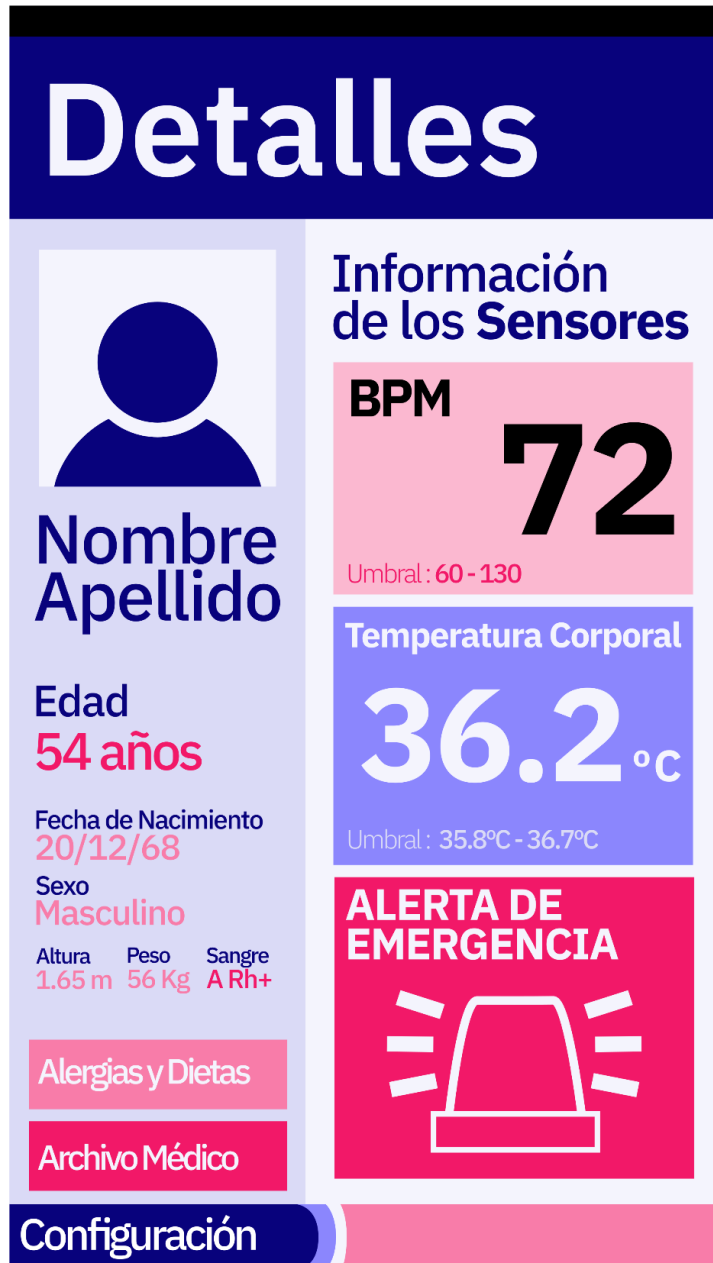
Fuente: Elaboración Propia

Figura 13: Inicio




Fuente: Elaboración propia

Figura 14: Detalles



Detalles



Nombre Apellido

Edad
54 años

Fecha de Nacimiento
20/12/68

Sexo
Masculino


Altura **Peso** **Sangre**
1.65 m 56 Kg A Rh+

Información de los Sensores

BPM
72
Umbral: 60 - 130

Temperatura Corporal
36.2 °C
Umbral: 35.8°C - 36.7°C

ALERTA DE EMERGENCIA



Configuración

Alergias y Dietas

Archivo Médico

Fuente: Elaboración propia

5.2. Herramientas y técnicas

Para poder programar en el Raspberry Pi, la lógica y tratamiento de datos utilizaremos:

- Python
- Visual Studio Code
- MySql

Para el lado de la app utilizaremos:

- React Native

Para obtener los datos de los sensores utilizaremos la librería RPi.GPIO, esta es una librería de python para poder obtener entradas de los dispositivos de entrada y salida de propósito general.

6. Implementación

Para llevar a cabo el proyecto, se implementó diferentes librerías, las cuales se detallarán a continuación

GPIOzero

Es una biblioteca para python que proporciona las herramientas necesarias para gestionar las entradas y salidas digitales del GPIO de la raspberry Pi

Opencv4

Es una librería de python encargada de dar soporte a las tareas relacionadas con el procesamiento de imagen y de visualización por computadora. Gracias a esta librería se puede observar en vivo al PSD.

Serial-async

Es una extensión de la biblioteca serial, esta agrega soporte para las operaciones asíncronas utilizando la biblioteca 'asyncio'.

Serial

Es una biblioteca base para la comunicación serial en python, proporciona una interfaz para interactuar con puertos serie y otros dispositivos de comunicación serial

websockets

Es una biblioteca de python que se utiliza para crear servidores y clientes WebSocket.

PulseSensorPlaygroundde

Es una biblioteca de arduino que permite medir el ritmo cardíaco y la frecuencia cardíaca de una persona. Este utiliza un sensor de pulso, como el Pulse Sensor, para detectar los cambios en el flujo sanguíneo que se producen con cada latido del corazón. Esta biblioteca se utiliza en arduino.

También se utilizaron diferentes técnicas y tecnologías para poder realizar la comunicación entre los componentes del sistema ALBA, alguno de estos fueron:

1. WebSocket

Según ionos: *“WebSocket es un protocolo de red basado en TCP que establece cómo deben intercambiarse datos entre redes. Puesto que es un protocolo fiable y eficiente, es utilizado por prácticamente todos los clientes. El protocolo TCP establece conexiones entre dos puntos finales de comunicación, llamados sockets. De esta manera, el intercambio de datos puede producirse en las dos direcciones.”* (“WebSocket | Un canal de comunicación para la web en tiempo real”)

Según lo anterior mostrado, la implementación en código se compone de: la aplicación móvil, servidor y recopilación de datos de los sensores. Se mostrará lo que se considera más importante para el sistema, las funcionalidades.

1. Aplicación Móvil

2. Servidor

Recibimiento de los datos del sensor de frecuencia cardiaca en la aplicación de dispositivos móviles

Figura 15: recibir datos

```
// Código para poder obtener los datos del sensor cardiaco enviados por el servidor
const [bpmSensor, setBPMSensor] = useState('');

useEffect(() => {
  obtenerRutaImagen();

  const ws = new WebSocket('ws://192.168.0.6:8090');

  ws.onopen = () => {
    console.log("Conexión abierta");
  };

  ws.onmessage = (event) => {
    const receiveData = JSON.parse(event.data);
    console.log(receiveData.type);
    if (receiveData.type === "bpm"){
      console.log(receiveData.bpm);
      setBPMSensor(receiveData.bpm);
    }
  };

  ws.onerror = (error) => {
    alert()
  };

  ws.onclose = () => {
    alertClose();
  };

  return () => {
    ws.close();
  };
}, []);
```

Fuente: Elaboración Propia

Recibimiento de los datos de la cámara

Figura 16: Recibir datos de cámara

```
useEffect(() => {
  const ws = new WebSocket('ws://172.20.10.4:8090');
  ws.onopen = () => {
    console.log("Conexión abierta");
  };
  ws.onmessage = (event) => {
    const receiveData = JSON.parse(event.data);

    if (receiveData.type == "video") {
      setImageData(`data:image/jpeg;base64,${receiveData.frameData}`);
    }
  };
  ws.onerror = (error) => {
    alert();
  };
  ws.onclose = () => {
    alertClose(),
    ws.close();
  };
  return () => {
    ws.close();
  };
}, []);
```

Fuente: Elaboración Propia

Se puede observar que se está utilizando websocket `const ws = new WebSocket('ws://172.20.10.4:8090')`, el código anterior crea una conexión WebSocket para recibir los datos del sensor de frecuencia cardiaca y de la cámara enviadas desde la raspberry pi.

Envío de los datos de frecuencia cardiaca mediante websocket

Figura 16: envío frecuencia cardiaca

```
async def bpm_feed(websocket):
    setted = False
    paused = False
    protocol: InputChunkProtocol

    while True:
        if not setted:
            try:
                serialCoroutine = serial_asyncio.create_serial_connection(
                    asyncio.get_running_loop(),
                    InputChunkProtocol,
                    PORT,
                    BAUD_RATE,
                    timeout=1
                )
                _, protocol = await serialCoroutine
            except:
                print('error connecting')
                return
            protocol.set_websocket(websocket)
            setted = True
        elif active["bpm"] and setted and paused:
            protocol.resume_reading()
        elif not active["bpm"] and setted and not paused:
            protocol.pause_reading()
        else:
            pass
```

Fuente: Elaboración propia

Envío de las imágenes de la cámara mediante websocket.

Figura 17: Envío de imágenes

```
async def video_feed(websocket):
    frame_rate = 30
    delay = 1 / frame_rate

    while active["video"]:
        frame = await generate_frames()
        await websocket.send(json.dumps({"frameData": frame, "type": "video"}))
        print("camera data sent")
        await asyncio.sleep(delay)

    print('stopped video')
    return True

active = {'bpm': False, 'video': False}
```

Fuente: Elaboración propia

Conexión entre raspberry y arduino mediante el servidor

Figura 17: conexión raspberry-arduino

```
puerto_serie = serial.Serial('/dev/ttyACM0', 115200, timeout=1)

try:
    while True:
        # Lee una línea de datos desde el puerto serie
        linea = puerto_serie.readline().decode('utf-8').strip()

        # Convierte la línea a un número entero (si es un número válido)
        try:
            valor_sensor = int(linea)
            print(f'Valor del sensor: {valor_sensor}')
        except ValueError:
            print('Error al convertir el valor a entero.')

        # Espera un segundo (ajusta según sea necesario)
        time.sleep(1)

except KeyboardInterrupt:
    print('Programa detenido manualmente.')

finally:
    # Cierra el puerto serie al salir
    puerto_serie.close()
```

3. Sensor

En los siguientes códigos se muestra la lógica para que el arduino reciba los datos del sensor cardiaco.

Figura 18: Lectura del sensor de frecuencia cardiaca

```
#define USE_ARDUINO_INTERRUPTS true
#include <PulseSensorPlayground.h>

// Variables
const int PulseWire = 0;      // Pulse
const int LED = LED_BUILTIN;  //
int Threshold = 572;          // Deter
                                // Use t
                                // Other

PulseSensorPlayground pulseSensor; //

void setup() {

  Serial.begin(9600);          // For S

  // Configure the PulseSensor object,
  pulseSensor.analogInput(PulseWire);
  pulseSensor.blinkOnPulse(LED);
  pulseSensor.setThreshold(Threshold);

  // Double-check the "pulseSensor" obj
  pulseSensor.begin();
}
```

Fuente: Elaboración propia

7. Comparativa de requerimientos

A continuación una tabla que evalúa los requerimientos que se han cumplido en el proceso de la implementación del sistema.

Tabla 24: Comparativa de requerimientos

Requerimiento	Cumplido	No Cumplido	Comentarios
Comunicación entre el PSD y el cuidador		x	No implementado, el sistema, a lo largo de la implementación, se centró en la monitorización.
Interfaz de usuario intuitivo	x		
Registro de mensajes		x	No implementado debido a la falta de conocimiento para enviar mensajes entre diferentes dispositivos.
Registro de los datos de los sensores	x		
Alerta al cuidador	x		
Registro de los datos del paciente	x		

Fuente: Elaboración propia

8. Trabajo a futuro

Como grupo creemos que el sistema ALBA posee un gran futuro para ser una herramienta enfocada en las personas especializadas en el cuidado de las personas en situación de dependencia. Esto debido a sus diferentes funcionalidades que se centran en la comunicación y monitoreo, la interfaz intuitiva permitirá al usuario utilizar con mayor facilidad la aplicación del sistema y los datos almacenados servirán de mucha ayuda para

Sin embargo, la falta de conocimiento y manejo de diferentes herramientas relacionadas con la programación (servidores, envío de datos, permisos en sistemas operativos, entre otros) y la gestión (compra de componentes, falta de implementos, entre otros), ha causado que la implementación del sistema no sea en su totalidad. A pesar de ello, se consideró las partes más importantes para el monitoreo como lo son la visualización en vivo de la persona y su frecuencia cardiaca, además del registro de datos.

9. Conclusión


Establecer límites y restricciones claros nos brinda claridad en el desarrollo de sistemas, tanto en programación como en investigación. Incluir objetivos de diseño es esencial para delimitar claramente el público objetivo del sistema, lo que permite ampliaciones o constricciones del mismo.

También se destaca la importancia de un sistema como este, que busca ser una herramienta para el cuidador y facilitar el área de trabajo con respecto a la monitorización y conocimiento del paciente (estado médico, alergias, entre otros).

En conclusión, el sistema ALBA cumple con uno de los objetivos principales relacionados con la monitorización del PSD, también se reconoce que faltarán funcionalidades por implementar. Sin embargo, el sistema ALBA promete ser un sistema eficiente de tal manera que cumpla con la mejoría del desempeño laboral, reduciendo la carga laboral y aumentando la buena comunicación entre cuidador y el PSD.

7. Referencias

1. Servicio Nacional de la Discapacidad [SENADIS], «Folleto Tercer Estudio Nacional de la Discapacidad - III ENDISC 2022», <https://www.senadis.gob.cl>, 2022. <https://www.senadis.gob.cl/descarga/i/7171/documento> (accedido 12 de septiembre de 2023).
2. R. P. Ltd, «Raspberry Pi OS – Raspberry Pi», *Raspberry Pi*. <https://www.raspberrypi.com/software/>
3. «Raspberry Pi Documentation - Raspberry Pi OS». <https://www.raspberrypi.com/documentation/computers/os.html>
4. 5. Paguayo, «Raspberry Pi 4 - Raspberry Pi», *Raspberry Pi*, sep. 2019, [En línea]. Disponible en: <https://raspberrypi.cl/raspberry-pi-4/>
6. «Termómetro infrarrojo». <https://mx.omega.com/prodinfo/termometro-infrarrojo.html#:~:text=Introducci%C3%B3n%20a%20los%20term%C3%B3metros%20infrarrojos,variaci%C3%B3n%20en%20la%20temperatura%20ambiente>
7. R. Medrán Medrán, «Medición del ritmo cardíaco mediante Fotopleletismografía», *e-reading «Trabajos y proyectos de estudios de la E.T.S.T.»*, 2018. <https://biblus.us.es/bibing/proyectos/abreproy/12461> (accedido 24 de septiembre de 2023).
8. «Aumento en la efectividad de la identificación de cimas y pies en el pulso fotopleletismográfico al reconstruirlo mediante filtrado adaptativo», *ScieELO*, [En línea]. Disponible en: https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-99402017000100061&lang=pt
9. SriTu Hobby, «How to use the heart Pulse Sensor with Arduino | Heart Pulse Monitoring System», *YouTube*. 3 de octubre de 2021. [En línea]. Disponible en: <https://www.youtube.com/watch?v=aKus0FV4deU>
10. A. Al-Qatatsheh *et al.*, «Blood pressure sensors: materials, fabrication methods, performance evaluations and future perspectives», *Sensors*, vol. 20, n.º 16, p. 4484, ago. 2020, doi: 10.3390/s20164484. <https://www.mdpi.com/1424-8220/20/16/4484>
11. K. Huang, F. Tan, T. Wang, y Y. Yang, «A highly sensitive Pressure-Sensing array for blood pressure estimation assisted by Machine-Learning techniques», *Sensors*, vol. 19, n.º 4, p. 848, feb. 2019, doi: 10.3390/s19040848. <https://www.mdpi.com/1424-8220/19/4/848>
12. «Ultrasound: accurate pulse wave velocity and blood pressure | IMEC». <https://www.imec-int.com/en/press/imecs-ultrasound-sensor-technology-yields-accurate-pulse-wave-velocity-and-blood-pressure>
13. M. Meusel *et al.*, «Measurement of blood pressure by Ultrasound—The applicability of devices, algorithms and a view in local hemodynamics», *Diagnostics*, vol. 11, n.º 12, p. 2255, dic. 2021, doi: 10.3390/diagnostics11122255. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8700406/>
14. N. Tamada, «US20160038117A1 - Ultrasonic blood pressure measuring device and blood pressure measuring Method - Google Patents», 11 de agosto de 2014. <https://patents.google.com/patent/US20160038117A1/en>

15. Waterloo Engineering, «Ultrasonic blood pressure sensor», *YouTube*. 1 de noviembre de 2017. [En línea]. Disponible en: <https://www.youtube.com/watch?v=cjEPz9xqllQ>
16. Enreta Domotica, « ADIOS a los SENSORES DE MOVIMIENTO - HOLA SENSOR DE PRESENCIA», *YouTube*. 16 de julio de 2023. [En línea]. Disponible en: <https://www.youtube.com/watch?v=1MXeFdh508Q>
17. “Salario para Programador en Chile - Salario Medio,” Talent.com. <https://cl.talent.com/salary?job=programador>
18. “Salario para Tecnico Informatico en Chile - Salario Medio,” Talent.com. <https://cl.talent.com/salary?job=tecnico+informatico>
19. “Salario para Diseñador en Chile - Salario Medio,” Talent.com. <https://cl.talent.com/salary?job=dise%C3%B1ador>
20. “Salario para Tecnico En Redes en Chile - Salario Medio,” Talent.com. <https://cl.talent.com/salary?job=tecnico+en+redes>
21. “Salario para Control Documental en Chile - Salario Medio,” Talent.com. <https://cl.talent.com/salary?job=control+documental>
22. *Sensor de temperatura GY-906 BAA Infrarrojo | Envío a todo Chile*. 2023. [En línea]. Disponible en: <https://www.mechatronicstore.cl/sensor-de-temperatura-gy-906-mlx90614esf/>
22. *Sensor de ritmo cardiaco y frecuencia cardiaca | Envío a todo Chile*. 2023. [En línea]. Disponible en: <https://www.mechatronicstore.cl/sensor-frecuencia-cardiaca/>
22. «Cámara Visión Nocturna y foco ajustable para Raspberry Pi Zero», *Altronics*. <https://altronics.cl/camara-ov5647-pi-zero-ir?search=camara>

7.1. Referencias Adicionales

“Sensores,” *National Institute of Biomedical Imaging and Bioengineering*. <https://www.nibib.nih.gov/espanol/temas-cientificos/sensores>