

**UNIVERSIDAD DE TARAPACÁ**  
*Universidad del Estado*



Departamento de Ingeniería en Computación e Informática



# **Plan de proyecto**

## **Control de asistencia**

### **biométrico**

Autor(es): Sebastián Cayupi

Manuel Lopez

Juan Meneses

Asignatura: Proyecto II

Profesor: Diego Arcena

ARICA, 03 Diciembre 2024

## Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
10/09/2024	1.0	Versión preliminar del formato	Sebastián Cayupi Manuel Lopez
10/09/2024	1.0.1	Se arregló errores denotados por el profesor	Sebastián Cayupi Manuel Lopez
24/09/2024	1.1	Se realizó la planificación de gestión de riesgos	Sebastián Cayupi Manuel Lopez Juan Meneses
05/11/2024	2.0	Se actualizó con los requerimientos de la Fase II	Sebastián Cayupi Manuel Lopez Juan Meneses
02/12/2024	3.0	Se actualizó con los requerimientos de la Fase III	Sebastián Cayupi Manuel Lopez Juan Meneses

# 1. Tabla de contenidos

1. Tabla de contenidos	3
2. Panorama General	4
2.1. Resumen del proyecto	4
2.1.1 Objetivo general	4
2.1.2. Objetivos específicos	4
2.2.3. Suposiciones y Restricciones	5
2.2.4. Entregables del proyecto	5
3. Organización del proyecto	6
3.1. Personal y entidades internas	6
3.2. Roles y Responsabilidades	6
3.3. Mecanismos de comunicación y organización	7
4. Planificación de los procesos de gestión	7
4.1. Planificación de recursos	7
4.1.1. Programación del proyecto	7
4.1.2. Electrónica del proyecto	9
4.1.3. Recursos Humanos	12
4.2. Costos totales	14
4.2.1. Recursos tangibles	14
4.2.2. Recursos intangibles	14
4.2.3. Tabla de costos totales	14
4.3. Distribución de tiempos	15
4.3.1. Asignación de tiempos	16
4.4. Planificación de la gestión de riesgos	16
5. Planificación de los procesos técnicos	18
5.1. Modelos de proceso	18
5.1.1. Requerimientos	18
5.1.2. Casos de uso	20
5.1.3. Interfaz de usuario	34
Bocetos de Pantallas Principales	34
5.2. Herramientas y técnicas	36
6. Conclusión	37
7. Referencias	38

## 2. Panorama General

### 2.1. Resumen del proyecto

El control de asistencia en las instituciones educativas sigue siendo un proceso que a menudo enfrenta problemas de precisión, eficiencia y carga administrativa. En la Universidad de Tarapacá, estos retos pueden afectar el enfoque de los docentes en la enseñanza y la gestión del tiempo en el aula. Este proyecto propone la implementación de un sistema de control de asistencia automatizado que utiliza tecnología biométrica y captura de imágenes para optimizar y automatizar el proceso de registro de asistencia, mejorando la precisión y reduciendo los errores humanos.

El proyecto “Control de Asistencia Biométrico” busca simplificar y hacer más eficiente la toma de asistencia, permitiendo que los profesores dediquen más tiempo a sus actividades académicas y reduciendo la carga administrativa. Además, ofrece una mayor fiabilidad al proceso de registro y ayuda a evitar posibles manipulaciones o errores en la asistencia manual.

#### 2.1.1 Objetivo general

El objetivo de este proyecto es el siguiente:

Desarrollar un sistema automatizado de control de asistencia basado en tecnología biométrica que permita registrar de manera precisa y eficiente la asistencia de los estudiantes, disminuyendo la carga administrativa de los profesores y aumentando la seguridad en el proceso de registro.

#### 2.1.2. Objetivos específicos

- Investigar y definir los recursos necesarios para la implementación del sistema.
- Diseñar el sistema de control de asistencia biométrica, asegurando su integración con la red de la universidad.
- Planificar las fases de desarrollo y pruebas del proyecto.
- Implementar el software y los dispositivos (Raspberry Pi, sensores de huellas dactilares, cámaras) para garantizar la funcionalidad del sistema.
- Realizar pruebas piloto del sistema para verificar su funcionamiento y hacer ajustes en base a los resultados obtenidos.

### 2.2.3. Suposiciones y Restricciones

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

Las suposiciones para este proyecto son las siguientes:

- Los estudiantes tendrán la capacidad y disposición para utilizar el sistema de control de asistencia biométrica sin inconvenientes.
- Los profesores estarán dispuestos a integrar el uso del sistema en sus clases y a adaptarse al nuevo método de registro de asistencia.
- Los recursos necesarios para construir e implementar el sistema, como sensores biométricos, Raspberry Pi y cámaras, estarán disponibles a tiempo.
- El equipo de proyecto cuenta con la experiencia y habilidades técnicas suficientes para desarrollar e implementar el sistema correctamente.

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:** El sistema está destinado a ser utilizado únicamente por estudiantes y profesores de la Universidad de Tarapacá.

### 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. Manual de usuario
8. Sistema "Sensor Biométrico"

## 3. Organización del proyecto

### 3.1. Personal y entidades internas

Entidades Internas	Persona encargada	Descripción
Jefe de Proyecto	Sebastian Cayupi	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	Manuel Lopez	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 Meneses	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

Los roles que tomarán cada integrante del equipo serán:

<b>Programador</b>	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.
<b>Técnico de hardware</b>	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.
<b>Diseñador</b>	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 atraktividad para el material audiovisual que genera.
<b>Técnico de redes</b>	Encargado de investigar sobre las herramientas y conocimientos necesarios para poder implementar una red y diseñar su estructura.

<b>Especialista en documentación</b>	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 del Sensor Biométrico 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.
- **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 <a href="#">Términos de licencia de Microsoft</a>
Lucidchart	Utilizado para diagramar y esquematizar interacciones con el sistema (usando UML)	Plan gratuito bajo los <a href="#">Términos de uso del Software Lucid</a>
Canva	Utilizado para diseñar diagramas demostrativos	Libre uso bajo los <a href="#">Términos de uso de Canva</a>
WhatsApp	Utilizado para la comunicación informal del grupo de trabajo	Libre uso bajo los <a href="#">Términos de servicio de WhatsApp</a>
Discord	Utilizado para la realización de reuniones remotas del grupo de trabajo	Libre uso bajo los <a href="#">Términos de servicio de Discord</a>
Redmine	Utilizado para planear y organizar tareas y documentos del proyecto	Proporcionado por la institución UTA
Unity 3D	Utilizado para la confección de la maqueta 3D y su ejemplificación de la idea a realizar	Software de libre disposición. Véase <a href="#">Términos de licencia</a>

## 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)

## 4.1.2. Electrónica del proyecto

Además del software, el hardware es fundamental para asegurar el funcionamiento del sistema de control de asistencia biométrica. A continuación, se detallan los componentes electrónicos que se usarán en este proyecto.

### Raspberry Pi 4

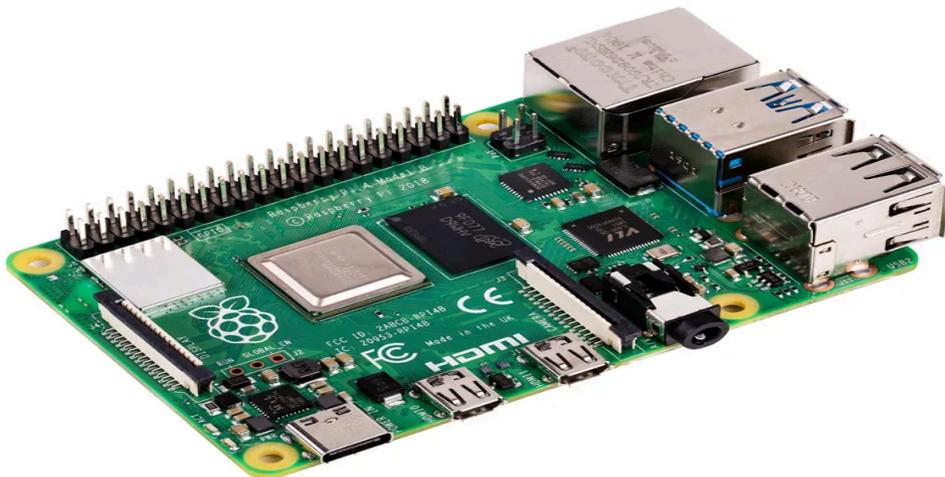
Es el componente central del sistema, actuando como el procesador principal que conecta y gestiona las comunicaciones entre los sensores biométricos y el software de control de asistencia. La Raspberry Pi 4 contiene el código necesario para la operación del sistema y facilita la interacción entre el usuario y el proceso de registro.

El proyecto utilizará el sensor biométrico **Adafruit Fingerprint**, que es esencial para la captura de huellas dactilares de los estudiantes al momento de ingresar al aula. Este sensor se integra fácilmente con la Raspberry Pi y permite un registro preciso y seguro de las huellas digitales.

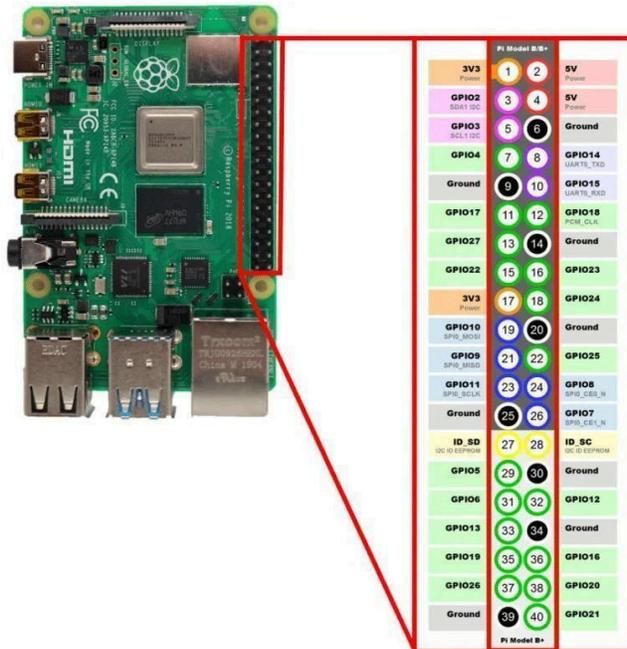
#### Características del sensor Adafruit Fingerprint:

**Precisión y fiabilidad:** Este sensor es conocido por su capacidad de reconocer huellas dactilares de forma rápida y precisa.

**Interfaz sencilla:** Se conecta a través de interfaces UART, lo que facilita su comunicación con la Raspberry Pi.



## Especificaciones técnicas



- 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\*)
- 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 la manipulación de datos biométricos, en este caso por medio de las huellas digitales.

Lista de sensores:

### 1. Considerados

- a. Sensor de huella digital
- b. Sensor de cámara
- c. pantalla

Sensor	Función	Descripción
Sensor de huella digital	Identificación de usuarios	Permite identificar a los usuarios mediante sus huellas dactilares. Brinda seguridad y precisión en la toma de asistencia.
Sensor de cámara	Supervisión de entorno	Proporciona una imagen visual del entorno para verificar asistencia y evitar suplantación de identidad.
Pantalla	Visualización de información	Muestra mensajes de confirmación de asistencia, errores, o el estado del sistema para el usuario.

### 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

Rol	Costo/hora	Personal que cumple el rol
Programador	\$5.292 CLP/hora	Sebastián Cayupi, Manuel Lopez, Juan Meneses
Técnico de hardware	\$3.077 CLP/hora	Sebastián Cayupi, Manuel Lopez
Especialista en documentación	\$2.769 CLP/hora	Juan Meneses

#### 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.

Tarea específica	Semanas	Horas Semanales	Horas Totales
Programación	9 sem.	6 hrs./sem.	54 hrs.
Preparación Hardware	11 sem.	3 hrs./sem.	33 hrs.
Documentar el proyecto	16 sem.	1.5 hrs./sem.	24 hrs.
<b>Total horas</b>		10.5 hrs./sem.	111 hrs.

Costo 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
Especialista en documentación	24 hrs.	3	72 hrs.	\$398.736 CLP
<b>Costo total recursos humanos</b>				\$1.449.240 CLP

## 4.2. Costos totales

### 4.2.1. 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	Sensor de huella digital	1 (p/sensor)	Un solo pago	~ \$20.000 CLP	~ \$45.000 CLP
	Sensor de cámara			~ \$5.000 CLP	
	Pantalla			~ \$20.000 CLP	

### 4.2.2. 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

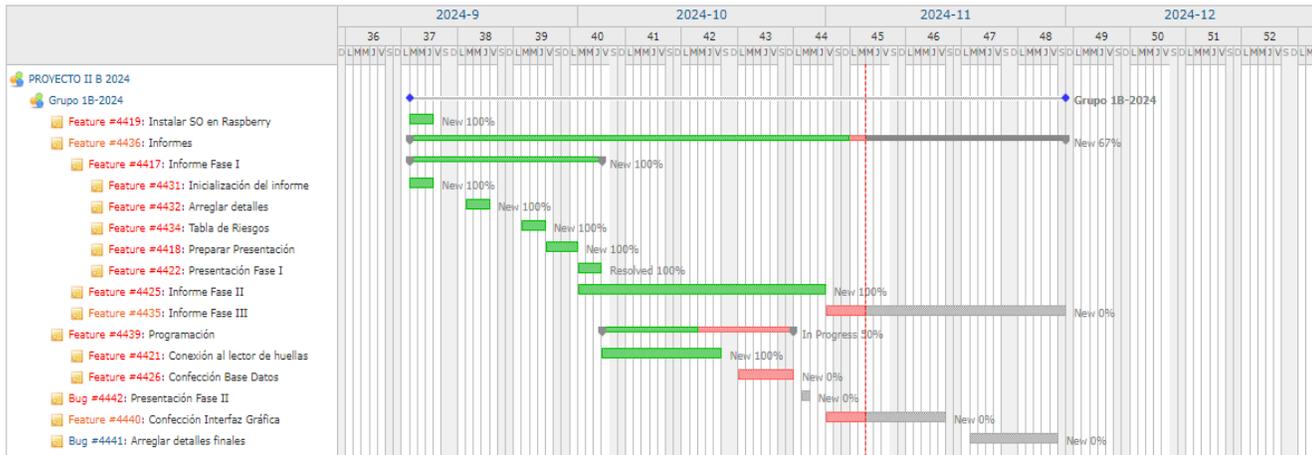
### 4.2.3. Tabla de costos totales

Elemento	Costo
Recursos tangibles	\$1.624.000 CLP
Recursos intangibles	\$34.950 CLP

Recursos humanos	\$1.449.240 CLP
<b>Costo proyecto total</b>	<b>\$3.108.190 CLP</b>

### 4.3. Distribución de tiempos

Se ha logrado de manera parcial los objetivos planteados con la carta gantt inicial, unos pequeños retrasos pero todo bastante realizable dentro de los tiempos.



### 4.3.1. 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.

### 4.4. Planificación de la gestión de riesgos

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

#	Riesgo	Probabilidad	Impacto
1	Personal sin experiencia.	60%	Crítico(2)
2	Mala planificación de tiempo	30%	Marginal(3)
3	Falta de participación de integrantes.	10%	Crítico(2)
4	Falta de componentes o problemas con el hardware.	25%	Crítico(2)
5	Abandono de proyecto por integrantes.	10%	Crítico(2)
6	Retraso en el reconocimiento de huellas.	50%	Crítico(2)
7	Pérdida de información.	5%	Catastrófico(1)
8	Pérdida de energía.	10%	Crítico(2)
9	Fallo en la conectividad de red.	10%	Crítico(2)
10	Incompatibilidad de hardware o software.	15%	Marginal(3)
11	Fallo al identificar huellas sucias, en mal estado, húmedas.	20%	Marginal(3)

Las acciones para remediar estas situaciones son las siguientes:

1. **Personal sin experiencia:** Estudiar componentes a usar y practicar su aplicación según el proyecto.
2. **Mala planificación de tiempo:** Identificar rol o roles que no cumpla con fecha de entrega y apoyar al rol.
3. **Falta de participación de integrantes:** Asignar trabajo particular al integrante y conversar la situación.
4. **Falta de componentes o problemas con el hardware:** Comprar componentes con anticipación y tener piezas de repuesto.
5. **Abandono de proyecto por integrantes:** Redistribuir las tareas entre el equipo restante e intentar adelantar lo posible
6. **Retraso en el reconocimiento de huellas:** Optimizar el código de reconocimiento y ajustar tiempos de procesamiento
7. **Pérdida de información:** Implementar backups automáticos y usar sistemas de control de versiones.
8. **Pérdida de energía:** Implementar una batería de respaldo para la Raspberry Pi.
9. **Fallo en la conectividad de red:** Verificar la red y el cableado.
10. **Incompatibilidad de hardware o software:** Verificar la compatibilidad antes de comprar componentes y hacer pruebas tempranas de funcionalidad.
11. **Fallo al identificar huellas sucias, en mal estado, húmedas.:** Mantener las recomendaciones para usar de óptima forma el sensor, pegar en la caja (secar siempre las manos antes de usar, limpiarse los dedos con agua, alcohol)

## 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.

##### **Requerimientos funcionales:**

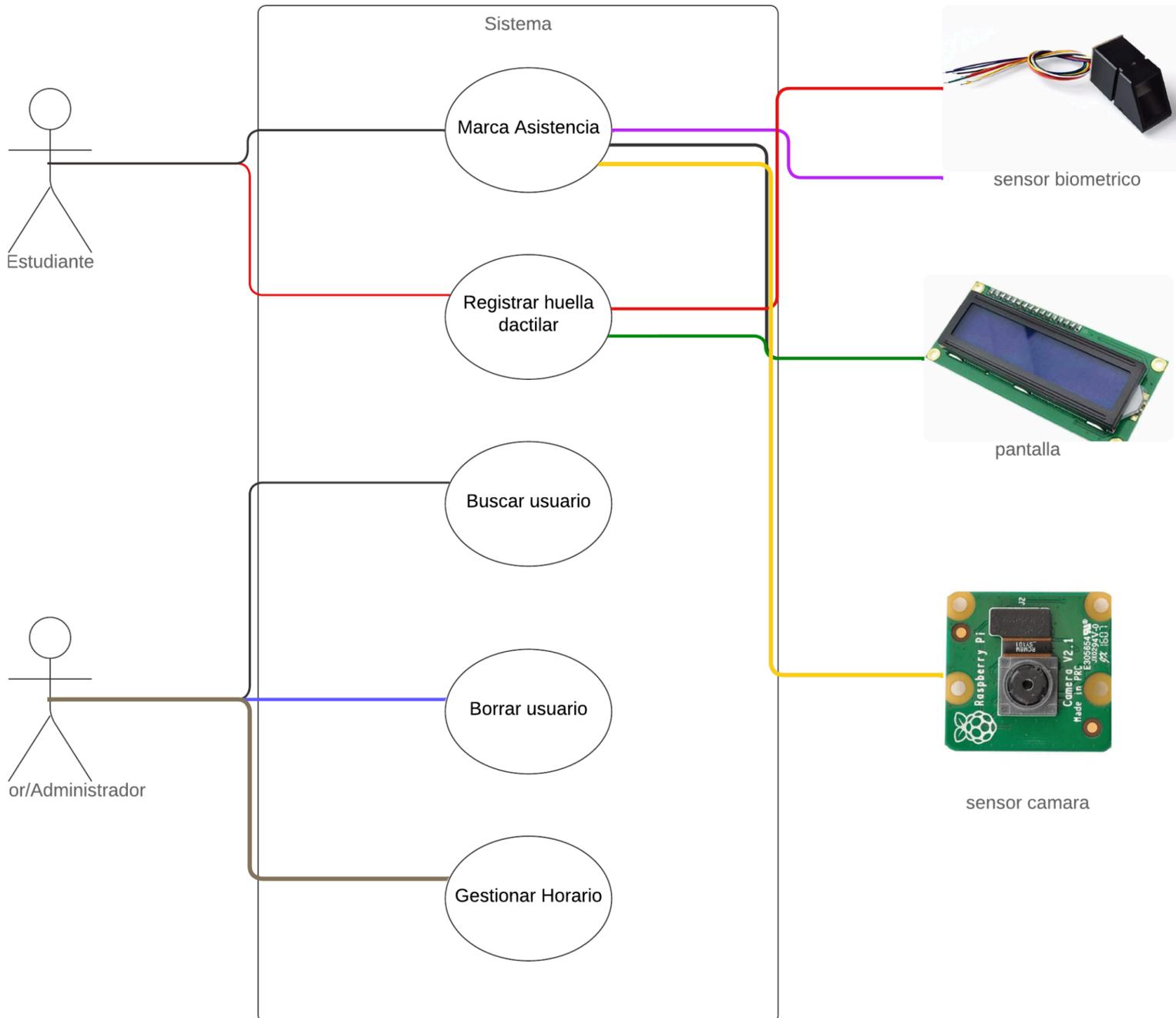
- El sistema debe capturar y registrar la asistencia de los estudiantes mediante el uso de un sensor de huellas dactilares.
- El software debe procesar y enviar los datos a la intranet de la universidad para mantener un registro actualizado.
- El sistema debe incluir una función de captura de imágenes para respaldar la autenticidad del registro de asistencia.
- Debe contar con una interfaz de usuario amigable para la gestión de registros por parte de los docentes.
- Debe permitir la revisión y exportación de los registros de asistencia en formato digital.

## Requerimientos no funcionales

- **Seguridad:** Los datos biométricos deben ser almacenados y transmitidos de forma cifrada para proteger la privacidad de los usuarios.
- **Usabilidad:** La interfaz debe ser intuitiva y fácil de usar tanto para estudiantes como para profesores.
- **Escalabilidad:** El sistema debe ser capaz de integrarse y adaptarse a nuevas funcionalidades o dispositivos si se requiere en el futuro.
- **Rendimiento:** El tiempo de respuesta del sistema debe ser inferior a 2 segundos por registro de asistencia.
- **Compatibilidad:** Debe ser compatible con la infraestructura tecnológica existente de la universidad, como la red y las bases de datos.

### 5.1.2. Casos de uso

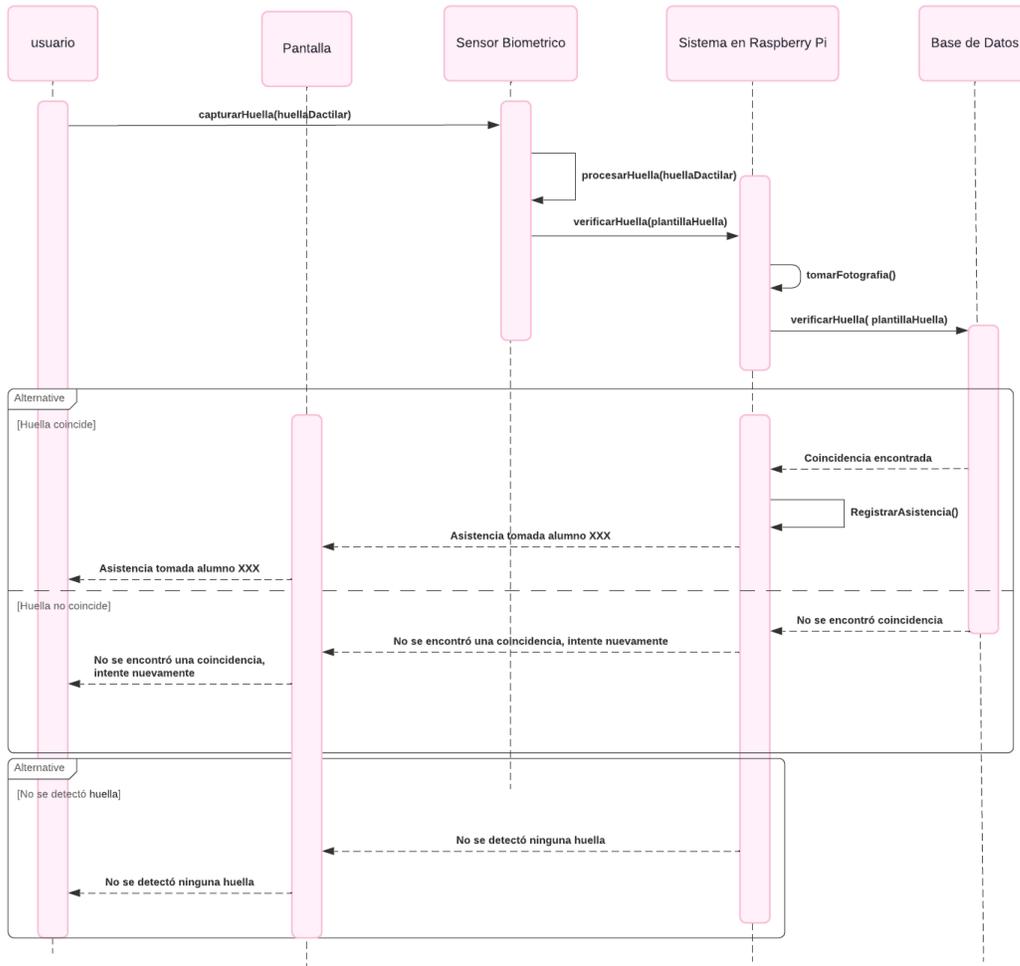
#### Diagrama de casos de uso general



(caso de uso general img Uso 01)

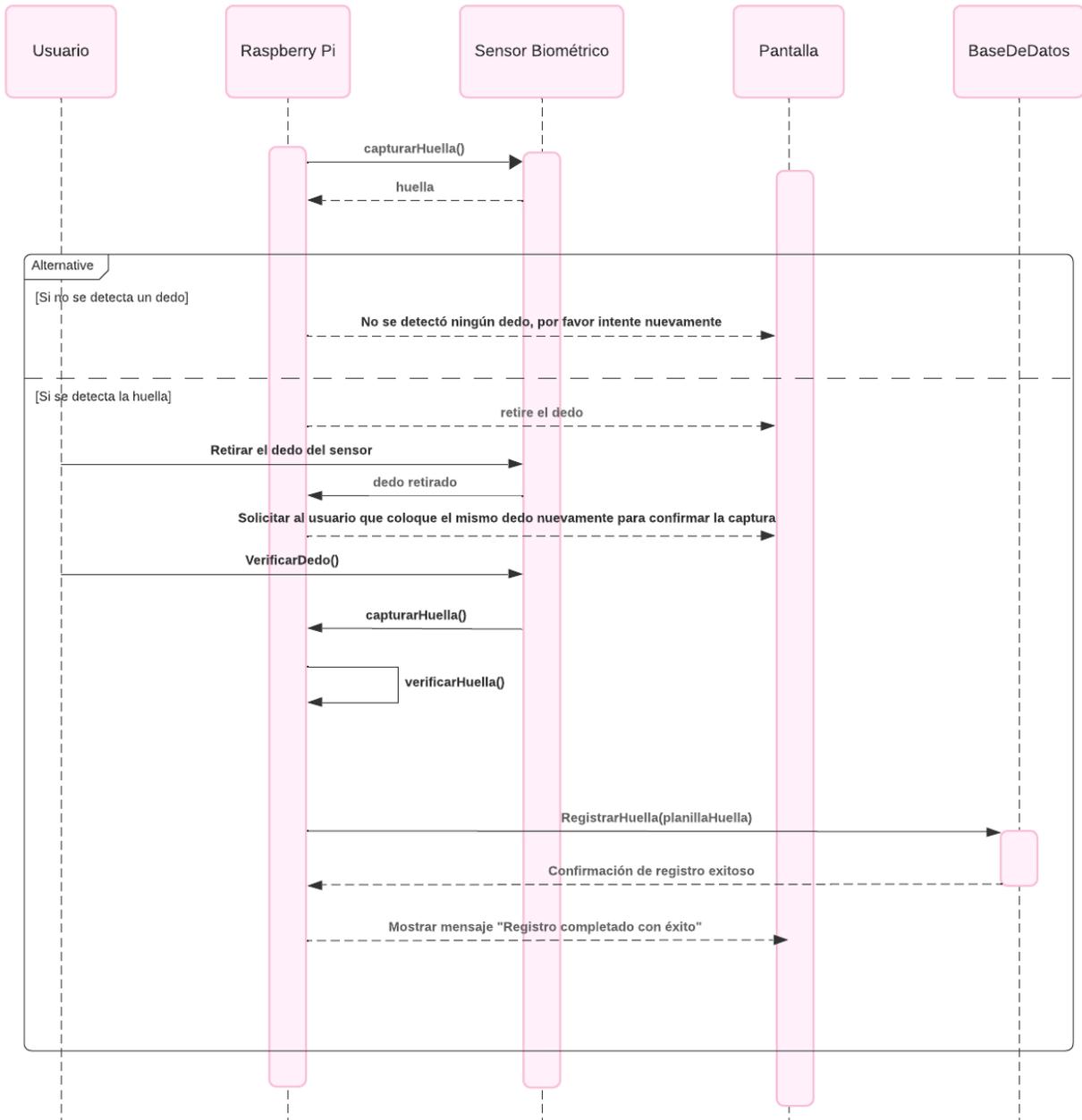
Descripción de casos de uso.

Nombre cus: Marcar Asistencia	
Descripción: Permite registrar la huella dactilar de un estudiante en el sistema, utilizando un sensor biométrico, para que pueda ser identificado en el sistema de asistencia.	
Actor: Estudiante, profesor/administrador, sensor biométrico, pantalla, camara sensor	
Precondición: <ol style="list-style-type: none"> <li>1. El estudiante debe ser alumno regular del sistema.</li> <li>2. El estudiante debe contar con una huella digital ingresada en el sistema.</li> </ol>	
Flujo Principal: Actor <ol style="list-style-type: none"> <li>1. Ingresa su huella en el sensor.</li> </ol>	Flujo Principal: Sistema <ol style="list-style-type: none"> <li>2. se muestra en pantalla "asistencia tomada alumno XXX"</li> </ol>
Flujo Alternativo:	<ol style="list-style-type: none"> <li>2.1. Se muestra un mensaje en pantalla: "No se detectó ningún dedo, por favor intente nuevamente"</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>2.2. muestra un mensaje en pantalla : " No se encontró una coincidencia, por favor intente nuevamente."</li> </ol>
Postcondiciones:La asistencia del usuario queda registrada en el sistema correctamente.	
Valor medible:Reduce el tiempo usado para tomar la asistencia en clases.	



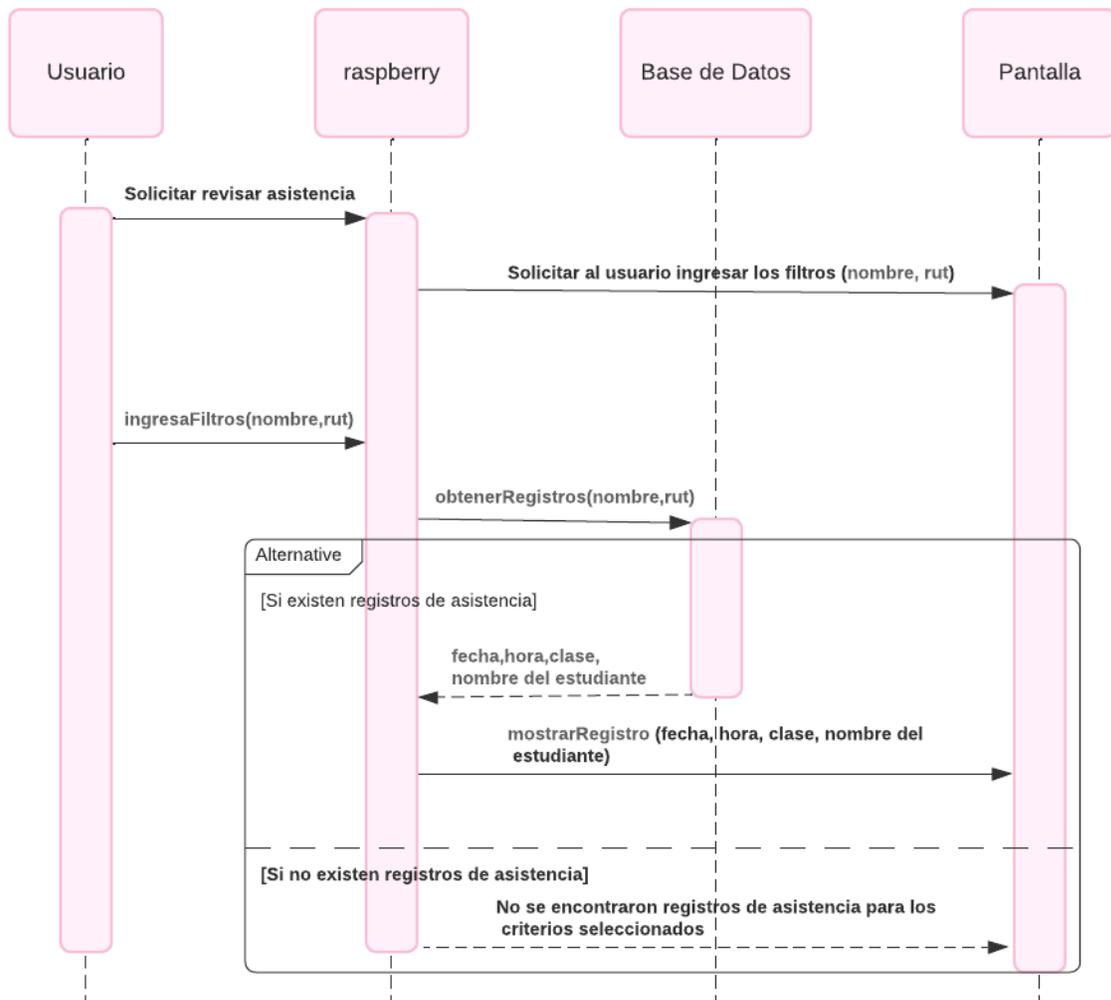
CASO DE USO (Marcar asistencia)

Nombre CUS: Registrar huella dactilar.	
Descripción: Permite al estudiante registrar su huella dactilar en el sistema.	
Actor: Estudiante, sensor biométrico, pantalla	
Precondición	
<p>Flujo Principal: Actor</p> <ol style="list-style-type: none"> <li>1. El usuario pone el dedo en el sensor.</li>      <li>4. El usuario quita el dedo del sensor.</li>      <li>6. El usuario ingresa el mismo dedo nuevamente en el sensor.</li> </ol>	<p>Flujo Principal: Sistema</p> <ol style="list-style-type: none"> <li>2. El sistema captura la imagen de la huella dactilar.</li> <li>3. El sistema solicita al usuario que retire el dedo.</li>     <li>5. El sistema solicita al usuario que coloque el mismo dedo nuevamente para confirmar la captura.</li>     <li>7. El sistema registra la huella en el sistema.</li> </ol>
Flujo Alternativo: "Si no se detecta un dedo"	<ol style="list-style-type: none"> <li>2.1. El sistema muestra un mensaje en la pantalla: "No se detectó ningún dedo, por favor intente nuevamente."</li> </ol>
Postcondiciones: La huella del usuario queda registrada en el sistema.	
Valor medible: Registra en el sistema los futuros usuarios de este.	



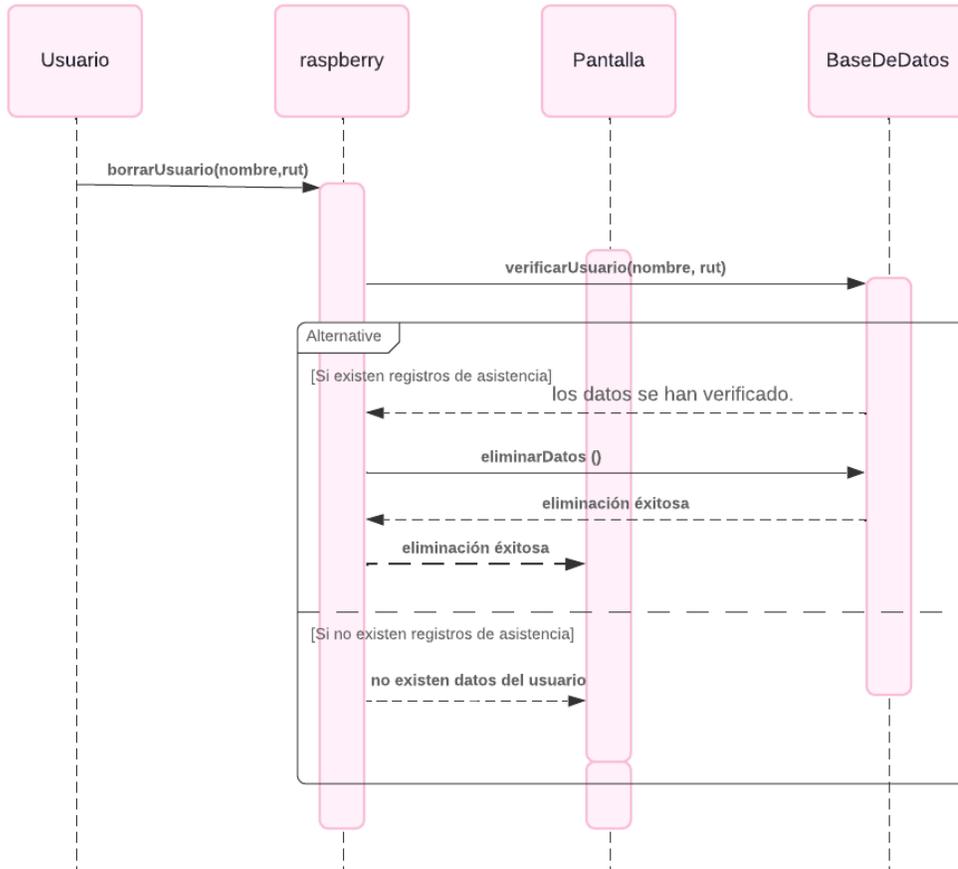
CASO DE USO (Registrar Huella Dactilar)

Nombre CUS: Buscar Usuario	
Descripción: permite al usuario revisar los detalles de la asistencia.	
Actor: profesor, administrador.	
Precondición: el sistema debe mantener registros válidos de asistencia previos para que el usuario pueda consultarlos.	
<p>Flujo Principal: Actor</p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de ver asistencia en el sistema.</li>   <li>3. El usuario ingresa los filtros.</li> </ol>	<p>Flujo Principal: Sistema</p> <ol style="list-style-type: none"> <li>2. El sistema solicita filtro para los datos(nombre,rut).</li>   <li>3. El sistema muestra los registros de asistencia al usuario, incluyendo información detallada..</li> </ol>
Flujo Alternativo:	<ol style="list-style-type: none"> <li>3.1. El sistema muestra un mensaje: "No se encontraron registros de asistencia para los criterios seleccionados."</li> </ol>
Postcondiciones:	
Valor medible: Visualizar la información sobre las clases y la asistencia a ellas.	

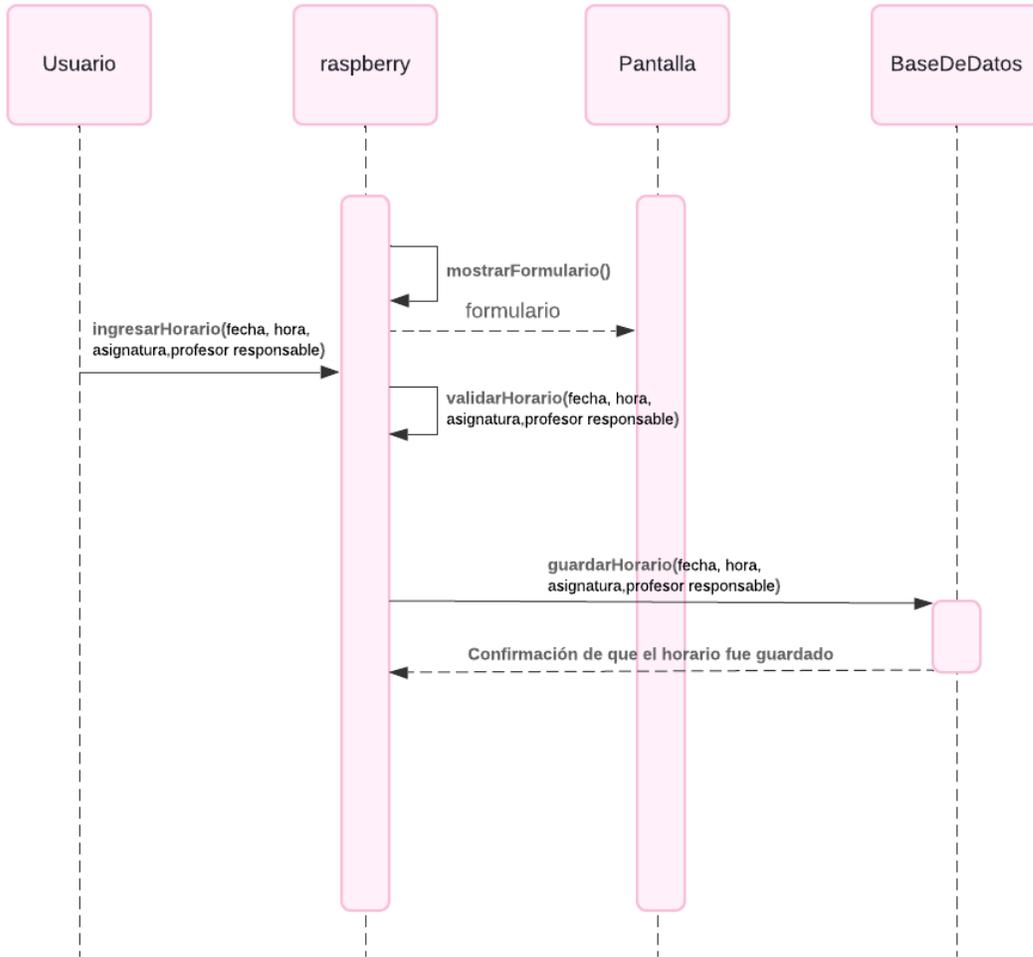


CASO DE USO (Buscar Usuario)

Nombre CUS: Borrar usuario	
Descripción: Permite borrar el usuario ingresando nombre y rut	
Actor: profesor, administrador.	
Precondición: Debe existir registros de asistencia en el sistema.	
<p>Flujo Principal: Actor</p> <ol style="list-style-type: none"> <li>1. El usuario ingresa el nombre y RUT del usuario a borrar.</li> </ol>	<p>Flujo Principal: Sistema</p> <ol style="list-style-type: none"> <li>2. El sistema verifica si existen registros de asistencia asociados al usuario</li> <li>3. El sistema confirma la eliminación exitosa del usuario</li> </ol>
<p>Flujo Alternativo:</p> <p>No se encuentra datos del usuario</p>	<ol style="list-style-type: none"> <li>3.1. No existen datos del usuario.</li> </ol>
Postcondiciones: se actualizó la base de datos del sistema al eliminar un usuario	
Valor medible: Verificar si el usuario ha sido eliminado correctamente	

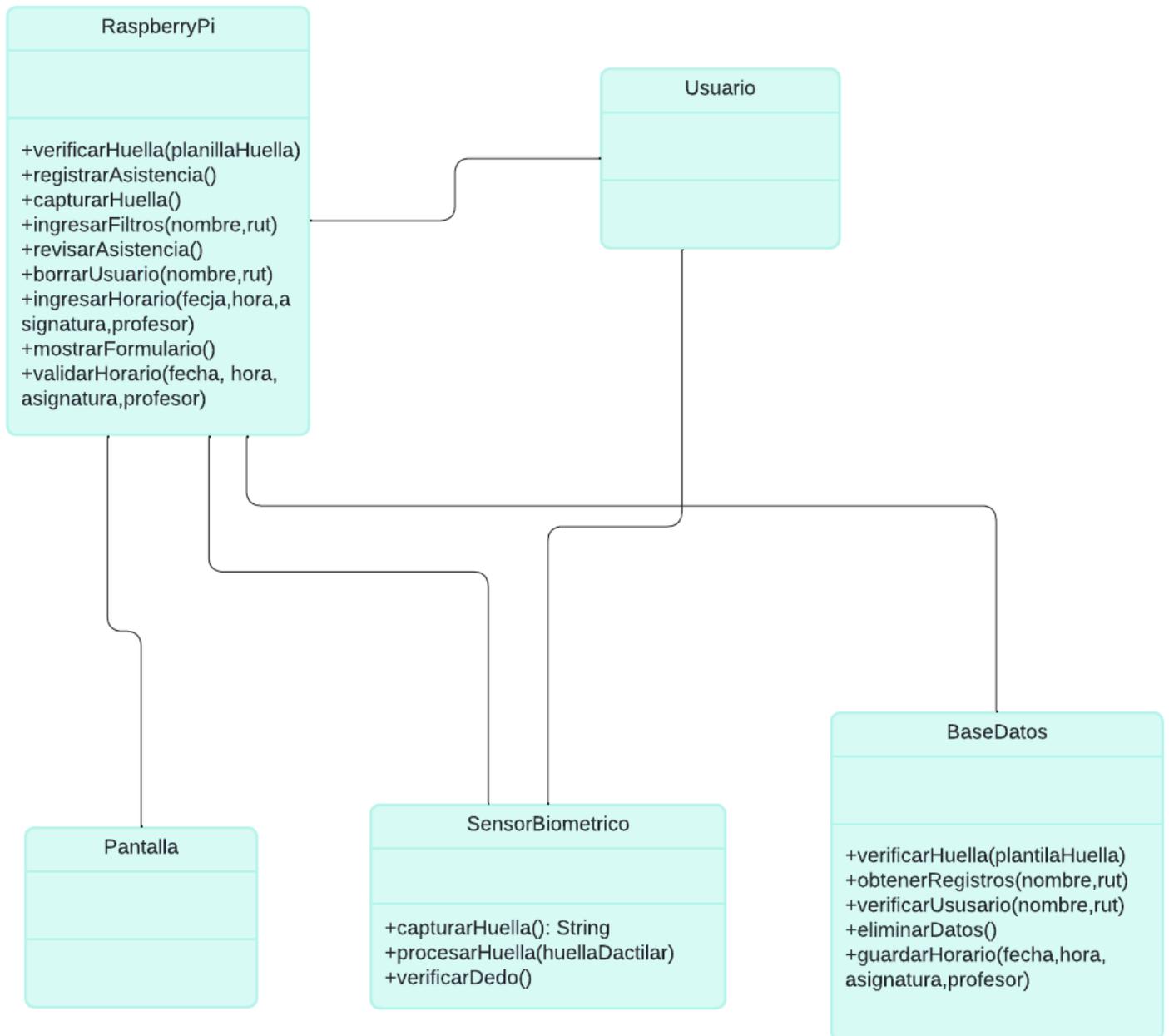


Nombre CUS: Gestionar horario	
Descripción: Permite al administrador gestionar el horario de clases, estos nos permiten saber a qué clase se toma la asistencia.	
Actor: administrador, profesor.	
Precondición:	
<p>Flujo Principal: Actor</p> <ol style="list-style-type: none"> <li>1. El actor ingresa la información del horario de clases (fecha, hora, asignatura, profesor responsable).</li> </ol>	<p>Flujo Principal: Sistema</p> <ol style="list-style-type: none"> <li>1. El sistema muestra el formulario para ingresar los datos del horario de clases</li> <li>2. El sistema valida la información ingresada (verifica formato y que no haya conflictos de horarios)</li> <li>4. El sistema guarda el horario en la base de datos.</li> </ol>
Flujo Alternativo:	
Postcondiciones: Queda guardado en el sistema el horario de clases.	
Valor medible: El tiempo promedio para gestionar y guardar un nuevo horario no debe superar los 5 minutos.	



CASO DE USO (Gestionar Horario)

## Diagrama de clases



### 5.1.3. Interfaz de usuario

Logo del proyecto :



#### Bocetos de Pantallas Principales

El diseño de las pantallas del sistema de control de asistencia biométrica se desarrolló como prototipos de bajo nivel, con un enfoque minimalista para facilitar la navegación de estudiantes y profesores. Se prioriza la simplicidad y la accesibilidad, colocando los botones cerca del borde derecho para un uso cómodo.

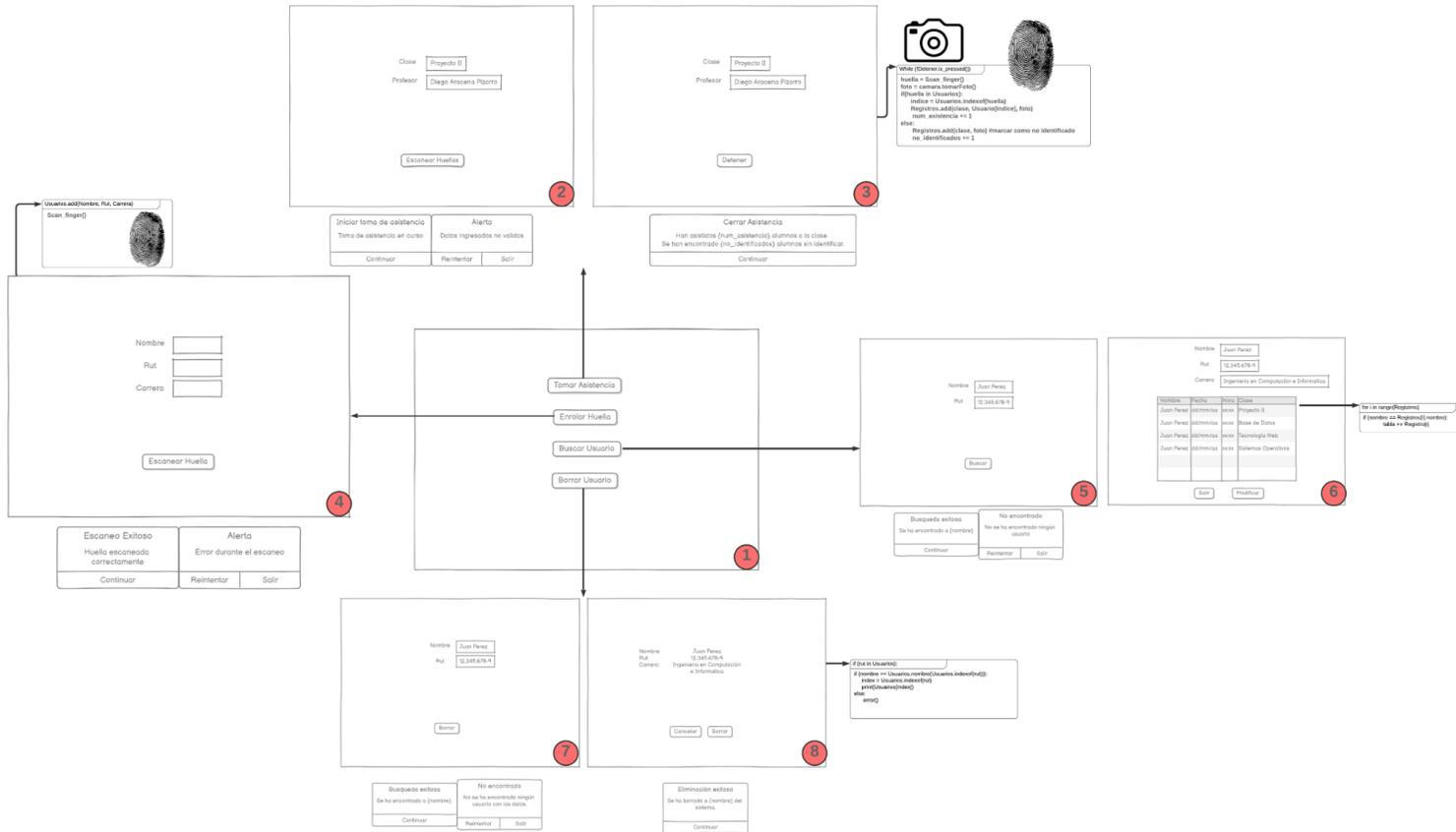
Bocetos de Pantallas:

1. Pantalla de Inicio: Acceso rápido a "Registrar Asistencia", "Revisar Asistencia" y "Configuración".
2. Pantalla de Registro de Asistencia: Área para escaneo de huella, indicador de progreso y mensajes de confirmación.
3. Pantalla de Confirmación: Mensaje de éxito con datos del registro y botón para volver al inicio.
4. Pantalla de Configuración: Opciones de personalización y acceso a soporte.

Estos prototipos aseguran una interacción intuitiva y rápida, permitiendo optimizar la experiencia de uso antes del desarrollo detallado.

Prototipo:

Se realiza un prototipo de bajo nivel del proyecto para lograr visualizar correctamente la idea a futuro para facilitar la comprensión de la estructura y flujo de ventanas para los miembros del equipo. Además de identificar problemas o mejoras en la interfaz para el momento de su realización.



1. Menú principal: con sus botones respectivos para navegar a las demás ventanas.
2. Tomar Asistencia: se le consulta al usuario el nombre de la clase y el profesor a tomar asistencia.
3. Durante esta ventana, se almacenarán todas las huellas entrantes al sensor realizando un match entre la huella recibida y las almacenadas en el sistema en conjunto a una fotografía de la persona, en caso de no identificar al usuario se marcará para una futura revisión manualmente.
4. Enrolar Huella: Se solicitará nombre, rut y carrera del usuario al momento de presionar “Escanear Huella” el sistema esperará la respuesta del sensor y escanea la huella dos veces para comprobar la correcta lectura de la huella comparando ambos intentos y procederá a almacenar la información del usuario.
5. Buscar Usuario: Se solicitará nombre y rut del usuario y al presionar el botón “Buscar” se buscará al usuario dentro del sistema para proceder con la siguiente ventana.
6. Se mostrará la información del usuario y se desplegará una tabla con todas sus ocurrencias dentro del sistema.
7. Borrar Usuario: Se solicitará nombre y rut del usuario y al presionar el botón “Borrar” se buscará al usuario dentro del sistema para proceder con la siguiente ventana.
8. Se mostrará la información del usuario y dos botones, “Cancelar” para no realizar ninguna acción y uno “Borrar” para eliminar al Usuario del sistema.

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

- Descripción de la Arquitectura con respecto a los modelos?????

# Implementación

## Plan de Implementación

### Plan de Integración

Para el plan de implementación e integración del sistema de control de asistencia biométrico,

mediante la identificación de huellas dactilares, se quiere aumentar la integridad de los datos y disminuir el tiempo que se requiere para la toma de asistencia en clases.

Se debe llevar a cabo las siguientes fases para la implementación del del sistema biométrico:

### Integración del hardware y software

**Raspberry pi:** La Raspberry Pi 4 actúa como cerebro del sistema, está conectado y gestiona la Comunicación entre el sensor biométrico, la cámara, la pantalla y el software de control. Esta debe estar siempre encendida para garantizar la disponibilidad del sistema.

**Sensores Biométricos, Cámara y pantalla:** El sensor de huella biométrica se integra a un adaptador USB A TTL mientras que la cámara se conecta directamente al puerto USB sin la necesidad de un adaptador, para la pantalla se utilizó un grove pi + v del 2015 el cual se

conecta a los pines GPIO. La cámara será utilizada para capturar imágenes y validar la asistencia en caso de inconsistencia.

**Aislamiento de Componentes:** para el buen funcionamiento del sistema y evitar fallas de humedad, roces , toques, el proyecto será encapsulados en una caja impresa en 3d la cual se planea poner en la pared de la sala.

### **Pruebas de Funcionamiento**

**Prueba de Hardware:** Cada uno de los componentes electrónicos (Raspberry Pi, sensores de huellas, cámara) será sometido a pruebas individuales para asegurar su funcionamiento. La Raspberry Pi se probará junto con los sensores de huellas para verificar la correcta identificación de los usuarios.

**Pruebas de Software:** Se realizarán pruebas del software para verificar el correcto registro de asistencia, la comunicación entre la interfaz de usuario y los sensores biométricos, y el envío de datos al sistema.

### **Pruebas de Integración Finales**

**Pruebas Piloto:** El sistema será desplegado para una prueba piloto en aulas seleccionadas de la Universidad de Tarapacá, donde se evaluará su eficiencia y precisión en condiciones reales. Se realizarán ajustes en base al feedback recibido de los usuarios, para optimizar la experiencia de uso.

Este plan de integración tiene como fin asegurar que todos los componentes del sistema funcionen de manera armónica, garantizando que la toma de asistencia biométrica sea una experiencia intuitiva, eficiente y segura para estudiantes.

## Módulos Implementados

El sistema de control de asistencia biométrica se estructuró en una serie de módulos que garantizan un funcionamiento eficiente y una implementación escalable. Cada módulo fue diseñado para cumplir funciones específicas dentro del sistema, optimizando la interacción entre hardware y software:

### 1. Módulo de Captura de Huellas

- Este módulo gestiona la interacción con el sensor biométrico Adafruit Fingerprint, permitiendo el registro y la lectura de huellas digitales.
- Integra librerías específicas para la comunicación serial con la Raspberry Pi, asegurando una captura precisa y rápida.

```
import time

import adafruit_fingerprint

import serial

# Configuración del puerto serial para el sensor de huellas
uart = serial.Serial("/dev/ttyUSB0", baudrate=57600, timeout=1)

finger = adafruit_fingerprint.Adafruit_Fingerprint(uart)

def get_fingerprint():

    """Obtiene una imagen de la huella, la convierte en una plantilla y
    verifica si coincide."""

    while finger.get_image() != adafruit_fingerprint.OK:

        pass

    if finger.image_2_tz(1) != adafruit_fingerprint.OK:

        return False
```

```
if finger.finger_search() != adafruit_fingerprint.OK:

    return False

return True
```

## 2. Módulo de Registro de Usuarios

- Permite registrar a nuevos usuarios en el sistema, asociando sus datos personales con su huella digital y su RUT.

```
def enroll_finger_with_details():

    """Solicita detalles del usuario antes de enrolar la huella."""

    nombre = nombre_entry.get().strip()

    apellido = apellido_entry.get().strip()

    carrera = carrera_entry.get().strip()

    if not nombre or not apellido or not carrera:

        messagebox.showerror("Error", "Todos los campos son obligatorios.")

        return

    location = int(id_entry.get())

    if location < 1 or location > 127:

        messagebox.showerror("Error", "El ID debe estar entre 1 y 127.")

        return

    if enroll_finger(location):

        usuarios[location] = {"nombre": nombre, "apellido": apellido,
"carrera": carrera}

        messagebox.showinfo("Éxito", f"Huella enrolada: {nombre} {apellido},
```

```
Carrera: {carrera}")

    else:

        messagebox.showerror("Error", "Error en enrolar huella")
```

### 3. Módulo de Verificación y Autenticación

- Este módulo valida las huellas digitales ingresadas comparándolas con las registradas previamente en el sistema.
- En caso de coincidencia, genera una respuesta positiva que permite el acceso o registro de asistencia.

```
def buscar_huella():

    """Busca una huella y muestra los datos del usuario asociado."""

    if get_fingerprint():

        user_id = finger.finger_id

        if user_id in usuarios:

            datos = usuarios[user_id]

            print(f"Huella ID {user_id}: {datos['nombre']} {datos['apellido']}, Carrera: {datos['carrera']}")

        else:

            print("ID no encontrado en la base de datos")

    else:

        print("Huella no encontrada")
```

### 4. Módulo de Control de Asistencia

- Registra la fecha y hora de cada usuario que se autentica exitosamente en el sistema.

- Estos registros se almacenan en un archivo JSON, organizados para facilitar la consulta y generación de reportes.

```
def registrar_asistencia(user_id):  
  
    """Registra la fecha y hora de asistencia del usuario."""  
  
    fecha_hora = time.strftime("%Y-%m-%d %H:%M:%S")  
  
    with open("asistencia.json", "a") as file:  
  
        file.write(f'{"id": {user_id}, "fecha_hora": "{fecha_hora}"}\n')
```

## 5. Módulo de Interfaz de Usuario

- Proporciona una interfaz sencilla para que los usuarios puedan interactuar con el sistema, tanto para el registro como para la verificación de huellas.
- Diseñada siguiendo los bocetos definidos en las fases iniciales del proyecto, asegurando una experiencia de usuario intuitiva.

```
import tkinter as tk  
  
from tkinter import messagebox  
  
usuarios_enrolados = []  
  
def procesar_enrolamiento():  
  
    nombre = entrada_nombre.get().strip()  
  
    rut = entrada_rut.get().strip()  
  
    carrera = combo_carreras.get()  
  
    if nombre and rut and carrera:  
  
        if any(usuario['rut'] == rut for usuario in usuarios_enrolados):  
  
            messagebox.showerror("Error", "Ya existe un usuario con ese RUT")  
  
            return  
  
        usuarios_enrolados.append({"nombre": nombre, "rut": rut, "carrera":  
carrera})  
  
        messagebox.showinfo("Éxito", "Usuario enrolado correctamente")
```

```
ventana_enrolar.destroy()

else:

    messagebox.showerror("Error", "Complete todos los campos")

ventana = tk.Tk()

ventana.title("Sistema de Asistencia")

ventana.geometry("400x500")

tk.Label(ventana, text="Nombre:").pack()

entrada_nombre = tk.Entry(ventana)

entrada_nombre.pack()

tk.Label(ventana, text="RUT:").pack()

entrada_rut = tk.Entry(ventana)

entrada_rut.pack()

tk.Label(ventana, text="Carrera:").pack()

combo_carreras = tk.StringVar()

combo_carreras.set("Selecciona una carrera")

tk.OptionMenu(ventana, combo_carreras, "Ingeniería", "Medicina",
"Derecho").pack()

boton_iniciar = tk.Button(ventana, text="Iniciar Enrolamiento",
command=procesar_enrolamiento)

boton_iniciar.pack(pady=20)

ventana.mainloop()
```

## 6. Módulo de Reportes

- Genera reportes detallados de asistencia, clasificando los datos por usuario o fecha.
- Estos reportes pueden ser exportados para análisis externo, facilitando su integración en

otros sistemas.

```
def generar_reporte():  
  
    """Genera un reporte de asistencia."""  
  
    with open("asistencia.json", "r") as file:  
  
        registros = file.readlines()  
  
        for registro in registros:  
  
            print(registro.strip())
```

Este enfoque modular permite un desarrollo ordenado y eficiente, superando los desafíos encontrados durante el proceso y asegurando la entrega de un sistema funcional y confiable.

## Reporte de Revisión

### Implementación del sensor de huellas

La primera función implementada fue la interacción con el sensor Adafruit Fingerprint. Se realizaron pruebas para registrar, almacenar y autenticar huellas, logrando finalmente su correcto funcionamiento tras ajustar librerías y configuraciones en la Raspberry Pi.

### Registro de usuarios y asistencia

Se desarrolló la funcionalidad para registrar usuarios y almacenar su información, junto con los registros de asistencia, en un archivo JSON. Aunque al inicio se detectaron duplicados, se implementaron validaciones que corrigieron este problema.

### Interfaz gráfica

Se diseñó una interfaz básica que permite registrar y verificar huellas de manera intuitiva. Este módulo fue probado y ajustado para garantizar una experiencia de usuario sencilla y funcional.

### Integración de cámara

Se implementó una cámara conectada a la Raspberry Pi para capturar una fotografía cada vez que un usuario escanea su huella. Esto permitió agregar una capa adicional de registro visual al sistema. Inicialmente, hubo problemas con el reconocimiento y la configuración de la cámara, pero estos fueron resueltos tras pruebas y ajustes en los controladores.

### Uso de GrovePi y pantalla de texto

Se integró un módulo GrovePi con una pantalla para mostrar mensajes sobre el progreso del escaneo de huellas y otros estados del sistema. La pantalla fue configurada para mostrar notificaciones como "Escaneando huella", "Huella registrada" y "Error en el escaneo". Esta adición mejoró significativamente la interacción con el usuario.

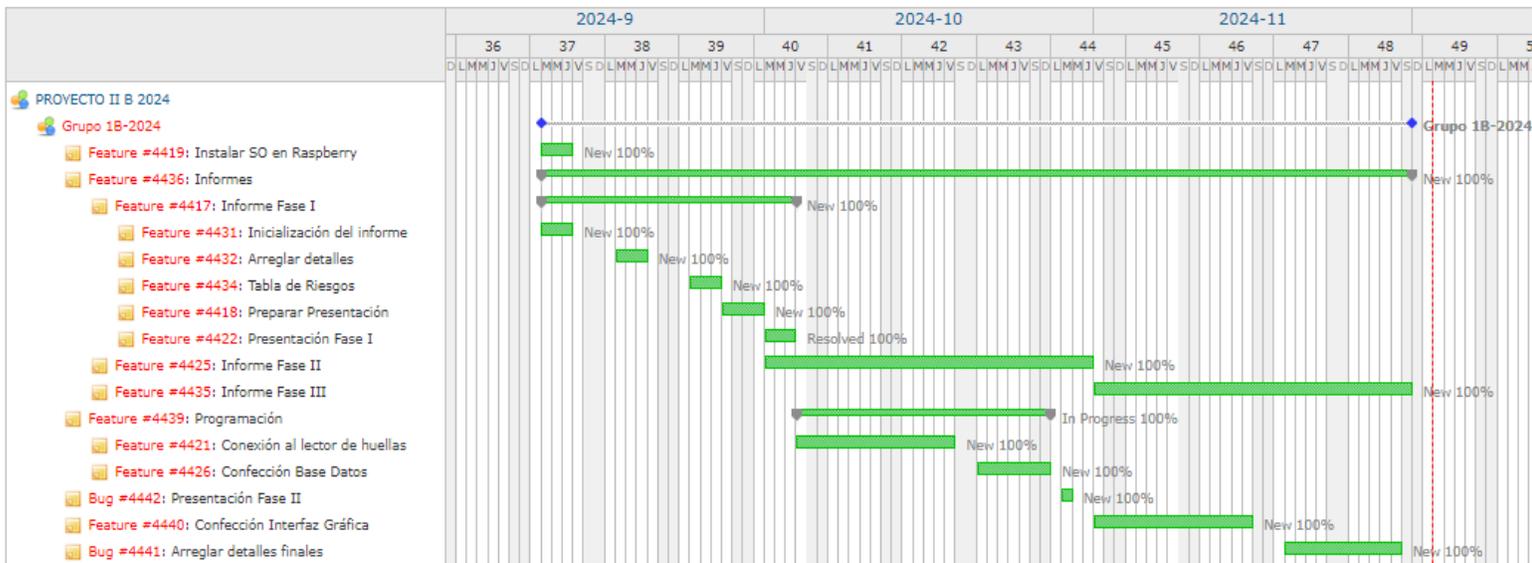
### Integración final del sistema

La integración de todos los módulos fue exitosa, permitiendo que la interfaz, el sensor, la cámara y la pantalla trabajaran en conjunto de manera fluida. Aunque la funcionalidad principal está completa, se identificaron posibles mejoras en la velocidad de respuesta y escalabilidad.

## Progreso del proyecto en vista a como fué planteado desde un inicio mediante la Carta Gantt

El proyecto ha cumplido con todos los objetivos establecidos desde su planificación inicial. Si bien se presentaron algunos retrasos en ciertos requisitos claves, todos los entregables se completaron para la fecha final establecida. Los principales inconvenientes se concentraron en la conexión con el sensor de huellas digitales y la instalación de las librerías necesarias, lo cual demandó más tiempo del planeado originalmente. Sin embargo, gracias a ajustes en la organización y priorización de tareas, se logró recuperar el ritmo del cronograma.

La Carta Gantt refleja claramente los avances realizados y las adaptaciones necesarias para cumplir con los objetivos finales. A pesar de las dificultades, el equipo trabajó de forma eficiente para resolver los problemas técnicos, logrando finalizar exitosamente las tareas críticas antes de la fecha límite. Esto demuestra un manejo adecuado de la planificación y ejecución del proyecto, ajustándose a las realidades enfrentadas durante su desarrollo.



## Problemas Encontrados y Soluciones Implementadas Durante el Desarrollo del Proyecto

Durante el desarrollo del proyecto, se presentaron varios problemas técnicos que requirieron análisis detallado y diversas soluciones. A continuación, se describen los principales inconvenientes y las acciones tomadas para resolverlos:

### 1. No hay manual de usuario:

#### Descripción del problema:

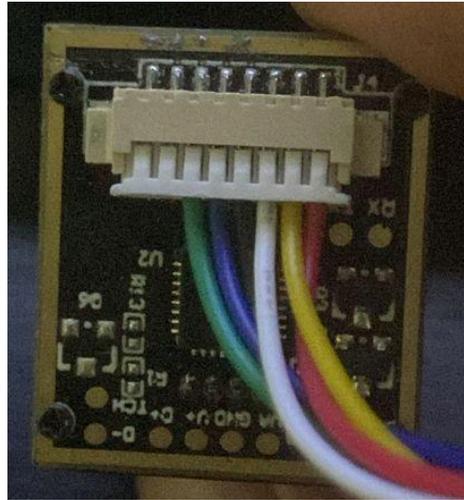
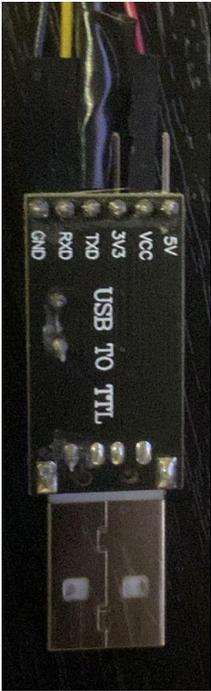
Una de las principales dificultades que enfrentamos fue la falta de documentación detallada sobre el sensor. La única información disponible provenía del vendedor, la cual se limitaba a aspectos básicos sobre el funcionamiento del dispositivo. Sin embargo, no incluía detalles cruciales como las especificaciones de los cables, las conexiones necesarias ni otros aspectos técnicos relevantes para su integración correcta en el sistema.

#### Solución:

La solución consistió en explorar y manipular el sensor directamente para descubrir sus características y funciones. Además, se cruzaron estos datos con la información proporcionada por el vendedor. A continuación, se detallan algunos datos clave que son esenciales para el correcto funcionamiento del sensor.

- \* Modelo: 071405
- \* Chip: AS608
- \* Voltaje de alimentación: 3.3V
- \* Corriente de operación: 100mA - 150mA
- \* Interfaz: UART TTL
- \* Modo de paridad de huella: 1:1 1:N
- \* Baud Rate: 9600\*N
- \* N = 1 a 12 (Por defecto es 6)
- \* Tiempo de adquisición menor a 1 segundo
- \* 5 Niveles de seguridad
- \* Dimensión de la ventana: 14x18mm
- \* Entorno de trabajo: -10°C a 40°C (Humedad Relativa 40% a 85%)

De la foto anterior podemos destacar que funciona a 3.3v, la corriente de operación, la interfaz UART TTL.



(foto adaptador USB A TTL)

(foto placa del sensor)

para la conexión tenemos que tener en cuenta lo siguiente:

del sensor al adaptador

cable de alimentación (cabel rojo) -> 3.3 o VCC

cable de transmisor de datos(TXD amarillo) -> puerto receptor (RXD)

cable de receptor (RXD blanco) -> puerto transmisor de datos (TXD)

cable tierra(negro) -> puerto GND

## 2. No se detecta el sensor

### Descripción del problema:

Durante varias semanas se realizaron pruebas exhaustivas para verificar la comunicación con el sensor de huellas dactilares, sin obtener una respuesta exitosa. Inicialmente, se llegó a pensar que el sensor estaba defectuoso. Sin embargo, al conectarlo directamente a un PC con Windows, el sensor funcionó correctamente.

### Solución:

La solución consistió en reinstalar el sistema operativo en su versión más reciente y oficial. Posteriormente, se activó la comunicación UART y se instalaron las librerías necesarias dentro de un "Entorno Virtual", como **Adafruit Fingerprint Sensor Library**, lo que permitió que el sensor funcionara correctamente. Este proceso eliminó las inconsistencias detectadas en la comunicación entre el sensor y el microcontrolador.

### 3. Instalación de librerías

**Descripción del problema:**

El sistema no lograba instalar las librerías del sensor correctamente, lo que impedía el desarrollo del software.

**Solución:**

Se creó un entorno virtual dedicado y, dentro de este entorno, se instalaron las librerías necesarias. Esto permitió aislar dependencias y asegurar la compatibilidad del sistema con los módulos requeridos.

### 4. Problemas de compatibilidad del Grove Pi+

**Descripción del problema:**

El Grove Pi+ presentaba problemas de compatibilidad, ya que es una versión del 2015 que se estaba utilizando con una Raspberry Pi 4. Esto resultó en incompatibilidades debido a la diferencia en hardware y software.

**Soluciones planteadas:****1. Reinstalar una versión antigua del sistema operativo:**

Una solución fue instalar una versión del sistema operativo que sea compatible con el Grove Pi+. Sin embargo, esto implicaba reinstalar todas las dependencias y reconfigurar el sistema desde cero, lo cual no era ideal.

**2. Activar I2C:** El Grove Pi+ utiliza el bus I2C para la comunicación con los sensores.

```
sudo raspi-config
```

Una vez dentro de la interfaz de configuración hay que ir al apartado de **I2C** y presionar "yes" una vez realizado esto se debe reiniciar la raspberry pi.

**3. Actualizar el firmware del Grove Pi+:**

Se investigó la posibilidad de actualizar el firmware del Grove Pi+, y se confirmó que era factible. Los pasos realizados fueron los siguientes:

- Instalar la herramienta necesaria para actualizar el microcontrolador:

```
sudo apt install avrdude -y //instalamos las dependencias
```

```
avrdude -v // verificamos si se instaló directamente
```

- Clonar el repositorio oficial de soporte de Dexter Industries:

```
git clone https://github.com/DexterInd/GrovePi.git ~/Dexter/GrovePi
```

- Entrar al directorio de firmware y ejecutar la actualización:

```
cd ~/Dexter/GrovePi/Firmware  
sudo bash firmware_update.sh
```

Tras la actualización del firmware, el Grove Pi+ pudo comunicarse correctamente con la Raspberry Pi 4.

## 5. Configuración de la Pantalla LCD

Para utilizar la pantalla LCD RGB Grove, fue necesario instalar las siguientes dependencias y configuraciones:

1. Instalar los paquetes necesarios:

```
sudo apt install python3-smbus i2c-tools
```

2. Instalar la librería específica

```
pip3 install grove_rgb_lcd  
git clone https://github.com/DexterInd/GrovePi.git
```

```
cd GrovePi  
sudo apt-get install python3-smbus python3-pip  
sudo apt-get install python3-grovepi
```

Con esto, se logró que la pantalla LCD funcionara adecuadamente, integrándose perfectamente en el proyecto.

Gracias a estas soluciones, se logró superar los problemas iniciales, garantizando la funcionalidad del sistema y permitiendo avanzar en el desarrollo del proyecto de manera efectiva.

## 6. Conclusión

En la etapa final del proyecto de control de asistencia biométrica, se han alcanzado todos los objetivos planteados, consolidando los avances realizados durante su desarrollo. Desde la planificación inicial hasta la implementación final, se lograron cumplir con las suposiciones y restricciones definidas, asegurando que el sistema funcionara de manera eficiente y acorde a las especificaciones planteadas.

Se implementaron con éxito los componentes electrónicos esenciales, como la Raspberry Pi y el sensor biométrico Adafruit Fingerprint, logrando una integración óptima entre hardware y software. Asimismo, los bocetos iniciales de las pantallas principales evolucionaron en una interfaz funcional, asegurando una experiencia de usuario intuitiva y práctica.

Durante las fases de prueba y ajustes, se superaron desafíos técnicos relacionados con la conexión del sensor de huellas y la instalación de las librerías, lo que permitió entregar un sistema completamente operativo. Este enfoque organizado y el trabajo colaborativo del equipo garantizaron que el proyecto cumpliera con los plazos finales establecidos, sentando las bases para futuras mejoras o expansiones del sistema.

### Planes a Futuro

En relación con los objetivos establecidos para el sistema de control de asistencia biométrica, hemos logrado implementar exitosamente la captura de huellas dactilares como método principal de autenticación. Consideramos que el sistema tiene un gran potencial para evolucionar hacia una solución aún más robusta y eficiente.

Un área clave de mejora sería la diversificación de los métodos de autenticación. Actualmente, el sistema permite registrar huellas dactilares mediante un sensor biométrico, pero la integración de tecnologías adicionales, como el reconocimiento facial o de iris, podría incrementar significativamente la versatilidad del sistema. Esto no solo potenciaría la seguridad y la precisión del proceso de registro, sino que también dificultará la manipulación de los datos, lo que resultaría en un sistema más confiable y eficiente.

Otro aspecto crucial es la interfaz de usuario. Si bien el sistema ya funciona adecuadamente, siempre existe la oportunidad de optimizar la experiencia para hacerla más fluida y accesible. Es fundamental mantener un enfoque constante en la mejora de la experiencia del usuario, ya sea para estudiantes o docentes, asegurando que puedan interactuar con el sistema de manera intuitiva y sin complicaciones.

Otro aspecto interesante para implementar en el futuro sería el control de asistencia de los ayudantes, tanto los ayudantes de cátedra, de laboratorio, como los ayudantes pares. Esto permitiría registrar y gestionar de manera precisa las horas trabajadas por cada uno, asegurando que se cumplan con los horarios establecidos y optimizando la administración de estos roles dentro de las asignaturas. Al integrar este control, no solo se ganaría en eficiencia, sino que también se facilita la rendición de cuentas de los ayudantes, mejorando la organización y transparencia en el proceso.

## 7. Referencias

**Documentación de Raspberry Pi 4:** Guías y manuales oficiales para configurar y programar la [Raspberry Pi](#), esencial para el desarrollo del sistema.

**Manual de sensores Adafruit Fingerprint:** Documentación técnica que explica cómo integrar y usar el [sensor biométrico de huellas dactilares](#).

**Python Documentation:** Referencia oficial de [Python](#), el lenguaje usado en el proyecto para desarrollar el software de control de asistencia.

**Artículos académicos sobre el uso de biometría en educación:** Estudios que muestran cómo la biometría mejora la eficiencia en entornos educativos, como los publicados en el [Journal of Educational Technology Development and Exchange](#).

**GitHub Documentation:** Guías para trabajar con GitHub en la gestión del proyecto y el control de [versiones del código](#)

**Redmine Documentation:** Información sobre cómo usar [Redmine](#) para el seguimiento de tareas y gestión de proyectos, que es útil para la organización del trabajo.

**Artículos sobre mejores prácticas en desarrollo de software:** Material que ayuda a aplicar [buenas prácticas de programación](#) para garantizar un código limpio y eficiente. Ejemplo en

