

Plataforma de Emulación de Hardware para Sistemas Embebidos

Autor

Esp. Lic. Juan Agustin Bassi

Director del trabajo

Dr. Ing. Pablo Gomez (UBA)

Este plan de trabajo ha sido realizado para la asignatura Gestión de la Tecnología y la Innovación.

Tabla de contenido

Registros de cambios	3
Acta de Constitución del Proyecto	4
Descripción técnica-conceptual del Proyecto a realizar	5
Identificación y análisis de los interesados	7
1. Propósito del proyecto	8
2. Alcance del proyecto	8
3. Supuestos del proyecto	9
4. Requerimientos	10
5. Entregables principales del proyecto	11
6. Desglose del trabajo en tareas	12
7. Diagrama de Activity On Node	13
8. Diagrama de Gantt	15
9. Matriz de uso de recursos de materiales	16
10. Presupuesto detallado del proyecto	18
11. Matriz de asignación de responsabilidades	19
12. Gestión de riesgos	21
13. Gestión de la calidad	23
14. Comunicación del proyecto	27
15. Gestión de Compras	28
16. Seguimiento y control	29
17. Procesos de cierre	31

Registros de cambios

Revisión	Detalle de los cambios realizados
1.0	Creación del documento
1.1	Requerimientos del proyecto
1.2	Entregables, desglose del trabajo en tareas
1.3	Diagrama AON, Gantt, recursos materiales, presupuesto, asignación de responsabilidades.
1.4	Gestión de riesgos, calidad, comunicación
1.5	Compras, seguimiento, cierre
1.6	Primera entrega
1.7	Correcciones de la primera entrega

Acta de Constitución del Proyecto

Buenos Aires, .

Por medio de la presente se acuerda con el Sr. Juan Agustin Bassi que su Proyecto Final de la Maestría en Sistemas Embebidos se titulará "Plataforma de Emulación de Hardware para Sistemas Embebidos". Consistirá esencialmente en la creación de una aplicación para teléfono móvil que emulará la presencia de hardware y una librería embebida de hardware virtual que controlará el hardware de la aplicación móvil mediante un módulo Bluetooth. Tendrá un presupuesto preliminar estimado de 600 hs de trabajo y un capital \$10000.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director de la MSE-FIUBA

Dr. Ing. Pablo Gomez
Director del Trabajo Final

Descripción técnica-conceptual del Proyecto a realizar

En función de la experiencia en el desarrollo de Sistemas Embebidos, como así también en la enseñanza de programación, es notable que la curva de aprendizaje inicial es muy dura para los principiantes.

El mundo de los Sistemas Embebidos está compuesto por distintas disciplinas interrelacionadas. Entre las principales están hardware, firmware y software. Para realizar desarrollos es necesario contar con una base de conocimiento previa en varios aspectos, principalmente en programación procedural, electrónica, hardware, periféricos, microcontroladores, lenguaje de programación C, entre otros. Esta base de conocimiento inicial hace que la inserción en este campo sea una experiencia frustrante en muchos casos.

Para personas que nunca interactuaron con dispositivos electrónicos, una de los mayores problemas es el hardware. El principal inconveniente se presenta a la hora conectar cables o placas electrónicas entre sí, a la vez que para que funcione correctamente el hardware se deben respetar a rajatabla los niveles de tensión, polarización, condiciones mínimas y máximas, etc.

Con este panorama presente, y teniendo como objetivo expandir las fronteras de los Sistemas Embebidos para vincularlos con más personas y a la vez generar conocimiento valioso, se pensó en realizar una plataforma que emule la presencia de hardware de manera virtual mediante una aplicación para teléfono móvil. Una conceptualización del proyecto se puede ver en la figura 1.

En la figura se observa que hay una PC donde se desarrollará el firmware para descargar en el Sistema Embebido, generalmente a través de un cable USB. En el Sistema Embebido correrá un programa que se conectará con un módulo Bluetooth. El módulo Bluetooth creará un canal de comunicación entre el Sistema Embebido y el teléfono móvil. Dentro del teléfono móvil correrá una aplicación que emulará hardware. Los elementos de hardware emulados serán principalmente pulsadores, LEDs, potenciómetros, displays, etc. El usuario podrá interactuar con ese hardware emulado en la aplicación a través de la pantalla táctil del teléfono.

Un ejemplo de uso de la aplicación puede estar representado de la siguiente manera. Se supone que un programador quiere desarrollar un algoritmo que al presionar un pulsador encienda un LED y al mismo tiempo informe en un display LCD si tal LED está encendido o apagado. En tal caso, el programador deberá adquirir los componentes necesarios para la aplicación y conectarlos entre sí a través de un hardware diseñado para tal fin o en una placa de prototipado. Luego de eso, debería configurar adecuadamente los periféricos para que el programa deseado funcione correctamente. Es decir, debería configurar un puerto de entrada para el pulsador, un puerto de salida para el LED y puertos de entradas y salidas para el display LCD. Una vez realizados estos pasos, podría probar si el algoritmo cumple con el funcionamiento esperado.

Con el escenario anterior planteado se supone que el programador quiere correr el mismo programa, que al presionar un pulsador se enciende un LED y el estado del LED es informado en un display LCD, mediante el uso de la plataforma de emulación de hardware. En este caso el programador únicamente deberá

conectar un módulo Bluetooth al Sistema Embebido. Dentro de la aplicación móvil habrá un elemento visual que simulará un pulsador, otro que simulará un LED y otro que simulará un Display LCD. El usuario de la aplicación podrá pulsar el botón virtual, esa información le llegará al Sistema Embebido, y el Sistema Embebido le enviará la orden a la aplicación que encienda el LED y a la vez que informe el estado del LED en el Display LCD.

Las ventajas del segundo caso es que el programador puede probar el algoritmo y obtener virtualmente el mismo resultado, sin tener que adquirir los componentes, conectar los componentes entre sí, y eventualmente diseñar una placa de circuito impreso. Con esta plataforma se podrá realizar pruebas de algoritmos de manera muy rápida y prácticamente con los mismos resultados. Por otro lado, es importante pensar en la flexibilidad que presenta una aplicación móvil a la hora de cambiar sus funciones, rápidamente con acceder a otra “placa virtual” se podrán leer varios botones, encender varios LEDs y escribir varios mensajes en distintos displays, todo esto sin costo monetario ni de conexionado para el desarrollador.

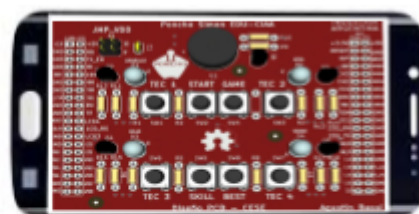
Con esta aplicación, una persona que recién comienza, tendrá la experiencia de relacionarse con un Sistema Embebido solamente conectando el Sistema Embebido a una PC y a un módulo Bluetooth, liberando así el camino a los recién iniciados de probar algoritmos de interacción con hardware manera sencilla.

Pensando que en la actualidad, un teléfono móvil es accesible para la mayoría de la población Argentina, a la aplicación se le podrá ir agregando nuevas funciones y capacidades, y la distribución de las nuevas características puede hacerse de manera muy rápida y escalable al alojar tal aplicación en un servidor mundial.

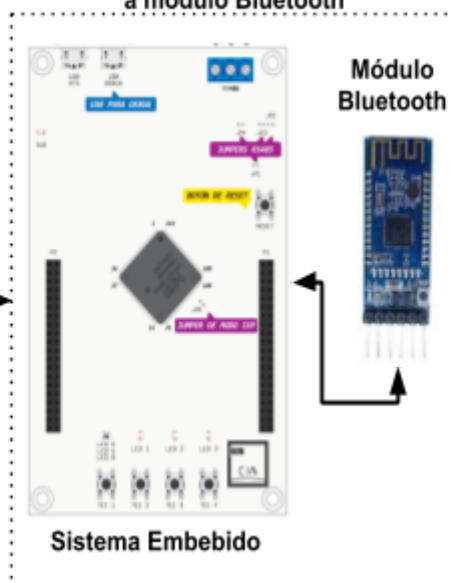
Desarrollo de software en PC



Smartphone con aplicación de
emulación de hardware



Sistema Embebido conectado
a módulo Bluetooth



Programación
del código fuente
desarrollado

Vinculación entre el
Sistema Embebido y
Smartphone mediante la
tecnología Bluetooth

Figura 1: Diagrama en bloques del proyecto, donde se aprecia al Sistema Embebido interactuando con la PC para descargar el código programado a la vez que se conecta a un teléfono móvil para interactuar con múltiples hardware de manera virtual.

Identificación y análisis de los interesados

Rol	Nombre y Apellido	Departamento	Puesto
Cliente	Proyecto CIAA	Investigación y desarrollo Enseñanza	
Responsable	Juan Agustin Bassi	Investigación y desarrollo Producción	Creador
Orientadores	Eric Pernia Grupo de Google Embebidos32	Desarrollo de software	
Usuarios	Docentes y alumnos de enseñanza de Sistemas Embebidos y/o programación Programadores de Sistemas Embebidos		

Cliente: Mediante el desarrollo de la plataforma, el proyecto CIAA expandirá sus fronteras en nuevas aplicaciones y métodos de enseñanza de Sistemas Embebidos.

Responsable: Creador e inventor del proyecto. Trabaja horas extras y fines de semana para lograr el objetivo en tiempo y forma.

Orientadores: En el caso de Eric Pernia una relación personal respalda el apoyo en desarrollo del proyecto. En el caso del grupo de Google Embebidos32 es una comunidad de habla hispana, sus principales miembros son de Argentina y se pueden hacer consultas de todo tipo para las cuales siempre hay respuestas.

Usuarios finales: Personas a cargo de cursos de enseñanza de Sistemas Embebidos que quieran transmitir conceptos de programación a sus alumnos de manera rápida y económica.

Programadores de Sistemas Embebidos que deseen probar algoritmos de interacción con hardware de manera rápida y eficiente.

1. Propósito del proyecto

El propósito del proyecto es crear una plataforma de emulación de hardware con el objetivo de probar algoritmos que interactúan con periféricos sin contar realmente con los mismos. Este proceso facilita el desarrollo de nuevos productos y pruebas de conceptos de manera muy rápida y económica, ya que a priori no se deberá invertir en la compra de los periféricos y por otro lado, no será necesario diseñar hardware ya que esa parte estará resuelta en la aplicación para smartphone.

Otro pilar fundamental del proyecto es integrar el mundo de los Sistemas Embebidos con otras tecnologías, en este caso la integración con un smartphone.

Este proyecto está orientado al prototipado rápido de productos y también a la educación y enseñanza de programación de Sistemas Embebidos, utilizando como principal plataforma la EDU CIAA NXP.

Por otro lado, también se lo realiza para finalizar la Maestría en Sistemas Embebidos y para contribuir con el proyecto de sistemas embebidos más relevante de Argentina, como lo es el Proyecto CIAA.

2. Alcance del proyecto

En el presente proyecto se incluirá:

- Desarrollo de aplicación en Android con hardware virtual emulado.
- Desarrollo de biblioteca de hardware virtual embebida para al menos 3 placas de desarrollo.
- Desarrollo de al menos 3 programas de ejemplo para cada placa de desarrollo.
- Documentación de referencia.
- Informe de avances.
- Memoria del proyecto.

En el presente proyecto no se incluirá:

- Desarrollo de aplicación para otro sistema operativo no Android.
- Desarrollo de aplicación para emular el hardware en PC.

3. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que habrá personal para trabajar diariamente en el proyecto a modo de lograr el planteamiento de los objetivos y cubrir el alcance propuesto.

Por otro lado se supone que se utilizarán los canales de difusión del Proyecto CIAA para difundir el proyecto, a la vez que el grupo embebidos³².

También se supone que el usuario de esta plataforma contará con un módulo Bluetooth y además una placa de desarrollo con puerto serie para la comunicación con el módulo Bluetooth.

También será necesario que se cuente con una herramienta para grabar la memoria flash del microcontrolador.

4. Requerimientos

1. Se deberá respetar un esquema de trabajo.
 - 1.1. Se deberá tener la información detallada del proyecto en directorios debidamente ordenados.
 - 1.2. Se deberá utilizar únicamente softwares libres para el desarrollo del proyecto.
 - 1.3. Se deberá versionar el proyecto en un repositorio Git y se utilizará la metodología de versionado Git Flow.
2. Se deberá investigar y definir la arquitectura del software.
 - 2.1. Se deberá investigar sobre plataformas de desarrollo de aplicaciones Android.
 - 2.2. Se deberá investigar sobre aplicaciones Android que establezcan conexiones Bluetooth.
 - 2.3. Se deberá investigar sobre protocolos de comunicaciones seriales para Sistemas Embebidos.
 - 2.4. Se deberá investigar y documentar arquitectura de software para Android.
 - 2.5. Se deberá investigar y documentar arquitectura de software para el Sistema Embebido.
 - 2.6. Se deberá investigar y documentar guía de estilo de código utilizada en Lenguaje C.
3. Se deberá desarrollar la aplicación en Android.
 - 3.1. Se deberá implementar protocolo de comunicación entre el Sistema Embebido y la aplicación.
 - 3.2. Se deberá disponer de al menos 5 periféricos virtuales (pulsadores, LEDs, display 7 segmentos, potenciómetros, DAC, etc.).
 - 3.3. Se deberá establecer un fondo de pantalla de un PCB para darle una estética realista.
4. Se deberá desarrollar la biblioteca embebida en lenguaje C.
 - 4.1. Se deberá respetar un estilo de código para toda la librería.
 - 4.2. Se deberá aplicar test unitario a cada una de las funciones.
 - 4.3. Se deberá crear documentación en formato Doxygen o Markdown.
 - 4.4. Se deberá adaptar la librería para soportar al menos 3 placas de desarrollo.
5. Se deberá realizar ejemplos de uso.
 - 5.1. Se deberán realizar al menos 3 ejemplos de uso para cada plataforma soportada.
6. Se deberán crear PCBs.
 - 6.1. Se deberá diseñar un PCB (Poncho o shield) para conectar el módulo Bluetooth a cada placa de desarrollo soportada.
 - 6.2. Se deberá fabricar al menos un PCB para cada placa de desarrollo soportada.
7. Se deberá entregar documentación del proyecto.
 - 7.1. Se deberá crear una matriz de trazabilidad de los requerimientos del proyecto.
 - 7.2. Se deberá elaborar un informe de avance del proyecto.
 - 7.3. Se deberá elaborar documentación de uso de la plataforma.
 - 7.4. Se deberá elaborar una memoria del proyecto.
 - 7.5. Se deberá elaborar una presentación de la memoria del proyecto.
8. Se deberá presentar el proyecto públicamente.

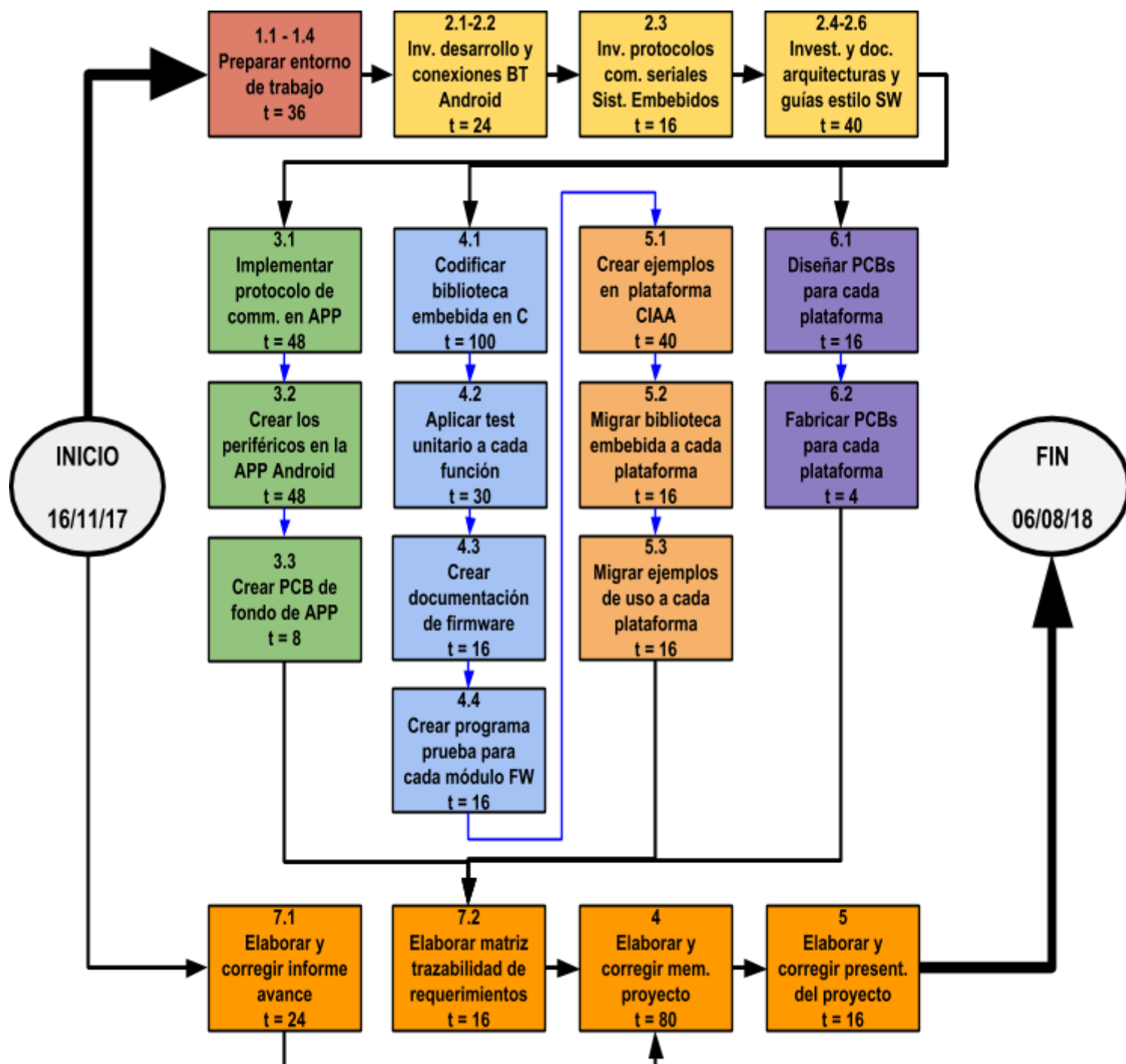
5. Entregables principales del proyecto

- Aplicación Android en Play Store de Google.
- Biblioteca de hardware virtual para al menos 3 placas de desarrollo.
- Al menos 3 ejemplos de uso para cada plataforma soportada.
- Placas de circuito impreso de conexión del módulo Bluetooth para cada placa de desarrollo soportada.
- Documentación técnica del proyecto.
 - Arquitectura del software.
 - Guía de estilo de código.
 - Matriz de trazabilidad de requerimientos.
 - Documentación de biblioteca embebida en formato Doxygen o Markdown.
- Documentación formal del proyecto.
 - Informe de avance del proyecto.
 - Memoria del proyecto.
 - Presentación de memoria del proyecto.

6. Desglose del trabajo en tareas

Tarea		Horas/hombre
1.	Preparar el entorno de trabajo.	-
1.1.	Crear estructura de directorios.	2
1.2.	Instalar las herramientas y programas necesarios.	24
1.3.	Estudiar metodología de versionado Git Flow.	8
1.4.	Crear repositorio Git.	2
2.	Investigar y definir arquitectura de software.	-
2.1.	Investigar sobre el desarrollo de aplicaciones Android.	16
2.2.	Investigar aplicaciones Android con conexión Bluetooth.	8
2.3.	Investigar protocolos de comunicación seriales para Sistemas Embebidos.	16
2.4.	Investigar y documentar arquitectura de software Android.	12
2.5.	Investigar y documentar arquitectura de software Sistema Embebido.	16
2.6.	Investigar y documentar guías de estilos de código en Lenguaje C.	12
3.	Desarrollar la aplicación Android.	-
3.1.	Implementar protocolo de comunicación entre APP y Sistema Embebido.	48
3.2.	Programar funcionamiento periféricos virtuales.	48
3.3.	Crear PCB con imagen de fondo.	8
4.	Desarrollar biblioteca embebida sobre plataforma CIAA.	-
4.1.	Codificar biblioteca en C.	100
4.2.	Aplicar test unitario a cada función.	30
4.3.	Crear documentación con Doxygen o Markdown.	16
4.4.	Crear programa de prueba para cada módulo de la biblioteca.	16
5.	Realizar ejemplos de uso para las diferentes plataformas soportadas.	-
5.1.	Crear programas de ejemplos para plataforma CIAA.	40
5.2.	Migrar biblioteca embebida hacia las demás plataformas.	16
5.3.	Migrar ejemplos creados hacia las demás plataformas.	16
6.	Crear PCBs para cada plataforma soportada.	-
6.1.	Diseñar PCB para cada plataforma soportada.	16
6.2.	Fabricar PCB para cada plataforma soportada.	4
7.	Crear documentación del proyecto.	-
7.1.	Elaborar y corregir informe de avance del proyecto.	24
7.2.	Elaborar matriz de trazabilidad de requerimientos.	16
7.3.	Elaborar y corregir memoria del proyecto.	80
7.4.	Elaborar y corregir presentación del proyecto.	16
Total horas/hombre		612

7. Diagrama de Activity On Node



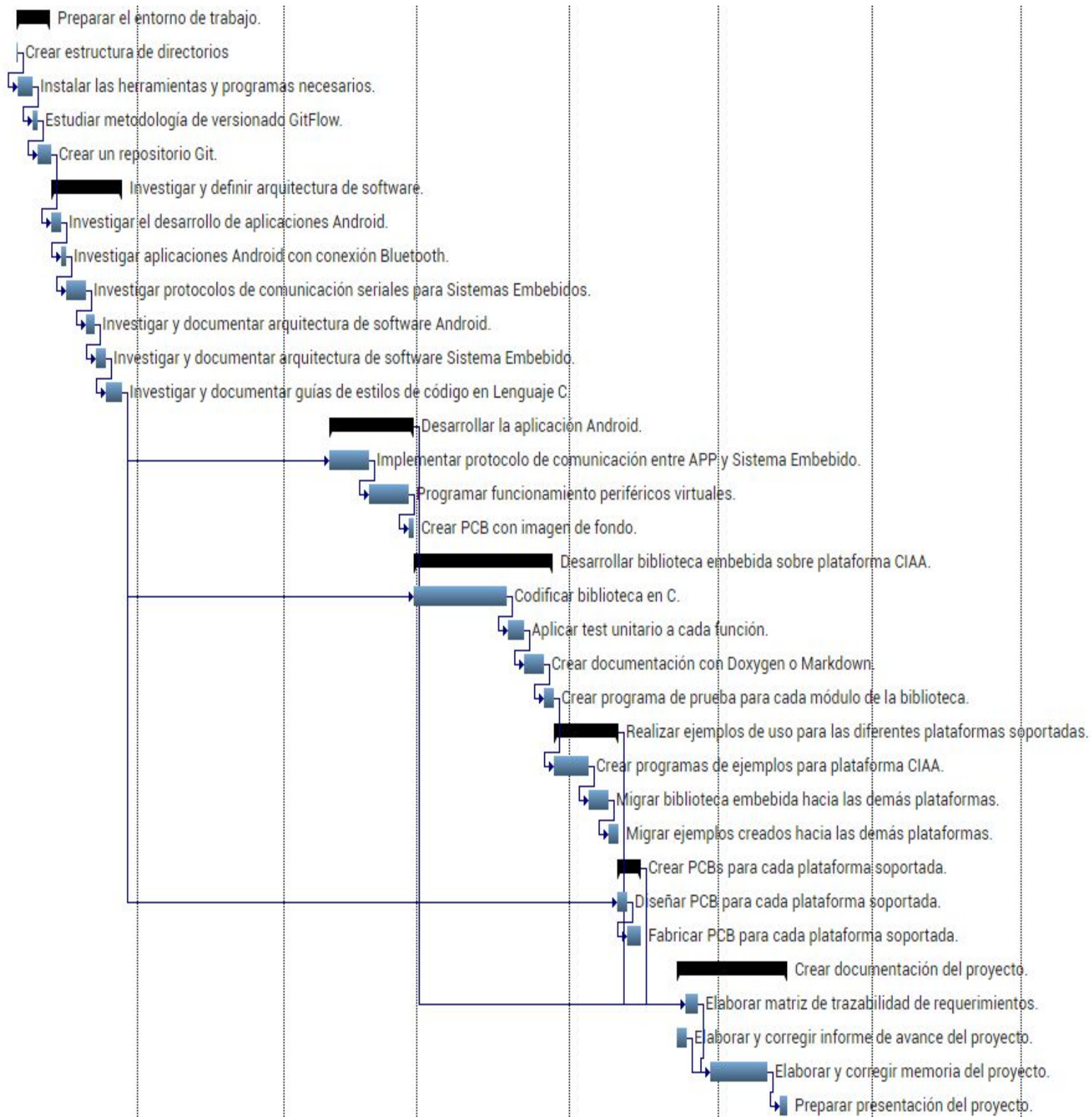
Unidad de tiempo t = horas/hombre.

Los colores de las tareas están ordenados por afinidad.

Es de interés notar cómo, luego de definir la arquitectura de la plataforma, se puede realizar en paralelo el trabajo de desarrollar la aplicación Android, desarrollar la biblioteca embebida de hardware virtual y desarrollar las placas de circuito impreso para las plataformas soportadas. Con esta planificación, y con más de una persona para trabajar se podrían paralelizar 140 horas de trabajo aproximadamente.

Otro detalle importante es que la tarea del informe de avance es una interrupción asincrónica en el proyecto, ya que será requerida durante la ejecución del mismo pero no se sabe cuándo. Tampoco depende de ninguna tarea, sino que su propósito es determinar cómo se está ejecutando la planificación del proyecto. Tener realizada esta tarea es indispensable para realizar la memoria del proyecto.

8. Diagrama de Gantt



9. Matriz de uso de recursos de materiales

Código WBS	Nombre de la tarea	Recursos requeridos (horas)				
		PC	EDU CIAA NXP	Placa desarrollo 2	Placa desarrollo 3	Smartphone Android
1.1-1.4	Preparar el entorno de trabajo. <ul style="list-style-type: none"> • Crear estructura de directorios. • Instalar las herramientas y programas necesarios. • Estudiar metodología de versionado Git Flow. • Crear repositorio Git. 	36	4			
2.1-2.6	Investigar y definir arquitectura de software. <ul style="list-style-type: none"> • Investigar sobre el desarrollo de aplicaciones Android. • Investigar aplicaciones Android con conexión Bluetooth. • Investigar protocolos de comunicación seriales para Sistemas Embebidos. • Investigar y documentar arquitectura de software Android. • Investigar y documentar arquitectura de software Sistema Embebido. • Investigar y documentar guías de estilos de código en Lenguaje C. 	80				
3.1-3.3	Desarrollar la aplicación Android. <ul style="list-style-type: none"> • Implementar protocolo de comunicación entre APP y Sistema Embebido. • Programar funcionamiento periféricos virtuales. • Crear PCB con imagen de fondo. 	104				52
4.1-4.4	Desarrollar biblioteca embebida sobre plataforma CIAA. <ul style="list-style-type: none"> • Codificar biblioteca en C. • Aplicar test unitario a cada función. • Crear documentación con Doxygen o Markdown. • Crear programa de prueba para cada módulo de la biblioteca. 	162	80			

5.1-5.3	Realizar ejemplos de uso para las diferentes plataformas soportadas. <ul style="list-style-type: none"> • Crear programas de ejemplos para plataforma CIAA. • Migrar biblioteca embebida hacia las demás plataformas. • Migrar ejemplos creados hacia las demás plataformas. 	72	20	20	20	20
6.1-6.3	Crear PCBs para cada plataforma soportada. <ul style="list-style-type: none"> • Diseñar PCB para cada plataforma soportada. • Fabricar PCB para cada plataforma soportada. 	20	20	20	20	
7.1-7.4	Crear documentación del proyecto. <ul style="list-style-type: none"> • Elaborar y corregir informe de avance del proyecto. • Elaborar matriz de trazabilidad de requerimientos. • Elaborar y corregir memoria del proyecto. • Elaborar y corregir presentación del proyecto. 	136				
TOTALES		610	124	40	40	72

10. Presupuesto detallado del proyecto

El valor de las unidades debajo es expresado en pesos argentinos ARS.

- Costos de empleados.
 - Hora / hombre = 400
 -
 - **Subtotal horas /hombre del proyecto (612) = 244800**
- Compra de materiales
 - 3 placas de desarrollo = 3000
 - Grabadores de memoria flash microcontroladores = 3000
 - 3 módulos bluetooth = 1000
 - Costo de materiales extra = 1000
 -
 - **Subtotal compra de materiales = 7000**

A la suma de los subtotales obtenidos se agrega un 20% como gastos extras no contemplados en el primer análisis, de modo que el costo total queda representado por los siguientes números.

- Subtotal horas /hombre del proyecto (620) = 244800
- Subtotal compra materiales = 7000
-
- **Subtotales sumados = 251800**
- Gastos extra no contemplados = 50360
-
- **Costo total proyecto = 302160**

11. Matriz de asignación de responsabilidades

Código WBS	Título de la tarea	Juan Agustin Bassi Resp.	Eric Pernia Orient.	Grupo Embebidos 32 Orient.	Pablo Gomez Director
1.1-1.4	Preparar el entorno de trabajo. <ul style="list-style-type: none"> Crear estructura de directorios. Instalar las herramientas y programas necesarios. Estudiar metodología de versionado Git Flow. Crear repositorio Git. 	P			
2.1-2.6	Investigar y definir arquitectura de software. <ul style="list-style-type: none"> Investigar sobre el desarrollo de aplicaciones Android. Investigar aplicaciones Android con conexión Bluetooth. Investigar protocolos de comunicación seriales para Sistemas Embebidos. Investigar y documentar arquitectura de software Android. Investigar y documentar arquitectura de software Sistema Embebido. Investigar y documentar guías de estilos de código en Lenguaje C. 	P	C	C	I
3.1-3.3	Desarrollar la aplicación Android. <ul style="list-style-type: none"> Implementar protocolo de comunicación entre APP y Sistema Embebido. Programar funcionamiento periféricos virtuales. Crear PCB con imagen de fondo. 	P			I
4.1-4.4	Desarrollar biblioteca embebida sobre plataforma CIAA. <ul style="list-style-type: none"> Codificar biblioteca en C. Aplicar test unitario a cada función. Crear documentación con Doxygen o Markdown. Crear programa de prueba para cada módulo de la biblioteca. 	P	C		
5.1-5.3	Realizar ejemplos de uso para las diferentes plataformas soportadas. <ul style="list-style-type: none"> Crear programas de ejemplos para plataforma CIAA. Migrar biblioteca embebida hacia las demás plataformas. Migrar ejemplos creados hacia las demás plataformas. 	P			I
6.1-6.3	Crear PCBs para cada plataforma soportada. <ul style="list-style-type: none"> Diseñar PCB para cada plataforma soportada. Fabricar PCB para cada plataforma soportada. 	P			I

7.1-7.4	<p>Crear documentación del proyecto.</p> <ul style="list-style-type: none"> • Elaborar y corregir informe de avance del proyecto. • Elaborar matriz de trazabilidad de requerimientos. • Elaborar y corregir memoria del proyecto. • Elaborar y corregir presentación del proyecto. 	P			A
---------	---	---	--	--	---

Referencias

- P = Responsabilidad Primaria
- S = Responsabilidad Secundaria
- A = Aprobación
- I = Informado
- C = Consultado

12. Gestión de riesgos

- **Riesgo 1: Daño del hardware con la información del proyecto.**
 - Severidad : 10
Se deberá volver a comenzar el proyecto.
 - Probabilidad de ocurrencia: 6
Es probable que se puede dañar el disco duro de la PC donde está alojada la información del proyecto.
- **Riesgo 2: Desconocimiento de la tecnología Android**
 - Severidad: 10
No se podrá concluir el proyecto.
 - Probabilidad de ocurrencia: 7
Hasta no saber las tecnologías existentes no se puede saber la complejidad del desarrollo de la aplicación.
- **Riesgo 3: Daño de placas de desarrollo soportadas**
 - Severidad: 6
No se podrá cumplir con el requerimiento de soportar al menos 3 placas de desarrollo.
 - Probabilidad de ocurrencia: 6
Es probable que ocurra, ya que en la fase de desarrollo en contacto con hardware, por descuido o manipulación incorrecta se puede dañar una placa de desarrollo.
- **Riesgo 3: Daño o extravío de los PCBs que se conectarán a cada placa de desarrollo**
 - Severidad: 6
No se podrá cumplir con el requerimiento de soportar al menos 3 placas de desarrollo.
 - Probabilidad de ocurrencia: 8
Un PCB es un elemento pequeño que puede extraviarse. También, dependiendo del método de fabricación puede ser frágil y dañarse.

Tabla de gestión de riesgos

Riesgo	Severidad	Ocurrencia	RPN	Severidad*	Ocurrencia*	RPN*
1	10	6	60	10	1	10
2	10	7	70	10	2	20
3	6	6	36			
4	6	8	48			

Criterio adoptado:

- Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a 50.

Nota:

- Los valores marcados con (*) en la tabla corresponden luego de haber aplicado la mitigación.

Plan de mitigación de los riesgos que originalmente excedían el PRN máximo establecido:

- **Riesgo 1: Daño del hardware con la información del proyecto.**
 - Plan de mitigación: Crear al menos un repositorio en la nube para tener un respaldo de la información y actualizarlo en la misma medida que los datos localmente.
 - Severidad : 10
La severidad que se pierda la información es igual de grave.
 - Probabilidad de ocurrencia: 1
Es muy poco probable que se pierda la información de un disco duro de la PC y de un servidor en la nube al mismo tiempo.
- **Riesgo 2: Desconocimiento de la tecnología Android**
 - Plan de mitigación: Investigar sobre tecnologías standard para desarrollo de aplicaciones Android para la plataforma Arduino, donde hay mucha información sobre la creación de aplicaciones en la plataforma Android.
 - Severidad: 10
La severidad que no se podrá concluir con el proyecto es la misma.
 - Probabilidad de ocurrencia: 2
Hay aplicaciones Android creadas por principiantes en programación y Sistemas Embebidos por lo que luego de investigar cómo se desarrollan las aplicaciones no sería una amenaza significativa para el proyecto.

13. Gestión de la calidad

N° Req	Detalle	Verificación	Validación
1.1	Se deberá tener la información detallada del proyecto en directorios debidamente ordenados.	Se verificará que la estructura de los directorios esté basada en proyectos existentes de sistemas embebidos.	Se validará que la información y archivos queden debidamente ordenados, es decir, que no haya archivos sueltos dentro de las ramas principales.
1.2	Se deberá utilizar únicamente softwares libres para el desarrollo del proyecto.	Se comprobará que estos softwares estén disponibles para el sistema operativo que se está utilizando y se comprobará que la PC donde se instalen los programas cumplan con los requerimientos necesarios.	Se realizará una prueba para validar que funcionen correctamente.
1.3	Se deberá versionar el proyecto en un repositorio Git y se utilizará la metodología de versionado Git Flow.	Se verificará que haya espacio disponible en disco para almacenar al menos 2GB para almacenar la historia del proyecto. Además se verificará que la carpeta donde se aloje el directorio tenga permisos de lectura/escritura desde múltiples accesos.	Para validar que el directorio realmente funciona, se realizarán cambios en el proyecto de manera local, se los subirá al repositorio <i>origin</i> (git push origin), se clonará el repositorio <i>origin</i> (git clone origin) y se comprobará que la información e historia del proyecto sea correctamente almacenada en el nuevo repositorio local.
2.1-2.2	Se deberá investigar sobre plataformas de desarrollo de aplicaciones Android. Se deberá investigar sobre aplicaciones Android que establezcan conexiones Bluetooth.	Se verificará que haya aplicaciones Android con conexión Bluetooth para Sistemas Embebidos.	Se validará la aplicación estableciendo los mismos criterios de implementación que aplicaciones ya existentes.
2.3	Se deberá investigar sobre protocolos de comunicaciones seriales para Sistemas Embebidos.	Se verificará que haya protocolos seriales basados en la separación del texto mediante caracteres especiales y delimitadores.	Se validará creando un protocolo separado los comandos a enviar por caracteres especiales basándose en algún criterio probado en proyectos existentes que tengan un funcionamiento apropiado para la aplicación. Se deberá crear documentación sobre el funcionamiento del protocolo y los comandos a intercambiar entre la aplicación y el Sistema Embebido antes de continuar.

2.4-2.6	<p>Se deberá investigar y documentar arquitectura de software para Android.</p> <p>Se deberá investigar y documentar arquitectura de software para el Sistema Embebido.</p> <p>Se deberá investigar y documentar guía de estilo de código utilizada en Lenguaje C.</p>	<p>Se verificará que haya sitios de internet de dónde sacar información, texto e imágenes para poder realizar la documentación de cada una los puntos.</p>	<p>Se validará creando documentación sobre la investigación e implementación de las arquitecturas para Android y el Sistema Embebido empleadas, así como también explicando las guías de estilo de código seguidas en la aplicación.</p>
3.1	<p>Se deberá implementar protocolo de comunicación entre el Sistema Embebido y la aplicación.</p>	<p>Se verificará que el protocolo de comunicación seleccionado esté separado por delimitadores y caracteres especiales.</p>	<p>Se validará que los comandos pueden ser correctamente interpretados por la aplicación enviando tales comandos manualmente dentro de un campo de texto interno.</p> <p>De esta manera se puede probar el protocolo de comunicación de manera sencilla y eficiente sin contar aún con el sistema embebido que le envíe los comandos realmente.</p>
3.2	<p>Se deberá disponer de al menos 5 periféricos virtuales (pulsadores, LEDs, display 7 segmentos, potenciómetros, DAC, etc.).</p>	<p>Se verificará que la plataforma de desarrollo Android seleccionada tenga elementos visuales similares a los periféricos necesarios.</p> <p>De manera general, en aplicaciones gráficas se tienen etiquetas (labels) donde se puede escribir texto y cambiar colores, que sería análoga a un LED y un display 7 segmentos. También se cuenta con botones para simular las teclas del periférico. Por último hay elementos deslizables que pueden ser leídos y escritos, simulando así un potenciómetro y un DAC.</p>	<p>Se verificará ubicando dentro de la aplicación cada uno de estos componentes y que respondan de manera análoga a interactuar con tales periféricos en un hardware real.</p> <p>La calidad del requerimiento estará centrada en darle un aspecto realista a los componentes visuales para que sean lo más parecidos posibles a los reales.</p>
3.3	<p>Se deberá establecer un fondo de pantalla de un PCB para darle una estética realista.</p>	<p>Se verificará que la aplicación pueda tener como fondo de pantalla una imagen.</p>	<p>Se validará creando el diseño de una placa de circuito impreso en un software de diseño CAD, luego se capturará una imagen en 3D del PCB creado, para así importarla como imagen de fondo dentro de la aplicación Android.</p> <p>Se realizarán los ajustes necesarios para que pueda verse correctamente la imagen de fondo.</p>

4.1	Se deberá respetar un estilo de código para toda la librería.	Se verificará que se haya seleccionado una guía de estilo de código previo a codificar.	Se validará ejecutando una herramienta de sintaxis de código luego de crear la librería, o bien se realizará una inspección visual de la misma validando el estilo del código.
4.2	Se deberá aplicar test unitario a cada una de las funciones.	Se verificará que haya una herramienta libre para aplicar test unitario al código.	Se validará ejecutando y pasando correctamente el test unitario a cada función del código. Se elaborará un informe sobre el estado del test.
4.3	Se deberá crear documentación en formato Doxygen o Markdown.	Se verificará instalando una herramienta para tal fin y que funcione correctamente dentro de la plataforma Eclipse.	Se validará generando la documentación del código en formato HTML o PDF mediante alguna de las herramientas de documentación seleccionadas.
4.4	Se deberá adaptar la librería para soportar al menos 3 placas de desarrollo.	Se verificará que la librería cumpla con el standard de código ANSI C que es aceptado por la mayoría de los compiladores C del mercado.	Se validará la librería copiando y pegando el mismo código dentro de cada plataforma y cambiando únicamente la función que envía y recibe bytes por la UART donde está conectado el módulo Bluetooth.
5.1	Se deberán realizar al menos 3 ejemplos de uso para cada plataforma soportada.	Se verificará que haya código de ejemplo dentro de la plataforma CIAA sobre los periféricos reales que serán soportados en la aplicación, adaptando únicamente las llamadas a las funciones que manejan el hardware real a las funciones que manejan el hardware virtual.	Se validará que los programas cumplan prácticamente con la misma funcionalidad, llamando tanto a funciones de hardware real como virtual. Se realizará una comparativa con dos placas de desarrollo corriendo al mismo tiempo, una que interactúe con el hardware real y la otra con el virtual, para establecer una comparativa de tecnologías en el mismo momento.
6.1	Se deberá diseñar un PCB (Poncho o shield) para conectar el módulo Bluetooth a cada placa de desarrollo soportada.	Se verificará que las placas de desarrollo soportadas tengan puertos de expansión o formas de acceder a los pines del microcontrolador correspondientes a la UART y a la alimentación. También se verificará que el diagrama esquemático de cada placa de desarrollo esté disponible para poder realizar el diseño de la placa de circuito impreso.	Se validará conectando un PCB a cada placa de desarrollo soportada y que la comunicación entre el microcontrolador y el módulo Bluetooth funcione correctamente.
6.2	Se deberá fabricar al menos un PCB para cada placa de desarrollo soportada.	Se verificará que las placas de circuito impreso diseñadas presenten características para fabricación artesanal con el propósito de replicar el diseño de manera económica, ya que uno de los pilares de este proyecto es utilizarlo en el	Se validará creando y documentando el proceso de fabricación artesanal de los circuitos impresos.

		ámbito de la educación y enseñanza de Sistemas Embebidos.	
7.1	Se deberá crear una matriz de trazabilidad de los requerimientos del proyecto.	Se verificará que se aplique trazabilidad de cada uno de los requerimientos del proyecto mediante una plantilla. Preferentemente esta plantilla deberá estar basada en las recomendaciones de alguna norma.	Se validará que cada uno de los requerimientos tenga un seguimiento de tareas comprobando si por cada uno existe al menos un responsable de ejecutar tal requerimiento y a la vez un resumen de lo necesario para implementarlo.
7.2	Se deberá elaborar un informe de avance del proyecto.	Se verificará que el informe de avances cumpla con un esquema determinado.	Se validará enviando las correcciones del informe al docente encargado de corregirlo.
7.3	Se deberá elaborar documentación de uso de la plataforma.	Se verificará que se disponga de un sitio online, foro, página web, wiki o plataforma similar para poder guardar y distribuir la información de la plataforma a los usuarios.	Se validará creando la plataforma de documentación y accediendo a la información del proyecto de una manera fácil y concisa.
7.4	Se deberá elaborar una memoria del proyecto.	Se verificará que la memoria del proyecto cumpla con un esquema determinado.	Se validará enviando las correcciones del informe al docente encargado de corregirlo.
7.5	Se deberá elaborar una presentación de la memoria del proyecto.	Se verificará que para realizar la presentación de la memoria se cuenta con un esquema o plantilla de la cual tomar referencia.	Se validará a través de la aprobación del docente tutor de la materia a cargo de la elaboración y presentación de la memoria del proyecto.
8.1	Se deberá presentar el proyecto públicamente.	Se verificará que la presentación del proyecto cumpla con un esquema determinado.	Se validará a través de la nota puesta por el jurado del proyecto.

14. Comunicación del proyecto

El plan de comunicación del proyecto es el siguiente:

PLAN DE COMUNICACIÓN DEL PROYECTO					
¿Qué comunicar?	Audiencia	Propósito	Frecuencia	Método de comunicac.	Responsable
Investigación y documentación del protocolo de comunicación	Colaboradores y Director	Utilizar un protocolo simple, efectivo basado en texto serial	Durante el proceso de selección las veces que sea necesario	Vía mail y teléfono particular	Juan Agustin Bassi
Finalización de la aplicación Android	Colaboradores y Director	Informar una de las partes principales del proyecto	Una vez finalizada	Vía mail	Juan Agustin Bassi
Desarrollo de la biblioteca embebida	Colaboradores	Realizar el desarrollo siguiendo una arquitectura y metodología adecuada	Durante el desarrollo de la librería las veces que sea necesario	Vía mail y teléfono particular	Juan Agustin Bassi
Finalización de la biblioteca y los programas de ejemplo	Colaboradores y Director	Informar una de las partes principales del proyecto	Una vez realizado	Vía mail	Juan Agustin Bassi
Creación de informe de avance	Jurados y director	Evaluar el estado del proyecto	Una vez finalizado	Vía mail	Juan Agustin Bassi
Creación de memoria de proyecto	Jurados y director	Evaluar el proyecto	Una vez finalizada	Vía mail	Juan Agustin Bassi
Presentación formal del proyecto	Jurados y director	Saber el día que deben concurrir a la presentación del proyecto.	Una vez confirmada la fecha.	Vía mail	Juan Agustin Bassi

15. Gestión de Compras

Los materiales necesarios para realizar el proyecto son:

- 3 placas de desarrollo.
- Eventualmente un grabador de memoria de cada una de las placas de desarrollo
- 3 módulos Bluetooth.
- Cables de conexión entre módulos Bluetooth y placas de desarrollo.
- Placas de circuito impreso virgen para realizar la impresión de los PCBs diseñados sobre el cobre.
- Pines de conexión.

Todos los elementos necesarios son componentes muy comunes en Sistemas Embebidos. Al haber muchos oferentes, se elegirá proveedores que estén en zonas cercanas al lugar de desarrollo.

También, al pensar este proyecto como plataforma educativa dentro del territorio nacional, se seleccionarán proveedores que puedan realizar envíos a distintos puntos de Argentina.

16. Seguimiento y control

SEGUIMIENTO DE AVANCE						
Tarea del WBS	Indicador de avance		Frecuencia de reporte	Responsable seguimiento	Persona a ser informada	Método de comunicac.
Preparar entorno de trabajo	Crear estructura de directorios.	10 %	Una vez finalizado	Juan Agustin Bassi	-	-
	Instalar las herramientas y programas necesarios.	40 %				
	Estudiar metodología de versionado Git Flow.	30 %				
	Crear repositorio Git.	20 %				
Investigar y definir arquitectura de software	Investigar sobre el desarrollo de aplicaciones Android.	15 %	Cuando surjan dudas respecto a la aplicación, o bien consejos para recibir consejos	Juan Agustin Bassi	Colaboradores	Personalmente o vía mail
	Investigar aplicaciones Android con conexión Bluetooth.	10 %				
	Investigar protocolos de comunicación seriales para Sistemas Embebidos.	30 %				
	Investigar y documentar arquitectura de software Android.	15 %				
	Investigar y documentar arquitectura de software Sistema Embebido.	15 %				
	Investigar y documentar guías de estilos de código en Lenguaje C.	15 %				
Desarrollar la aplicación Android.	Implementar protocolo de comunicación entre APP y Sistema Embebido.	50 %	Una vez finalizada, o en caso de requerir ayuda en alguno de estos puntos.	Juan Agustin Bassi	Director y/o colaboradores	En persona o vía mail
	Programar funcionamiento periféricos virtuales.	45 %				
	Crear PCB con imagen de fondo.	5 %				

Desarrollar biblioteca embebida sobre plataforma CIAA.	Codificar biblioteca en C.	50 %	Una vez finalizada	Juan Agustin Bassi	Director y/o colaboradores	En persona o vía mail
	Aplicar test unitario a cada función.	20 %				
	Crear documentación con Doxygen o Markdown.	10 %				
	Crear programa de prueba para cada módulo de la biblioteca.	20 %				
Realizar ejemplos de uso para las diferentes plataformas soportadas.	Crear programas de ejemplos para plataforma CIAA.	60 %	Una vez finalizados	Juan Agustin Bassi	Director y/o colaboradores	En persona o vía mail
	Migrar biblioteca embebida hacia las demás plataformas.	20 %				
	Migrar ejemplos creados hacia las demás plataformas.	20 %				
Crear PCBs para cada plataforma soportada.	Diseñar PCB para cada plataforma soportada.	70 %	Una vez finalizados	Juan Agustin Bassi	Director y/o colaboradores	En persona o vía mail
	Fabricar PCB para cada plataforma soportada.	30 %				
Crear documentación del proyecto.	Elaborar y corregir informe de avance del proyecto.	10 %	Cada punto realizado	Juan Agustin Bassi	Docente a cargo, Jurado y director	Vía mail
	Elaborar matriz de trazabilidad de requerimientos.	10 %				
	Elaborar y corregir memoria del proyecto.	60 %				
	Elaborar y corregir presentación del proyecto.	20 %				

17. Procesos de cierre

Al finalizar el proyecto se realizará:

- **Análisis del plan de trabajo:** Se constatarán los requerimientos originales planteados en este documento con los requerimientos cumplidos a lo largo de la ejecución del proyecto, ya sea, mediante el cumplimiento de requerimientos de producto como de metodologías aplicadas.
- **Identificación de técnicas y procedimientos realizados:** En la memoria del proyecto, habrá en la sección Anexos un apartado que hable acerca de las metodologías empleadas, cuales fueron buenas y cuales no, para tener como base ese conocimiento para futuros proyectos. Será una síntesis de consejos y tareas útiles.
- **Organización de acto de agradecimiento a los involucrados:** En agosto de 2018, la organización de la MSE realizará el acto de presentación de los trabajos en el cual este será presentado, en ese acto al que asistirán los involucrados, se agradecerá personalmente a los que estén y a los que no, un agradecimiento vía mail por el tiempo dedicado a la ayuda en el proyecto. Los gastos de la presentación de trabajo y brindis serán financiados por la organización de la MSE.