



## Desarrollo del firmware de la placa de comunicaciones de un Power Train

Autor:  
Marcos Raul Dominguez Shocron

Director:  
A definir (pertenencia)

Codirector:  
John Doe (FIUBA)

*Esta planificación fue realizada en el curso de Gestión de proyectos  
entre el 08 de marzo de 2022 y el 19 de abril de 2022.*



## Firmware de comunicaciones para un tren motor

Autor:  
Marcos Raul Dominguez Shocron

Director:  
A definir (pertenencia)

Codirector:  
John Doe (FIUBA)

*Esta planificación fue realizada en el curso de Gestión de proyectos  
entre el 08 de marzo de 2022 y el 19 de abril de 2022.*

Índice

1. Descripción técnica-conceptual del proyecto a realizar. . . . .	5
2. Identificación y análisis de los interesados . . . . .	6
3. Propósito del proyecto. . . . .	6
4. Alcance del proyecto . . . . .	6
5. Supuestos del proyecto. . . . .	7
6. Requerimientos . . . . .	7
7. Historias de usuarios ( <i>Product backlog</i> ). . . . .	9
8. Entregables principales del proyecto . . . . .	10
9. Desglose del trabajo en tareas . . . . .	10
10. Diagrama de Activity On Node. . . . .	12
11. Diagrama de Gantt . . . . .	12
12. Presupuesto detallado del proyecto . . . . .	15
13. Gestión de riesgos. . . . .	15
14. Gestión de la calidad . . . . .	16
15. Procesos de cierre. . . . .	17

Índice

1. Descripción técnica-conceptual del proyecto a realizar. . . . .	5
2. Identificación y análisis de los interesados . . . . .	6
3. Propósito del proyecto. . . . .	6
4. Alcance del proyecto . . . . .	6
5. Supuestos del proyecto. . . . .	7
6. Requerimientos . . . . .	7
7. Historias de usuarios ( <i>Product backlog</i> ). . . . .	9
8. Entregables principales del proyecto . . . . .	10
9. Desglose del trabajo en tareas . . . . .	10
10. Diagrama de Activity On Node. . . . .	12
11. Diagrama de Gantt . . . . .	12
12. Presupuesto detallado del proyecto . . . . .	14
13. Gestión de riesgos. . . . .	14
14. Gestión de la calidad . . . . .	15
15. Procesos de cierre. . . . .	15

### Registros de cambio

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	08 de marzo de 2022
1	Se completa hasta el punto 5 inclusive	13/03/2022
2	Se completa hasta el punto 9 inclusive	20/03/2022

### Registros de cambio

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	08 de marzo de 2022
1	Se completa hasta el punto 5 inclusive	13/03/2022
2	Se completa hasta el punto 9 inclusive	20/03/2022
3	Se completa hasta el punto 12 inclusive	29/03/2022

## Acta de constitución del proyecto

Buenos Aires, 08 de marzo de 2022

Por medio de la presente se acuerda con el Ing. Marcos Raul Dominguez Shocron que su Trabajo Final de la Carrera de Especialización en Sistemas Embebidos se titulará **"Desarrollo del firmware de la placa de comunicaciones de un Power Train"**, consistirá esencialmente en la implementación del firmware sobre la placa de comunicaciones de un Tren Motor, y tendrá un presupuesto preliminar estimado de **569** hs de trabajo y **\$450000**, con fecha de inicio 08 de marzo de 2022 y fecha de presentación pública a definir.

Se adjunta a esta acta la planificación inicial.

Ariel Lutenberg  
Director posgrado FIUBA

Guillermo Gebhart  
Voltu Motors

A definir  
Director del Trabajo Final

## Acta de constitución del proyecto

Buenos Aires, 08 de marzo de 2022

Por medio de la presente se acuerda con el Ing. Marcos Raul Dominguez Shocron que su Trabajo Final de la Carrera de Especialización en Sistemas Embebidos se titulará **"Firmware de comunicaciones para un tren motor"**, consistirá esencialmente en la implementación del firmware sobre la placa de comunicaciones de un Tren Motor, y tendrá un presupuesto preliminar estimado de **596** hs de trabajo y **\$3447132**, con fecha de inicio 08 de marzo de 2022 y fecha de presentación pública a definir.

Se adjunta a esta acta la planificación inicial.

Ariel Lutenberg  
Director posgrado FIUBA

Guillermo Gebhart  
Voltu Motors

A definir  
Director del Trabajo Final

## 1. Descripción técnica-conceptual del proyecto a realizar

Un Power Train o Tren Motor es el sistema encargado de impulsar los vehículos eléctricos. Estos convierten la energía eléctrica almacenada en baterías a motriz alimentando un motor eléctrico.

El **Power Train** es uno de los dos productos principales que desarrolla y comercializa la empresa Voltu Motors. Este Tren Motor puede adaptarse a diferentes tipos de vehículos, actualmente se han instalado en motos, cuatriciclos, colectivos y camiones.

El dispositivo consta de un pack de baterías, una unidad de control, el motor. La unidad de control de Voltu Motors utiliza dos placas microcontroladas para resolver el **Hardware**. La primera de ellas es un DSP que realiza el control para mover el motor y cargar las baterías. La segunda es la placa de comunicaciones que cumple la función de interactuar con el exterior de la unidad de control.

El usuario del vehículo puede controlar el Tren Motor mediante una interfaz gráfica, la cual se encuentra en el tablero del vehículo. **Este** tablero es una tablet que se comunica con la unidad de control mediante protocolo USB.

La placa de comunicaciones **posee**

- Entradas y salidas digitales para el manejo de periféricos como luces, ventiladores, etc.
- Interfaz USB para comunicarse con la tablet del usuario.
- Interfaz UART para comunicarse con la placa de control, una con fines de debug y otra para comunicarse con el BMS (*Batery Management System*).
- Interfaz para gestionar el conexionado de un EVSE (*Electric Vehicle Supply Equipment*).

Las características listadas anteriormente convierten a la placa de comunicaciones en la encargada de gestionar el estado de la unidad de control. Esta utiliza la información que recibe de todas sus interfaces para la toma de decisiones.

Actualmente, existe una versión funcional con un **Hardware** distinto en donde la placa de control gestiona gran parte de las tareas que se proponen para la nueva placa de comunicaciones.

La nueva implementación permitirá reducir las tareas del DSP a los algoritmos de control del motor y la carga, y mejora así la seguridad del sistema. Además, esta configuración incorpora nuevas características, como el manejo de distintos tipos de EVSE y la posibilidad de definir fuera de tiempo de compilación características del **Power Train** que se adapten a distintos tipos de vehículos.

En la Figura 1 se presenta el diagrama en bloques del sistema. Se puede observar como la placa de comunicaciones interactúa con la Tablet, el DSP y el BMS, mientras que la placa de control se encarga de controlar el flujo de energía para la carga de la batería o marcha del motor. **Adicionalmente** la placa de comunicaciones gestiona las entradas y salidas con los periféricos del vehículo.

Debido a que la nueva versión del **Power Train** está planificada con este **Hardware**, el desarrollo de este proyecto es fundamental para su lanzamiento.

## 1. Descripción técnica-conceptual del proyecto a realizar

Un Power Train o Tren Motor es el sistema encargado de impulsar los vehículos eléctricos. Estos convierten la energía eléctrica almacenada en baterías a motriz alimentando un motor eléctrico.

El **tren motor** es uno de los dos productos principales que desarrolla y comercializa la empresa Voltu Motors. Este Tren Motor puede adaptarse a diferentes tipos de vehículos, actualmente se han instalado en motos, cuatriciclos, colectivos y camiones.

El dispositivo consta de un pack de baterías, una unidad de control, el motor. La unidad de control de Voltu Motors utiliza dos placas microcontroladas para resolver el **hardware**. La primera de ellas es un DSP que realiza el control para mover el motor y cargar las baterías. La segunda es la placa de comunicaciones que cumple la función de interactuar con el exterior de la unidad de control.

El usuario del vehículo puede controlar el Tren Motor mediante una interfaz gráfica, la cual se encuentra en el tablero del vehículo. **Este** tablero es una tablet que se comunica con la unidad de control mediante protocolo USB.

La placa de comunicaciones **posee**:

- Entradas y salidas digitales para el manejo de periféricos como luces, ventiladores, etc.
- Interfaz USB para comunicarse con la tablet del usuario.
- Interfaz UART para comunicarse con la placa de control, una con fines de *debug* y otra para comunicarse con el BMS (*Batery Management System*).
- Interfaz para gestionar el conexionado de un EVSE (*Electric Vehicle Supply Equipment*).

Las características listadas anteriormente convierten a la placa de comunicaciones en la encargada de gestionar el estado de la unidad de control. Esta utiliza la información que recibe de todas sus interfaces para la toma de decisiones.

Actualmente, existe una versión funcional con un **hardware** distinto en donde la placa de control gestiona gran parte de las tareas que se proponen para la nueva placa de comunicaciones.

La nueva implementación permitirá reducir las tareas del DSP a los algoritmos de control del motor y la carga, y mejora así la seguridad del sistema. Además, esta configuración incorpora nuevas características, como el manejo de distintos tipos de EVSE y la posibilidad de definir fuera de tiempo de compilación características del **tren motor** que se adapten a distintos tipos de vehículos.

En la Figura 1 se presenta el diagrama en bloques del sistema. Se puede observar como la placa de comunicaciones interactúa con la Tablet, el DSP y el BMS, mientras que la placa de control se encarga de controlar el flujo de energía para la carga de la batería o marcha del motor. **Adicionalmente**, la placa de comunicaciones gestiona las entradas y salidas con los periféricos del vehículo.

Debido a que la nueva versión del **tren motor** está planificada con este **hardware**, el desarrollo de este proyecto es fundamental para su lanzamiento.

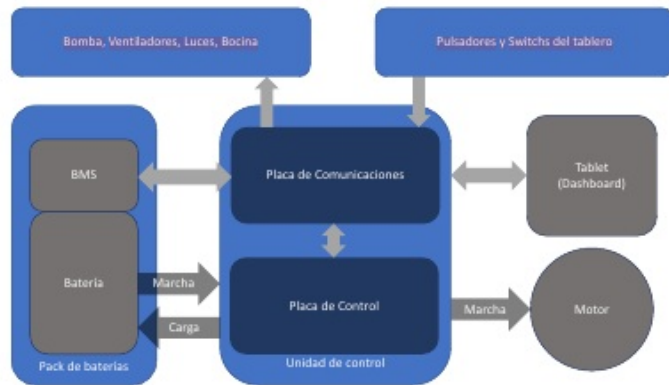


Figura 1. Diagrama en bloques del sistema

## 2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Auspiciante	Guillermo Gebhart	Voltu Motors	CEO y Fundador
Cliente	Guillermo Gebhart	Voltu Motors	CEO y Fundador
Impulsor	Luciano Vittori	Voltu Motors	Lider de Control y Firmware
Responsable	Marcos Raul Dominguez Shocron	FIUBA	Alumno
Orientador	A definir	pertenencia	Director Trabajo final
Usuario final	-	-	-

## 3. Propósito del proyecto

El propósito de este proyecto es desarrollar el firmware de la nueva placa de comunicaciones. **Este** firmware debe cubrir las características de la versión anterior y **ademas** contemplar el manejo del BMS, EVSE y periféricos. La nueva versión también será la encargada de realizar el control de temperatura del sistema.

## 4. Alcance del proyecto

En el presente proyecto se diseñará, desarrollará e implementará el firmware de la placa de comunicaciones hasta que sea funcional para un vehículo que utilice una sola unidad de **Power Train**. Esto incluye la implementación de las siguientes características:

- Manejo de entradas y salidas digitales.
- Comunicación USB con el protocolo actual para la comunicación con la Tablet.

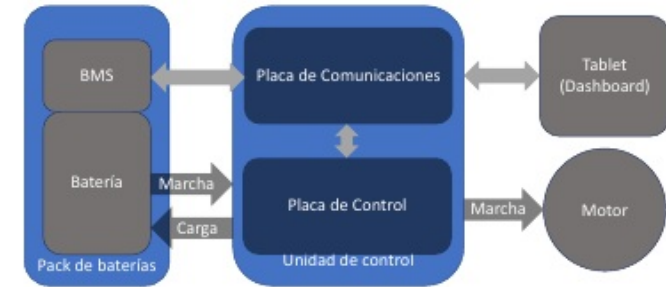


Figura 1. Diagrama en bloques del sistema

## 2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Auspiciante	Guillermo Gebhart	Voltu Motors	CEO y Fundador
Cliente	Guillermo Gebhart	Voltu Motors	CEO y Fundador
Impulsor	Luciano Vittori	Voltu Motors	Lider de Control y Firmware
Responsable	Marcos Raul Dominguez Shocron	FIUBA	Alumno
Orientador	A definir	pertenencia	Director Trabajo final
Usuario final	Gonzalo Cuenca	Voltu Motors	Testing y Calidad

## 3. Propósito del proyecto

El propósito de este proyecto es desarrollar el firmware de la nueva placa de comunicaciones. **Este** firmware debe cubrir las características de la versión anterior y **además** contemplar el manejo del BMS, EVSE y periféricos. La nueva versión también será la encargada de realizar el control de temperatura del sistema.

## 4. Alcance del proyecto

En el presente proyecto se diseñará, desarrollará e implementará el firmware de la placa de comunicaciones hasta que sea funcional para un vehículo que utilice una sola unidad de **tren motor**. Esto incluye la implementación de las siguientes características:

- Manejo de entradas y salidas digitales.
- Comunicación USB con el protocolo actual para la comunicación con la Tablet.

- Comunicación con placa de control, transmisión de datos y recepción de datos.
- Gestión del estado del vehículo.
- Administración de la batería (definir cuando cargar y balancear las celdas).
- Control de temperatura del sistema.

Este proyecto no incluye el diseño del **Hardware** debido a que ya está preestablecido. El montaje del sistema en un vehículo particular tampoco está dentro del alcance del proyecto.

## 5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- Se dispone del **Hardware** de la nueva placa de comunicaciones.
- Se dispone de un banco de pruebas con el sistema completo para realizar pruebas de inmunidad al ruido del firmware.
- Se dispone de un EVSE para las pruebas de carga.
- Se dispone de módulos de batería con placas de BMS montada para desarrollar las estrategias de administración de batería.
- Se dispone de un software de depuración USB para relevar la información del sistema.
- Se dispone de alimentación trifásica para realizar los ciclos de carga.

## 6. Requerimientos

### 1. Interfaces

- 1.1. El sistema debe poder comunicarse con el tablero con protocolo USB.
- 1.2. El sistema debe poseer un botón de encendido
- 1.3. El sistema debe poder actuar sobre los periféricos de refrigeración.
- 1.4. El sistema debe responder a las señales digitales de los pulsadores y switches del vehículo. El tiempo de respuesta debe ser instantáneo para la percepción del usuario **final**.
- 1.5. El sistema debe comunicarse con el BMS.
- 1.6. El sistema debe comunicarse con la placa de control. Esta comunicación debe ser redundante y robusta.
- 1.7. El sistema debe comunicarse con el EVSE.

### 2. Requerimientos funcionales

- 2.1. Comunicación con el tablero.
  - 1) La tasa de actualización de datos debe ser de **10ms**.

- Comunicación con placa de control, transmisión de datos y recepción de datos.
- Gestión del estado del vehículo.
- Administración de la batería (definir cuando cargar y balancear las celdas).
- Control de temperatura del sistema.

Este proyecto no incluye el diseño del **hardware** debido a que ya está preestablecido. El montaje del sistema en un vehículo particular tampoco está dentro del alcance del proyecto.

## 5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- Se dispone del **hardware** de la nueva placa de comunicaciones.
- Se dispone de un banco de pruebas con el sistema completo para realizar pruebas de inmunidad al ruido del firmware.
- Se dispone de un EVSE para las pruebas de carga.
- Se dispone de módulos de batería con placas de BMS montada para desarrollar las estrategias de administración de batería.
- Se dispone de un software de depuración USB para relevar la información del sistema.
- Se dispone de alimentación trifásica para realizar los ciclos de carga.

## 6. Requerimientos

### 1. Interfaces

- 1.1. El sistema debe poder comunicarse con el tablero con protocolo USB.
- 1.2. El sistema debe poseer un botón de encendido
- 1.3. El sistema debe poder actuar sobre los periféricos de refrigeración.
- 1.4. El sistema debe responder a las señales digitales de los pulsadores y switches del vehículo. El tiempo de respuesta debe ser instantáneo para la percepción del usuario **final (menor a 100 ms)**.
- 1.5. El sistema debe comunicarse con el BMS.
- 1.6. El sistema debe comunicarse con la placa de control. Esta comunicación debe ser redundante y robusta.
- 1.7. El sistema debe comunicarse con el EVSE.

### 2. Requerimientos funcionales

- 2.1. Comunicación con el tablero.
  - 1) La tasa de actualización de datos debe ser de **10 ms**.



- 2) Se debe transmitir al tablero toda la información reportada por la placa de control.
- 3) El estado de las entradas y salidas digitales debe ser reportado al tablero.
- 4) Las temperaturas de la unidad de control, batería y motor deben ser reportadas al tablero.
- 5) El sistema puede ser configurado mediante el usuario a través del tablero.

## 2.2. Entradas digitales

- 1) El sistema debe poder recibir señales digitales de los pulsadores y switches del vehículo.
- 2) Las señales deben ser remapeables por cada vehículo, es decir, disociar el pin físico de la funcionalidad específica.
- 3) La activación de las entradas digitales (con tensión o masa) deben ser configurables fuera del tiempo de compilación por personal de la empresa.

## 2.3. BMS

- 1) El sistema debe conocer el estado de todas las celdas.
- 2) El sistema debe conocer las temperaturas en el interior de todos los módulos de batería.
- 3) El sistema debe ser capaz de mantener las baterías balanceadas. Un máximo de 50 mV entre celdas.
- 4) El sistema debe ser capaz de conectarse a distintas configuraciones de baterías según los parámetros del vehículo. Cada vehículo posee distinta cantidad de módulos y distintos modelos de módulos.

## 2.4. Control de temperaturas

- 1) El sistema debe conocer la temperatura de la unidad de control.
- 2) El sistema debe conocer la temperatura de la batería.
- 3) El sistema debe conocer la temperatura del motor.
- 4) El sistema debe accionar los actuadores de refrigeración cuando la temperatura de algún componente exceda el 60 % de la temperatura de falla.
- 5) Las bombas de circulación de líquido deben accionarse siempre que el vehículo se encuentre en marcha o carga.

## 2.5. Carga con EVSE

- 1) El sistema debe poder conectarse con un EVSE Mennekes.
- 2) El sistema debe interpretar el límite de corriente que informa el EVSE para configurar el valor de la corriente de carga.
- 3) El sistema debe conocer la temperatura del EVSE.
- 4) El sistema debe disparar la orden de carga a la placa de control cuando el EVSE esté conectado y listo para cargar.
- 5) La placa de comunicaciones debe sacar al vehículo de marcha si detecta la conexión de un EVSE.

## 2.6. Comunicación con placa de control

- 1) La placa de comunicaciones debe indicar a la placa de control el estado objetivo (marcha, carga o reposo).
- 2) La placa de comunicaciones debe indicar a la placa de control la dirección de avance (Directa o Reversa).
- 3) La placa de comunicaciones debe informar la tensión máxima y mínima de las celdas a la placa de control.

- 2) Se debe transmitir al tablero toda la información reportada por la placa de control.
- 3) El estado de las entradas y salidas digitales debe ser reportado al tablero.
- 4) Las temperaturas de la unidad de control, batería y motor deben ser reportadas al tablero.
- 5) El sistema puede ser configurado mediante el usuario a través del tablero.

## 2.2. Entradas digitales

- 1) El sistema debe poder recibir señales digitales de los pulsadores y switches del vehículo.
- 2) Las señales deben ser remapeables por cada vehículo, es decir, disociar el pin físico de la funcionalidad específica.
- 3) La activación de las entradas digitales (con tensión o masa) deben ser configurables fuera del tiempo de compilación por personal de la empresa.

## 2.3. BMS

- 1) El sistema debe conocer el estado de todas las celdas.
- 2) El sistema debe conocer las temperaturas en el interior de todos los módulos de batería.
- 3) El sistema debe ser capaz de mantener las baterías balanceadas. Un máximo de 50 mV entre celdas.
- 4) El sistema debe ser capaz de conectarse a distintas configuraciones de baterías según los parámetros del vehículo. Cada vehículo posee distinta cantidad de módulos y distintos modelos de módulos.

## 2.4. Control de temperaturas

- 1) El sistema debe conocer la temperatura de la unidad de control.
- 2) El sistema debe conocer la temperatura de la batería.
- 3) El sistema debe conocer la temperatura del motor.
- 4) El sistema debe accionar los actuadores de refrigeración cuando la temperatura de algún componente exceda el 60 % de la temperatura de falla.
- 5) Las bombas de circulación de líquido deben accionarse siempre que el vehículo se encuentre en marcha o carga.

## 2.5. Carga con EVSE

- 1) El sistema debe poder conectarse con un EVSE Mennekes.
- 2) El sistema debe interpretar el límite de corriente que informa el EVSE para configurar el valor de la corriente de carga.
- 3) El sistema debe conocer la temperatura del EVSE.
- 4) El sistema debe disparar la orden de carga a la placa de control cuando el EVSE esté conectado y listo para cargar.
- 5) La placa de comunicaciones debe sacar al vehículo de marcha si detecta la conexión de un EVSE.

## 2.6. Comunicación con placa de control

- 1) La placa de comunicaciones debe indicar a la placa de control el estado objetivo (marcha, carga o reposo).
- 2) La placa de comunicaciones debe indicar a la placa de control la dirección de avance (Directa o Reversa).
- 3) La placa de comunicaciones debe informar la tensión máxima y mínima de las celdas a la placa de control.



4) La placa de comunicaciones debe ser capaz de recibir datos de la placa de control cada 10 ms.

### 3. Requerimientos de robustez

3.1. El sistema debe ser capaz de mantener sus comunicaciones en condiciones ruidosas (aceleraciones fuertes o cargas rápidas).

### 4. Requerimientos de testing

4.1. El sistema debe ser poseer un modo de servicio técnico.

4.2. El modo de servicio técnico debe permitir probar los actuadores del sistema de forma independiente.

4.3. El modo de servicio técnico debe ser vía PC mediante el mismo USB del del tablero.

4.4. El modo de servicio técnico debe permitir la configuración de los parámetros del sistema.

## 7. Historias de usuarios (*Product backlog*)

En esta sección se incluyen las historias de usuarios y su ponderación (*story points*)

"Como usuario quiero poder conectar el cargador y realizar una carga del vehículo."

Dificultad: alta (5) - Implica muchas horas de estudio, diseño e implementación para lograr una coordinación entre el EVSE, placa de comunicaciones, placa de control y BMS.

Complejidad: alta (5) - Realizar un diseño seguro y bien coordinado entre los sistemas es fundamental debido a que se utilizan potencias de varios KW.

Riesgo: alto (5) - Durante los ensayos de esta funcionalidad existe el riesgo de electrochoque e incendio de baterías.

*story points*: 13 ( $5 + 5 + 5 = 15 - 13$  es el valor más cercano en Fibonacci)

"Como desarrollador quiero poder validar el funcionamiento de los actuadores de refrigeración."

Dificultad: alta (5) - Implica el desarrollo de un sistema de servicio técnico que permita interpretar comandos por un protocolo USB.

Complejidad: media (3) - La creación del sistema y protocolo de comunicación implica una complejidad elevada. Pero la conmutación de los actuadores una vez que se interpreta el comando no lo es.

Riesgo: bajo (1) - No hay riesgos importantes al utilizar esta funcionalidad.

*story points*: 13 ( $5 + 3 + 1 = 9 - 8$  es el valor más cercano en Fibonacci)

"Como desarrollador quiero poder configurar los parámetros de un tren motor **conectandome** con una PC al equipo."

Dificultad: alta (4) - Implica muchas horas de diseño de firmware para parametrizar el vehículo con una lista acotada de parámetros.

4) La placa de comunicaciones debe ser capaz de recibir datos de la placa de control cada 10 ms.

### 3. Requerimientos de robustez

3.1. El sistema debe ser capaz de mantener sus comunicaciones en condiciones ruidosas (aceleraciones fuertes o cargas rápidas).

### 4. Requerimientos de testing

4.1. El sistema debe ser poseer un modo de servicio técnico.

4.2. El modo de servicio técnico debe permitir probar los actuadores del sistema de forma independiente.

4.3. El modo de servicio técnico debe ser vía PC mediante el mismo USB del del tablero.

4.4. El modo de servicio técnico debe permitir la configuración de los parámetros del sistema.

## 7. Historias de usuarios (*Product backlog*)

En esta sección se incluyen las historias de usuarios y su ponderación (*story points*)

"Como usuario quiero poder conectar el cargador y realizar una carga del vehículo."

Dificultad: alta (5) - Implica muchas horas de estudio, diseño e implementación para lograr una coordinación entre el EVSE, placa de comunicaciones, placa de control y BMS.

Complejidad: alta (5) - Realizar un diseño seguro y bien coordinado entre los sistemas es fundamental debido a que se utilizan potencias de varios KW.

Riesgo: alto (5) - Durante los ensayos de esta funcionalidad existe el riesgo de electrochoque e incendio de baterías.

*story points*: 13 ( $5 + 5 + 5 = 15 - 13$  es el valor más cercano en Fibonacci)

"Como desarrollador quiero poder validar el funcionamiento de los actuadores de refrigeración."

Dificultad: alta (5) - Implica el desarrollo de un sistema de servicio técnico que permita interpretar comandos por un protocolo USB.

Complejidad: media (3) - La creación del sistema y protocolo de comunicación implica una complejidad elevada. Pero la conmutación de los actuadores una vez que se interpreta el comando no lo es.

Riesgo: bajo (1) - No hay riesgos importantes al utilizar esta funcionalidad.

*story points*: 13 ( $5 + 3 + 1 = 9 - 8$  es el valor más cercano en Fibonacci)

"Como desarrollador quiero poder configurar los parámetros de un tren motor **conectándome** con una PC al equipo."

Dificultad: alta (4) - Implica muchas horas de diseño de firmware para parametrizar el vehículo con una lista acotada de parámetros.

Complejidad: baja (1) - La dependencia de valores y condiciones según valores de parámetros no conlleva una gran complejidad.

Riesgo: bajo (1) - El uso de una pc mediante USB para modificar valores del equipo no es una actividad riesgosa.

*story points:* 8 ( $4 + 1 + 1 = 6 - 5$  es el valor más cercano en Fibonacci)

“Como usuario quiero encender el vehículo y marcharlo”

Dificultad: alta (5) - Implica muchas horas de diseño de firmware para coordinar las instrucciones del usuario con la placa de control mientras se monitorea la integridad del sistema para evitar accidentes.

Complejidad: alta (5) - Contemplar las totalidad de la situaciones para garantizar la seguridad del usuario durante la marcha es una actividad muy compleja.

Riesgo: alto (5) - Una falla en es sistema no contemplado puede tener como consecuencia una fatalidad.

*story points:* 8 ( $5 + 5 + 5 = 15 - 13$  es el valor más cercano en Fibonacci)

“Como usuario quiero poder encender las luces de mi vehículo.”

Dificultad: baja (1) - Implica atender entradas digitales y activar salidas digitales.

Complejidad: baja (1) - El manejo de entradas y salidas digitales de baja complejidad en el desarrollo de firmware.

Riesgo: bajo (1) - No existen grandes riesgos encendiendo y apagando luces que funcionan a 12 V.

*story points:* 8 ( $1 + 1 + 1 = 3 - 3$  es el valor más cercano en Fibonacci)

## 8. Entregables principales del proyecto

Los entregables del proyecto son:

- Diagrama del firmware
- Código fuente del firmware
- Manual de uso para el sistema de servicio técnico
- Informe final

## 9. Desglose del trabajo en tareas

1. Comunicación con el tablero (92 horas).

1.1. Implementación de capa USB e implementación del protocolo de comunicación (40 horas).

Complejidad: baja (1) - La dependencia de valores y condiciones según valores de parámetros no conlleva una gran complejidad.

Riesgo: bajo (1) - El uso de una pc mediante USB para modificar valores del equipo no es una actividad riesgosa.

*story points:* 8 ( $4 + 1 + 1 = 6 - 5$  es el valor más cercano en Fibonacci)

“Como usuario quiero encender el vehículo y marcharlo”

Dificultad: alta (5) - Implica muchas horas de diseño de firmware para coordinar las instrucciones del usuario con la placa de control mientras se monitorea la integridad del sistema para evitar accidentes.

Complejidad: alta (5) - Contemplar las totalidad de la situaciones para garantizar la seguridad del usuario durante la marcha es una actividad muy compleja.

Riesgo: alto (5) - Una falla en es sistema no contemplado puede tener como consecuencia una fatalidad.

*story points:* 8 ( $5 + 5 + 5 = 15 - 13$  es el valor más cercano en Fibonacci)

“Como usuario quiero poder encender las luces de mi vehículo.”

Dificultad: baja (1) - Implica atender entradas digitales y activar salidas digitales.

Complejidad: baja (1) - El manejo de entradas y salidas digitales de baja complejidad en el desarrollo de firmware.

Riesgo: bajo (1) - No existen grandes riesgos encendiendo y apagando luces que funcionan a 12 V.

*story points:* 8 ( $1 + 1 + 1 = 3 - 3$  es el valor más cercano en Fibonacci)

## 8. Entregables principales del proyecto

Los entregables del proyecto son:

- Diagrama del firmware
- Código fuente del firmware
- Manual de uso para el sistema de servicio técnico
- Informe final

## 9. Desglose del trabajo en tareas

1. Comunicación con el tablero (92 horas).

1.1. Implementación de capa USB e implementación del protocolo de comunicación (40 horas).

- 1.2. Implementación del empaquetado de la información de la placa de control para su retransmisión (16 horas).
- 1.3. Implementación del reporte de entradas y salidas digitales al tablero (8 horas).
- 1.4. Implementación del empaquetado de las temperaturas y envío al tablero (8 horas).
- 1.5. Implementación de protocolo para recibir instrucciones desde el tablero (16 horas).
- 1.6. Validación de la comunicación a 10 ms (4 horas).
2. Entradas digitales (40 horas).
  - 2.1. Implementación de capa de atención a entradas digitales con abstracción entre la función y el pin físico (16 horas).
  - 2.2. Implementación de capa de activación de salidas digitales con abstracción entre la función y el pin físico (16 horas).
  - 2.3. Implementación de la lógica para configurar el mecanismo de activación de las entradas digitales (con tensión o masa) (8 horas).
3. BMS (80 horas).
  - 3.1. Implementar la configuración de los parámetros de los BMS (16 horas).
  - 3.2. Implementación de la inicialización del sistema de BMS según los parámetros de configuración (32 horas).
  - 3.3. Implementar las lecturas de tensiones de celdas (8 horas).
  - 3.4. Implementar las lecturas de temperaturas de módulos (8 horas).
  - 3.5. Implementar lógica y mecanismo de balanceo de celdas (16 horas).
4. Control de temperaturas (56 horas)
  - 4.1. Activación de entradas analógicas para la medición de temperatura de motor (16 horas).
  - 4.2. Sustracción de la temperaturas de la unidad de control informada por la placa de control (8 horas).
  - 4.3. Implementación de lógica de control de temperaturas según los límites establecidos (histéresis entre el 45 % y 55 % de las temperaturas de falla) (16 horas).
  - 4.4. Validación del correcto funcionamiento del sistema (16 horas).
5. Carga con EVSE (80 horas).
  - 5.1. Implementar la detección de conexión del EVSE (16 horas).
  - 5.2. Implementación de la decodificación del PWM para conocer la corriente máxima de la carga (16 horas).
  - 5.3. Implementación de la lógica de carga y coordinación entre el BMS, EVSE y placa de control (32 horas).
  - 5.4. Implementación de la salida de marcha ante el conexionado del EVSE (8 horas).
  - 5.5. Validación del sistema con una carga a 16 A por fase (8 horas).
6. Comunicación con placa de control (56 horas).
  - 6.1. Implementación de la recepción de datos de la placa de control vía UART (32 horas).
  - 6.2. Implementación del envío de datos de la placa de control vía UART (16 horas).

- 1.2. Implementación del empaquetado de la información de la placa de control para su retransmisión (16 horas).
- 1.3. Implementación del reporte de entradas y salidas digitales al tablero (8 horas).
- 1.4. Implementación del empaquetado de las temperaturas y envío al tablero (8 horas).
- 1.5. Implementación de protocolo para recibir instrucciones desde el tablero (16 horas).
- 1.6. Validación de la comunicación a 10 ms (4 horas).
2. Entradas digitales (40 horas).
  - 2.1. Implementación de capa de atención a entradas digitales con abstracción entre la función y el pin físico (16 horas).
  - 2.2. Implementación de capa de activación de salidas digitales con abstracción entre la función y el pin físico (16 horas).
  - 2.3. Implementación de la lógica para configurar el mecanismo de activación de las entradas digitales (con tensión o masa) (8 horas).
3. BMS (80 horas).
  - 3.1. Implementar la configuración de los parámetros de los BMS (16 horas).
  - 3.2. Implementación de la inicialización del sistema de BMS según los parámetros de configuración (32 horas).
  - 3.3. Implementar las lecturas de tensiones de celdas (8 horas).
  - 3.4. Implementar las lecturas de temperaturas de módulos (8 horas).
  - 3.5. Implementar lógica y mecanismo de balanceo de celdas (16 horas).
4. Control de temperaturas (56 horas)
  - 4.1. Activación de entradas analógicas para la medición de temperatura de motor (16 horas).
  - 4.2. Sustracción de la temperaturas de la unidad de control informada por la placa de control (8 horas).
  - 4.3. Implementación de lógica de control de temperaturas según los límites establecidos (histéresis entre el 45 % y 55 % de las temperaturas de falla) (16 horas).
  - 4.4. Validación del correcto funcionamiento del sistema (16 horas).
5. Carga con EVSE (80 horas).
  - 5.1. Implementar la detección de conexión del EVSE (16 horas).
  - 5.2. Implementación de la decodificación del PWM para conocer la corriente máxima de la carga (16 horas).
  - 5.3. Implementación de la lógica de carga y coordinación entre el BMS, EVSE y placa de control (32 horas).
  - 5.4. Implementación de la salida de marcha ante el conexionado del EVSE (8 horas).
  - 5.5. Validación del sistema con una carga a 16 A por fase (8 horas).
6. Comunicación con placa de control (56 horas).
  - 6.1. Implementación de la recepción de datos de la placa de control vía UART (32 horas).
  - 6.2. Implementación del envío de datos de la placa de control vía UART (16 horas).

- 6.3. Validación de la integridad de los datos a la tasa exigida por los requerimientos (10 ms) (8 horas).
7. Modo servicio técnico (72 horas).
- 7.1. Implementación del modo de servicio técnico (16 horas).
- 7.2. Creación e implementación de comandos de servicio técnico (16 horas).
- 7.3. Prueba de los comandos de servicio técnico (8 horas).
- 7.4. Redacción de manual de uso para el sistema de servicio técnico (32 horas).
8. Informe final (120 horas).
- 8.1. Redacción del marco teórico (40 horas).
- 8.2. Realización de diagramas del firmware (40 horas).
- 8.3. Redacción de resultados (40 horas).

Cantidad total de horas: 596 horas.

## 10. Diagrama de Activity On Node

Armar el AoN a partir del WBS definido en la etapa anterior.



Figura 2. Diagrama en Activity on Node

Indicar claramente en qué unidades están expresados los tiempos. De ser necesario indicar los caminos semicríticos y analizar sus tiempos mediante un cuadro. Es recomendable usar colores y un cuadro indicativo describiendo qué representa cada color, como se muestra en el siguiente ejemplo:

## 11. Diagrama de Gantt

Existen muchos programas y recursos *online* para hacer diagramas de gantt, entre los cuales destacamos:

- 6.3. Validación de la integridad de los datos a la tasa exigida por los requerimientos (10 ms) (8 horas).
7. Modo servicio técnico (72 horas).
- 7.1. Implementación del modo de servicio técnico (16 horas).
- 7.2. Creación e implementación de comandos de servicio técnico (16 horas).
- 7.3. Prueba de los comandos de servicio técnico (8 horas).
- 7.4. Redacción de manual de uso para el sistema de servicio técnico (32 horas).
8. Informe final (120 horas).
- 8.1. Redacción del marco teórico (40 horas).
- 8.2. Realización de diagramas del firmware (40 horas).
- 8.3. Redacción de resultados (40 horas).

Cantidad total de horas: 596 horas.

## 10. Diagrama de Activity On Node

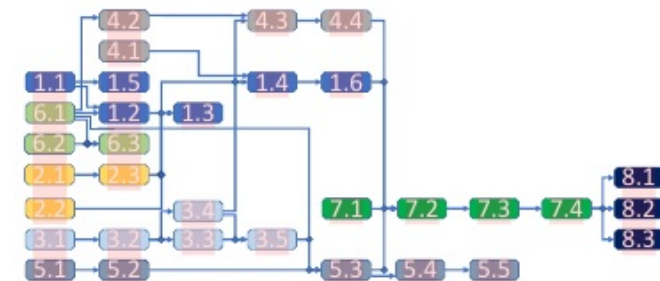


Figura 2. Diagrama en Activity on Node

## 11. Diagrama de Gantt

- **Planner**
- **GanttProject**
- **Trello + plugins.** En el siguiente link hay un tutorial oficial:  
<https://blog.trello.com/es/diagrama-de-gantt-de-un-proyecto>
- **Creately;** herramienta online colaborativa.  
<https://creately.com/diagram/example/ieb3p3ml/LaTeX>
- **Se puede hacer en latex con el paquete *pgfgantt***  
<http://ctan.dcc.uchile.cl/graphics/pgf/contrib/pgfgantt/pgfgantt.pdf>

Pegar acá una captura de pantalla del diagrama de Gantt, cuidando que la letra sea suficientemente grande como para ser legible. Si el diagrama queda demasiado ancho, se puede pegar primero la "tabla" del Gantt y luego pegar la parte del diagrama de barras del diagrama de Gantt.

Configurar el software para que en la parte de la tabla muestre los códigos del EDT (WBS).  
Configurar el software para que al lado de cada barra muestre el nombre de cada tarea.  
Revisar que la fecha de finalización coincida con lo indicado en el Acta Constitutiva.

En la figura 3, se muestra un ejemplo de diagrama de gantt realizado con el paquete de *pgfgantt*.  
En la plantilla pueden ver el código que lo genera y usarlo de base para construir el propio.

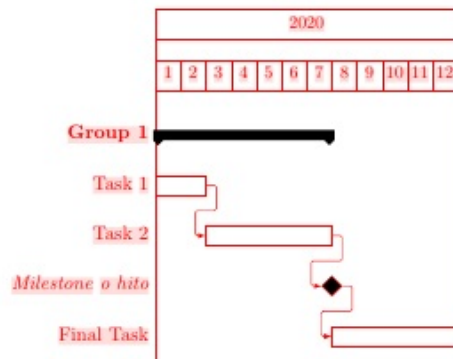


Figura 3. Diagrama de gantt de ejemplo

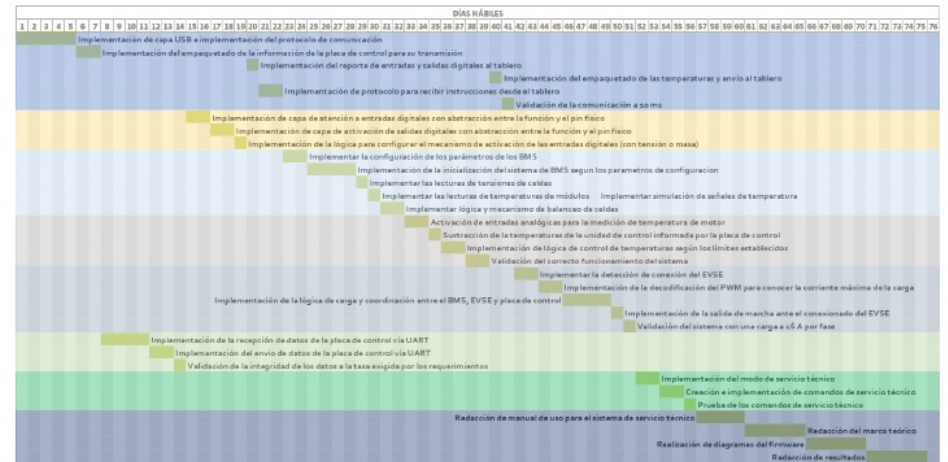


Figura 3. Diagrama de Gantt





Figura 4. Ejemplo de diagrama de Gantt rotado

Plan de proyecto de Trabajo final  
Carrera de Especialización en Sistemas Embebidos  
Marcos Raul Dominguez Shocron

COSTOS DIRECTOS				
Descripción		Cantidad	Valor unitario	Valor total
Placa de comunicaciones		2	10000	20000
Placa de control		2	15000	30000
EVSE		1	60000	60000
Modulo de baterías con BMS		4	410400	1641640
Honorarios de desarrollador		600	1500	900000
SUBTOTAL				2651640
COSTOS INDIRECTOS				
Descripción		Cantidad	Valor unitario	Valor total
30 % de los costos directos		1	795492	795492
SUBTOTAL				294300
TOTAL				3447132

a) Identificación de los riesgos (al menos cinco) y estimación de sus consecuencias:

**Riesgo 1:** detallar el riesgo (riesgo es algo que si ocurre altera los planes previstos de forma negativa)

- **Severidad (S):** mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- **Probabilidad de ocurrencia (O):** mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

**Riesgo 2:**

- Severidad (S):
- Ocurrencia (O):

### Riesgo 3:

- Severidad (S):
- Ocurrencia (O):

b) Tabla de gestión de riesgos: (El RPN se calcula como  $RPN=S \times O$ )

Criterio adoptado: Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a...

Nota: los valores marcados con (\*) en la tabla corresponden luego de haber aplicado la mitigación.





■ **Ocurrencia (O):**

b) Tabla de gestión de riesgos: (El RPN se calcula como  $RPN = S \times O$ )

Riesgo	S	O	RPN	S*	O*	RPN*

Criterio adoptado: Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a...

Nota: los valores marcados con (\*) en la tabla corresponden luego de haber aplicado la mitigación.

c) Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:

Riesgo 1: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación). Nueva asignación de S y O, con su respectiva justificación: - Severidad (S): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S); - Probabilidad de ocurrencia (O): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

Riesgo 3: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

#### 14. Gestión de la calidad

Para cada uno de los requerimientos del proyecto indique:

- Req #1: copiar acá el requerimiento.
  - Verificación para confirmar si se cumplió con lo requerido antes de mostrar el sistema al cliente. Detallar
  - Validación con el cliente para confirmar que está de acuerdo en que se cumplió con lo requerido. Detallar

Tener en cuenta que en este contexto se pueden mencionar simulaciones, cálculos, revisión de hojas de datos, consulta con expertos, mediciones, etc. Las acciones de verificación suelen considerar al entregable como "caja blanca", es decir se conoce en profundidad su funcionamiento interno. En cambio, las acciones de validación suelen considerar al entregable como "caja negra", es decir, que no se conocen los detalles de su funcionamiento interno.

- Indicar quién organizará el acto de agradecimiento a todos los interesados, y en especial al equipo de trabajo y colaboradores; - Indicar esto y quién financiará los gastos correspondientes.