

Electric Motor Control System

*Autores:*

*Jorge López Casillas*

*Jesús Peña*

*Marlen González Rey*

# Aprobaciones

# Historial de revisiones

Tabla de contenido

[Aprobaciones 2](#_Toc3400333)

[Historial de revisiones 2](#_Toc3400334)

[Alcance del proyecto 5](#_Toc3400335)

[Identificación del proyecto 5](#_Toc3400336)

[Descripción general del proyecto 5](#_Toc3400337)

[Entregables con el cliente 5](#_Toc3400338)

[Metodología de Desarrollo 6](#_Toc3400339)

[Metodología de desarrollo de SW 6](#_Toc3400340)

[Plan de entregas de software al cliente 6](#_Toc3400341)

[Procedimientos de control 6](#_Toc3400342)

[Estimación 7](#_Toc3400343)

[Recursos 7](#_Toc3400344)

[Recursos críticos 7](#_Toc3400345)

[Infraestructura y herramientas 7](#_Toc3400346)

[Recursos humanos 7](#_Toc3400347)

[Suposiciones 7](#_Toc3400348)

[Restricciones 7](#_Toc3400349)

[Riesgos 7](#_Toc3400350)

[Resolución de problemas 7](#_Toc3400351)

[Planeación 7](#_Toc3400352)

[Roles y responsabilidades 7](#_Toc3400353)

[Trazabilidad de entregables 8](#_Toc3400354)

[Administración de riesgos 8](#_Toc3400355)

[Análisis de requisitos de software 8](#_Toc3400356)

[Diseño de arquitectura de sowtfare 9](#_Toc3400357)

[Diagrama de bloques de SW 9](#_Toc3400358)

[Diagrama de control 9](#_Toc3400359)

[UML 9](#_Toc3400360)

[Árbol de llamadas 9](#_Toc3400361)

[Diagramas de flujo 9](#_Toc3400362)

[Final: ROM, RAM, STACK, Mapa de Memoria 9](#_Toc3400363)

[Throughput 9](#_Toc3400364)

[Complejidad de Código 9](#_Toc3400365)

[Construcción del software 9](#_Toc3400366)

[Aplicación de estándares de codificación y convenciones 9](#_Toc3400367)

[Nomenclatura 9](#_Toc3400368)

[Pruebas de software 9](#_Toc3400369)

[Pruebas de integración 9](#_Toc3400370)

[White Box Test 9](#_Toc3400371)

[Gray Box 9](#_Toc3400372)

[ABA test 9](#_Toc3400373)

[Liberación de versiones de software 9](#_Toc3400374)

[Git 9](#_Toc3400375)

[Verificación y aseguramiento de la calidad 9](#_Toc3400376)

[Prueba funcional y validación en campo 9](#_Toc3400377)

[Calendario de revisión. 9](#_Toc3400378)

[Referencias 10](#_Toc3400379)

# Alcance del proyecto

## Identificación del proyecto

El propósito de este proyecto es poner en práctica todos los conocimientos adquiridos en el curso de Software Embebido.

El propósito particular de este documento es proporcionar evidencia con respecto al proyecto desarrollado a lo largo de este curso.

## Descripción general del proyecto

El Sistema de control de motor eléctrico (EMCS) consiste en la implementación del control de un motor de corriente continua en una placa de desarrollo Synergy Renesas.

## Entregables con el cliente

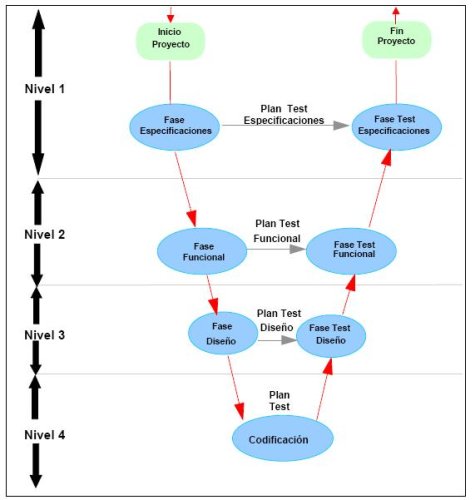
El entregable final consiste en la entrega física del EMCS así como la documentación generada a partir de la planeación, desarrollo y conclusión del mismo.

# Metodología de Desarrollo

## Metodología de desarrollo de SW

La metodología de desarrollo de software elegida para este proyecto es Agile. Dicha metodología emplea un enfoque iterativo e incremental para la optimización de la predictibilidad y el control de riesgo, es decir, los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.

Dado el enfoque anterior, se implementó el modelo de desarrollo de software de V, el cual define un procedimiento uniforme para el desarrollo de software. Este modelo de desarrollo es ideal por su robustez y por su claridad. La Figura 1 representa de manera gráfica el modelo en V.

**No se encuentran elementos de tabla de ilustraciones.**

**Figura 1: Modelo de desarrollo de SW en V**

## Plan de entregas de software al cliente

La entrega final del EMCS hacia el cliente consiste en una entrega final con el equipo funcionando y la documentación generada a partir de este, así como toda la infraestructura que formó parte del desarrollo del EMCS.

## Procedimientos de control

Las herramientas seleccionadas para el control del desarrollo del EMCS son:

* **Git:** es una herramienta de control de versiones cuyo propósito es tener un registro de los cambios en archivos de ordenador, además, coordina el trabajo que diferentes miembros del equipo de desarrollo realizan sobre archivos que son compartidos. Esta herramienta se utiliza para llevar un control de todos los archivos generados a partir del desarrollo del EMCS, incluyendo documentos de planeación, desarrollo y conclusiones, así como los archivos fuente que componen y describen la completa funcionalidad del EMCS.
* **Excel:** herramienta de hojas de cálculo utilizada para el registro de requisitos que serán implementados para componer el EMCS.

# Estimación

## Recursos

### Recursos críticos

### Infraestructura y herramientas

### Recursos humanos

Actualmente el equipo cuenta con 3 recursos humanos que dividen las tareas en los siguientes puntos:

* Ingeniería de sistemas.
* Ingeniería de software.
* Ingeniería de hardware.
* Ingeniería de documentación.

## Suposiciones

## Restricciones

## Riesgos

## Resolución de problemas

# Planeación

## Roles y responsabilidades

* **Equipo scrum:**
  + Dueño de producto: Cliente (asesor) al cuál se le entregará el producto final terminado, integrado y con funcionalidad completa.
  + Equipo de desarrollo: Ingenieros que realizan el trabajo de entregar un incremento del producto “terminado” y que potencialmente se puede poner en producción al final de cada periodo de trabajo o *sprint*.
  + Scrum master: Líder que ayuda a las personas externas el equipo a entender qué interacciones con el equipo pueden ser útiles y cuáles no. Maximiza el valor creado por el equipo al modificar dichas interacciones.

## Trazabilidad de entregables

Los entregables liberados hacia el cliente deberán encontrarse en estado final de entrega y traceados correctamente con el fin de saber cada una de las implementaciones que componen el proyecto.

## Administración de riesgos

## Análisis de requisitos de software

El análisis de requisitos del software se realizó de acuerdo a los requerimientos del cliente, los cuales a su vez, fueron traducidos a requisitos de sistema, indicando su debida canalización a las diferentes disciplinas que componen un sistema embebido.

El documento provisto por el cliente se encuentra embebido en el siguiente documento:



El documento con los requisitos de sistemas se encuentra en el siguiente objeto embebido:



## Diseño de arquitectura de sowtfare

### Diagrama de bloques de SW

### Diagrama de control

#### Entradas

#### Salidas

#### Ruido

#### Sistema

#### Flujo de datos

### UML

#### Diagrama de estados

#### Diagrama de tiempos

### Árbol de llamadas

### Diagramas de flujo

### Final: ROM, RAM, STACK, Mapa de Memoria

### Throughput

### Complejidad de Código

# Construcción del software

## Aplicación de estándares de codificación y convenciones

### Nomenclatura

# Pruebas de software

## Pruebas de integración

## White Box Test

## Gray Box

## ABA test

# Liberación de versiones de software

## Git

# Verificación y aseguramiento de la calidad

## Prueba funcional y validación en campo

## Calendario de revisión.

# Referencias