**UNIVERSIDAD TECNOLÓGICA DE QUERÉTARO**

**CESEQ**



**Diplomado en Software Embebido**

Proyecto Integrador: Control de velocidad de motor CD

DOCUMENT: Software Development

Document No. #CESEQ\_SDP\_001

Scrum Máster: Francisco, Díaz

Developers: Francisco, Díaz

Gibrán, Pedraza

Tester: Alejandra, Castro

Date (YYYYMMDD): 20190724

Version: 1.0.0.

Project Version: 1.0.10.

# Log

Document Version

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date (yyyymmdd) | Description | Reviewer |
| 1.0.0. | 20190405 | First release | Pérez, Adbeel |
|  |  |  |  |

Project Document Version

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date (yyyymmdd) | Description | Reviewer |
| 1.0.1. | 20190723 |  |  |
| 1.0.2. | 20190724 |  |  |

# Index

Table of Contents

[1. Log 2](#_Toc8215461)

[2. Index 3](#_Toc8215462)

[3. Project Scope 4](#_Toc8215463)

[4. Deliverables 4](#_Toc8215464)

[5. Development methodology 5](#_Toc8215465)

[6. Estimates 7](#_Toc8215466)

[7. Planning 11](#_Toc8215467)

[8. Solving Problem Strategy 11](#_Toc8215468)

[9. Design 11](#_Toc8215469)

[9.1. Standards 12](#_Toc8215470)

[9.2. Naming conventions 13](#_Toc8215471)

[10. Testing 13](#_Toc8215472)

[10.1. Verification strategy (black box test) 13](#_Toc8215473)

[10.2. White box strategy 13](#_Toc8215474)

[10.3. Cyclomatic Complexity Redundance index 13](#_Toc8215475)

[11. Release 13](#_Toc8215476)

[11.1. Software Development Folder 14](#_Toc8215477)

[11.2. Integration Tests Strategy 14](#_Toc8215478)

[11.3. Validation Testing / Functional Testing 14](#_Toc8215479)

[11.4. Throughput and Flash and RAM measurement 14](#_Toc8215480)

[12. Results 14](#_Toc8215481)

[13. Lessons Learned 14](#_Toc8215482)

# Project Scope

El alcance de este proyecto es aplicar durante el desarrollo todas las metodologías, estrategias, herramientas, diseño del software y métodos de prueba vistas durante el diplomado de software embebido.

**Definición del problema**.

Controlar la velocidad de un motor de corriente directa mediante la aplicación de una señal cuadrada que varía en su ancho de pulso y cuya frecuencia de trabajo

La frecuencia de trabajo **debe** estar en un rango de f=1KHz a f=10KHz (esta frecuencia puede ser modificada para obtener dentro del rango para mejorar la señal de retroalimentación del motor).

Una vez seleccionada la frecuencia de trabajo (señal del sensor de efecto hall con menor ruido), esta **debe** ser fija, variando únicamente el “*duty cycle*”.

Mediante el uso de un sensor de efecto hall acoplado al rotor del motor se **debe** medir la velocidad del motor el cual proveerá una serie de pulsos cada que se complete una vuelta completa, esta medición.

Así, a mayor velocidad del motor, mayor será el número de pulsos leídos y mientras menor sea la velocidad, menor será el número de pulsos.

El voltaje de alimentación de la tarjeta de potencia **debe** ser de 12 Volts.

El motor **debe** seguir el valor de referencia o “*SetPoint*” (velocidad deseada), el cual estará dado por una entrada de la tarjeta de control.

La pantalla LCD o interfaz gráfica **debe** mostrar la velocidad del motor y SetPoint (ambos en RPM’s); así como el porcentaje de trabajo de la señal cuadrada.

Para más detalles del proyecto consulte el siguiente documento:

Documentacion/1) Requirements/stakeholder

**Requisitos del Proyecto.**

Este documento de Requisitos del Proyecto también puede ser encontrado en el siguiente archivo.

Documentacion/1) Requirements/3. SWRA\_20190802.xlsx

# Deliverables

A continuación, se menciona la lista de los entregables tanto de documentación como de código, así como su ubicación.

Codigo:

[www.github.com/JPacho/ProyectoIntegrador](http://www.github.com/JPacho/ProyectoIntegrador)

Documentacion:

[www.github.com/JPacho/ProyectoIntegrador\_Documentacion](http://www.github.com/JPacho/ProyectoIntegrador_Documentacion)

1) Requirements

* 3. SWRA\_20190802.xlsx

2) Planning

* 7. Planning\_20190802.xlsx
* 8. DFMEA\_20190802.xlsx

3) Design

* 9. SoftwareDesignDocument\_20190809.docx
* 9.1. SoftwareStandards\_20190809.docx
* 9.2. NamingConventions \_20190809.docx

4) Verification

* 10.1. BlackboxTest\_baseline\_20191025.docx
* 10.2. WhiteboxTest\_baseline\_20191025.docx
* 10.3. CCR\_baseline\_20191025.docx
* 11.1. IntegrationTesting\_baseline\_20191025.docx
* 11.2. ValidationTesting\_baseline\_20191025.docx
* 11.3. ThroughputRAMFlash\_procedure\_20191025.docx

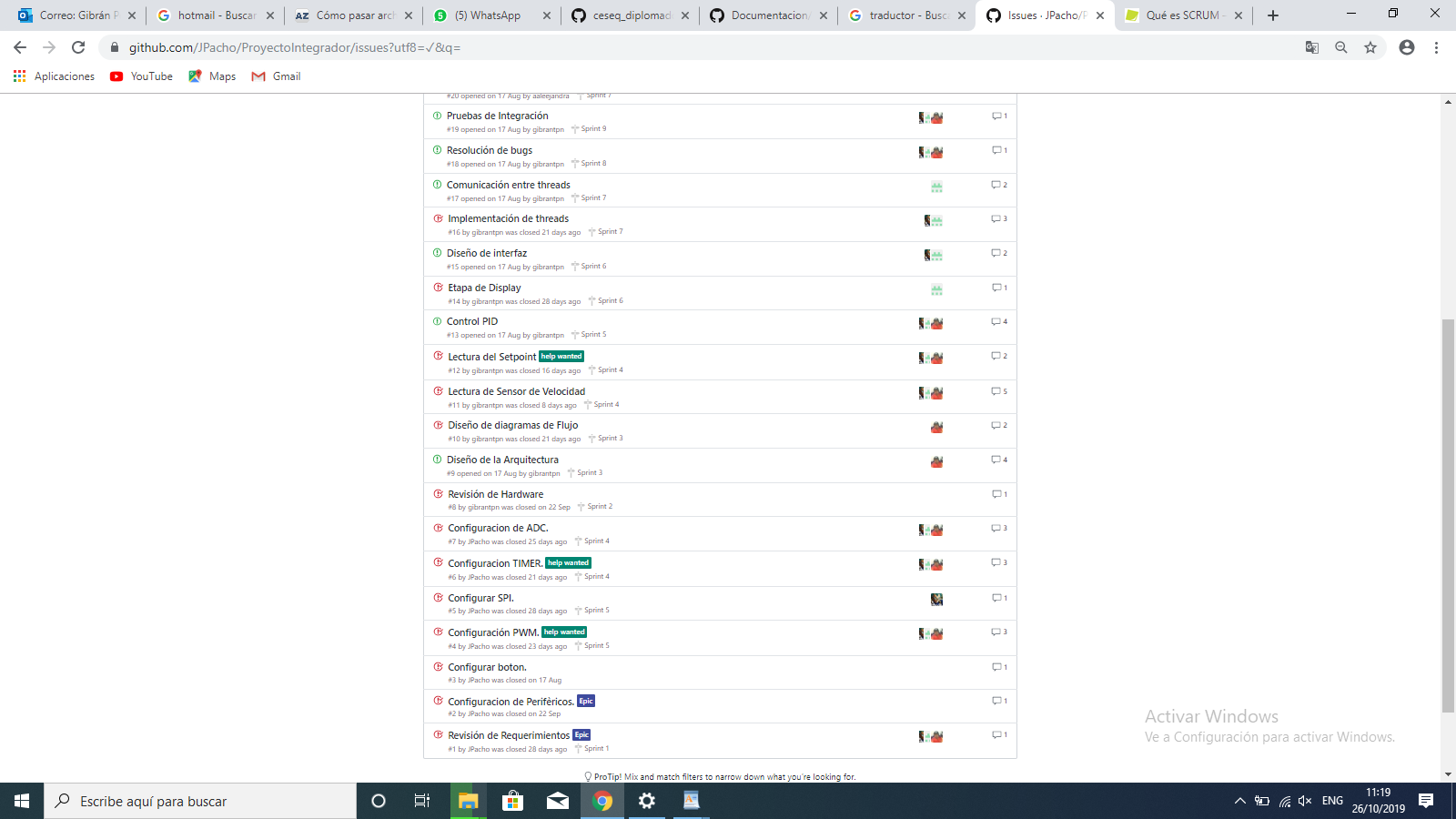
4) Verification\Results

* 10.1. BlackboxTest\_20191026.docx
* 10.2. WhiteboxTest\_20191026.docx
* 10.3. CCR\_20191026.docx
* 11.1. IntegrationTesting\_20191026.docx
* 11.2. ValidationTesting\_20191026.docx

# Development methodology

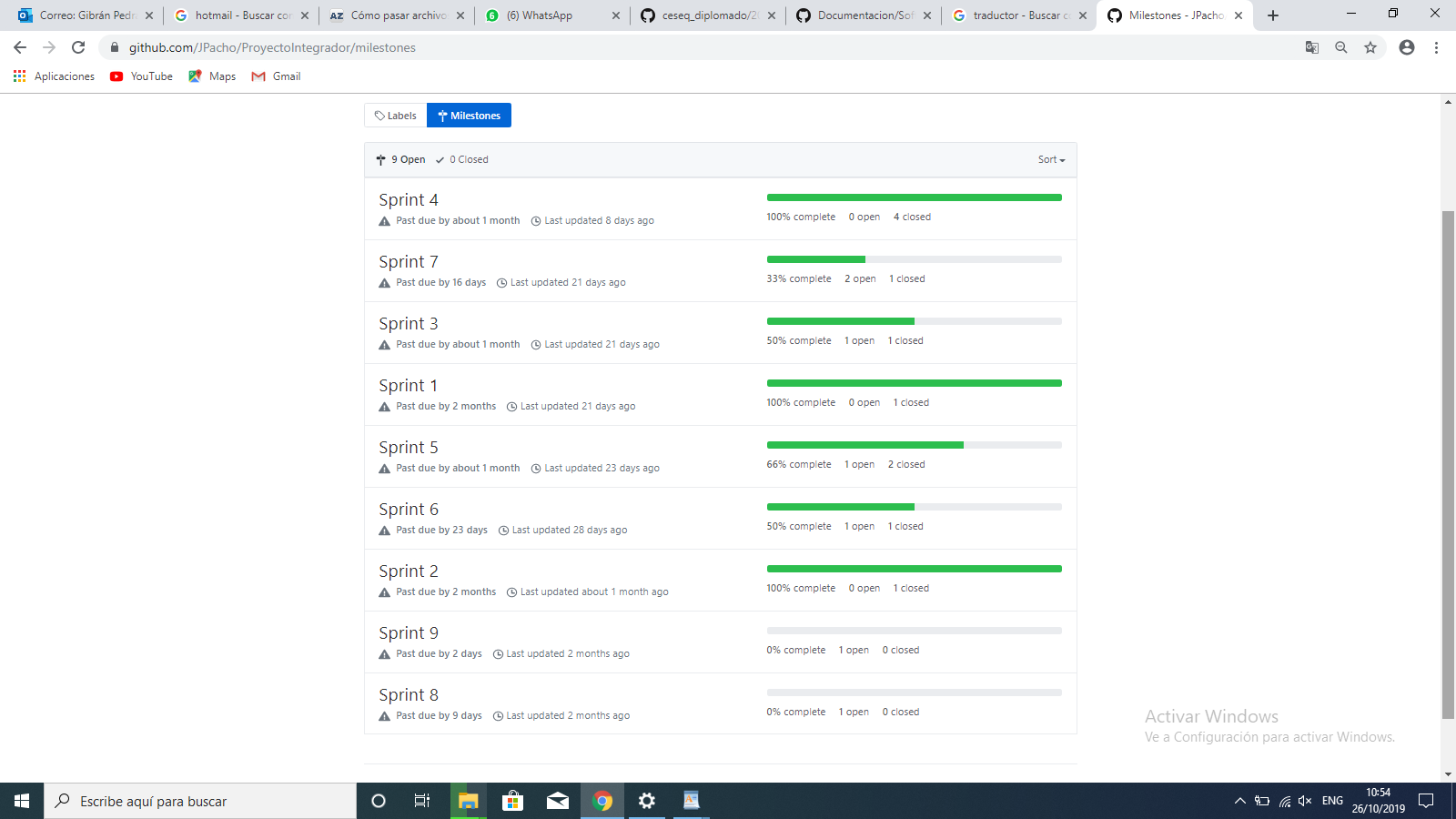
Como metodología de desarrollo se ocupó el Ágil Scrum, como herramienta de control de versiones se utiliza GitHub

**SCRUM BOARD**



**SPRINTS**

La duración de los sprint para el desarrollo fue de uno a dos sesiones



**CALENDARIO DE LAS REUNIONES**

las reuniones se realizaban diariamente para verificar el avance de cada sprint.

**POSICIONES**

Scrum Máster: Francisco Díaz Colorado

Developers: Francisco Díaz Colorado, Gibrán Pedraza Monroy

Tester: Marta Alejandra Castro Morales

Product owner:

**JUNTA DE PLANEACION**

Las juntas se realizaron al final de cada sprint para la asignación de tickets entre los desarrolladores, en estas también se discutía la metodología a seguir durante el proyecto.

# Estimates

**HW FACTS:**

* **Tarjeta de desarrollo:** Se conto con una tarjeta de desarrollo Renesas Synergy modelo SK-S7G2, el kit contenía su cable USB para alimentación y conexión a la PC.

****

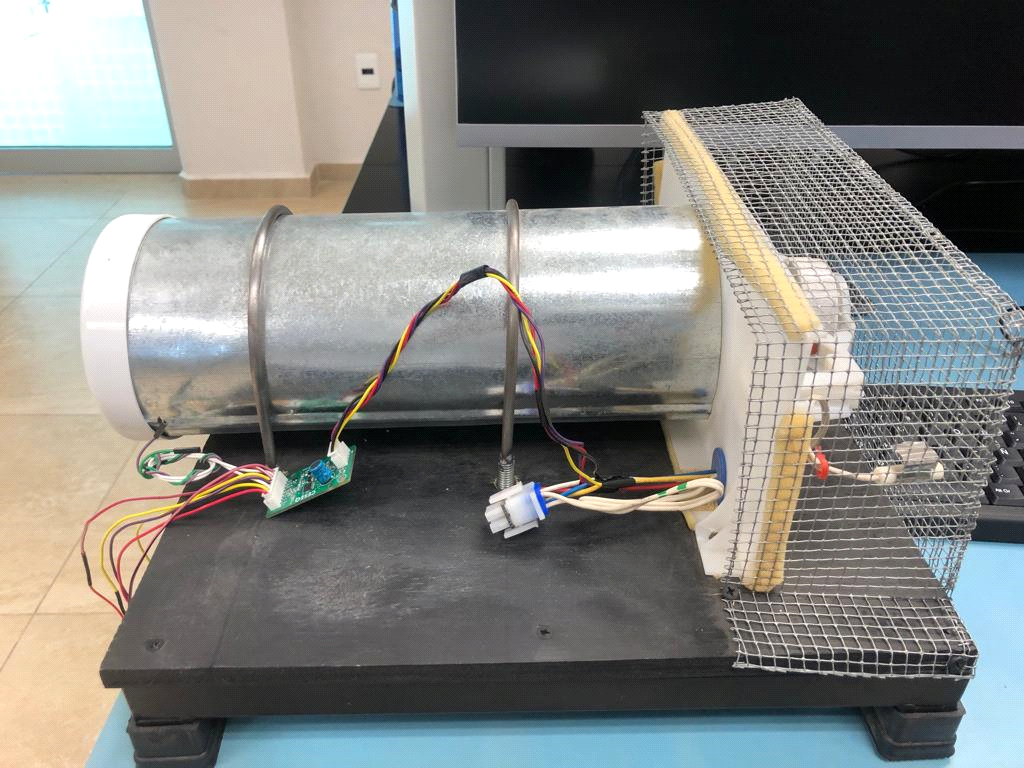
* **La planta de control:**

Componentes:

Motor DC

Sensor de efecto Hall integrado

Tarjeta de potencia



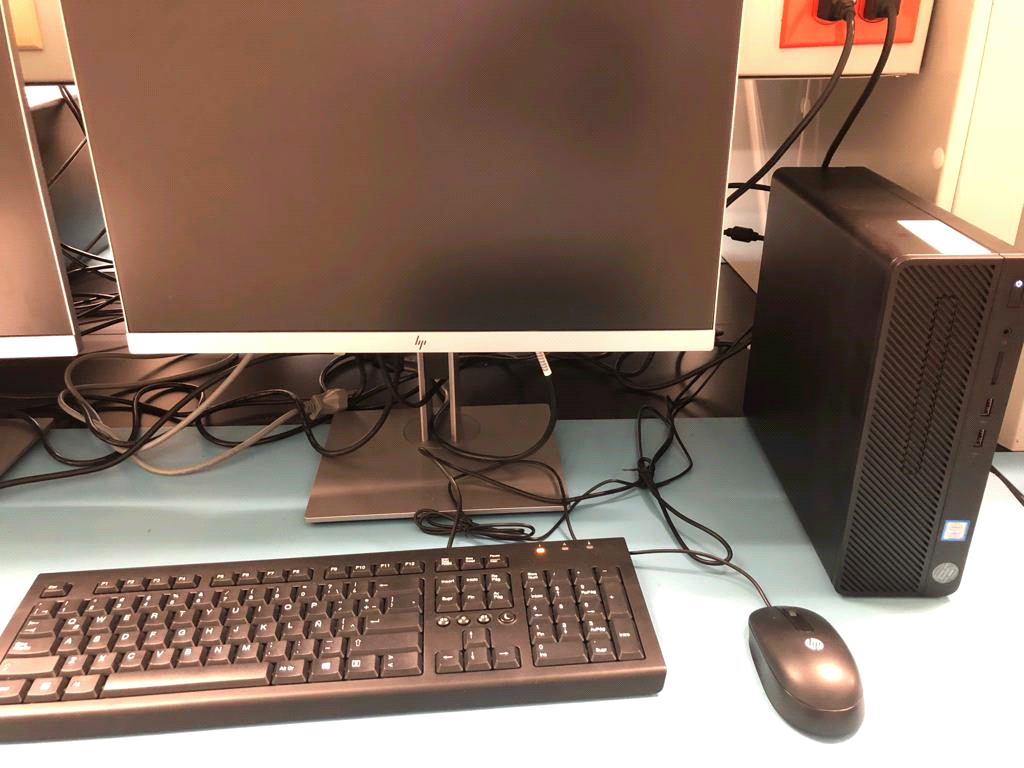
* **Computadora de escritorio:** Se dispone de una computadora de escritorio marca HP con las siguientes características:

SO: Windows 10

RAM: 32 GB

Procesador: i7-8700

Sistema: 64 Bits



* **Osciloscopio:** Se cuenta con un equipo marca Tektronix modelo TBS1102B-EDU

****

* **Generador de señal:** Se cuenta con un equipo marca Tektronix modelo AFG1062

****

* **Multímetro:** Se cuenta con un equipo Marca Keithley modelo 2110 5 1/2

****

* **Fuente de alimentación:** Se cuenta con un equipo Keithley 2231A-30-3

**Imagen que contiene electrónica

Descripción generada automáticamente**

**ACTIVITIES FACTS:**

* **Recursos Humanos.**
  + El product owner del proyecto integrador fue el tutor de nuestro equipo el Doc.
  + El rol de Scrum Máster estuvo a cargo del ing.
* **Numero de desarrolladores:** el número de desarrolladores se decidió por 2 miembros del equipo.

**SW FACTS:**

* **Sistema de operación:** Threadx

**HW ASSUMPTIONS:**

* **Hardware dañado:** Los componentes de Hardware se entregaron sin ningún daño, si acaso algunos cables dañados, los cuales fueron reemplazados, por otra parte, la falta de puntas para osciloscopio.
* **Disponibilidad de tiempo de laboratorio:** El tiempo en laboratorio fue amplio, pero realmente el tiempo para realizar el desarrollo en él fue escaso, ya que tanto los requerimientos como la formación de equipos tardo bastante, pasado la mitad del tiempo del curso, por otro lado, se calendarizaron los tiempos para desarrollo los días, viernes pasando medio día como los sábados, lo cual no se cumplía. Los avances que se realizaron en el código se lograron los días que se asistía entre semana al laboratorio.

El equipo no tuvo mucho tiempo de desarrollo ya que cada miembro de este pertenece a una empresa diferente y era muy difícil coincidir en horarios.

**ACTIVITIES ASSUMPTIONS:**

* **Disponibilidad de tiempo del equipo:** La disponibilidad del equipo se limitó a pasado la mitad del curso, ya que, al no tener los requerimientos ni equipo asignado, nada se podía hacer. De igual manera se limitaban a las tardes entre semana, así como unas pocas horas en horario del curso.
* **Hardware en buenas condiciones:** En cuanto a los equipos destinados al desarrollo sus condiciones eran buenas por no decir excelentes.

**HW ASSUMPTIONS:**

* **Lenguaje de programación, SW IDE o plataforma HW desconocida:** en cuanto a lenguaje de programación por ser C++, realmente ningún miembro del equipo tubo ninguna dificultad.

En tanto al Hardware utilizado no todos miembros del equipo estaban familiarizados con estas herramientas, puesto que su área de desarrollo no es enfocada a la electrónica.

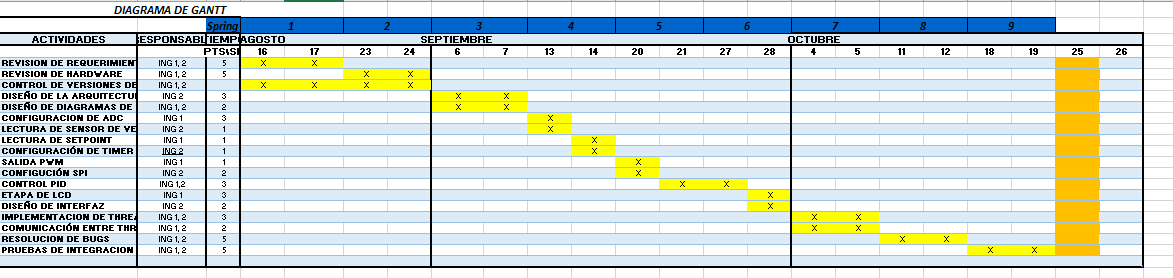
**Modulo SW desconocido:** El ambiente de desarrollo de Renesas e2 studio es nuevo para el equipo de trabajo y ya que cuenta con una gran cantidad de herramientas como el Renesas Synergy, fue de cierta dificultad para comprenderlos, aunque una vez que se entiende es fácil de usar.

Algunos de los riesgos que presenta el proyecto se refieren a la perdida de clases y por ende de hora de labor en el proyecto debido a actividades laborales. Así como días de asueto oficiales.

El aprender a utilizar el software de control de versiones es algo nuevo para los integrantes del equipo, esto podría provocar confusión en uso o el surgimiento de problemas inesperados.

Se conto con un caso inusual de un compañero que cambio de trabajo y se le dificultaba presentarse en las sesiones de los viernes.

Se crea un diagrama Gantt con el desglose de todas las actividades a realizar.



# Planning

Se realizó un diagrama de Gantt con las actividades del proyecto para realizar.

No siempre se siguió al pie de la letra ya que algunas actividades se llevaron mas tiempo del estimado

El documento se puede encontrar en la ruta siguiente, dentro de la carpeta principal del proyecto integrador.

Documentacion/3) Planning/7. Planning\_20190802.xlsx.xlsx

# Solving Problem Strategy

En esta sección se presenta el análisis de Modos de Falla y Efectos (FMEA) por sus siglas en inglés, dicho documento puede ser encontrado en el siguiente archivo:

Documentacion/2) Planning/8. DFMEA\_20190802.xlsx

# Design

En esta sección se muestran los diagramas de Flujo del diseño del software.

Dicho documento puede ser encontrado en el siguiente archivo:

Documentacion/2) Design/9. SoftwareDesignDocument\_20190809.docx

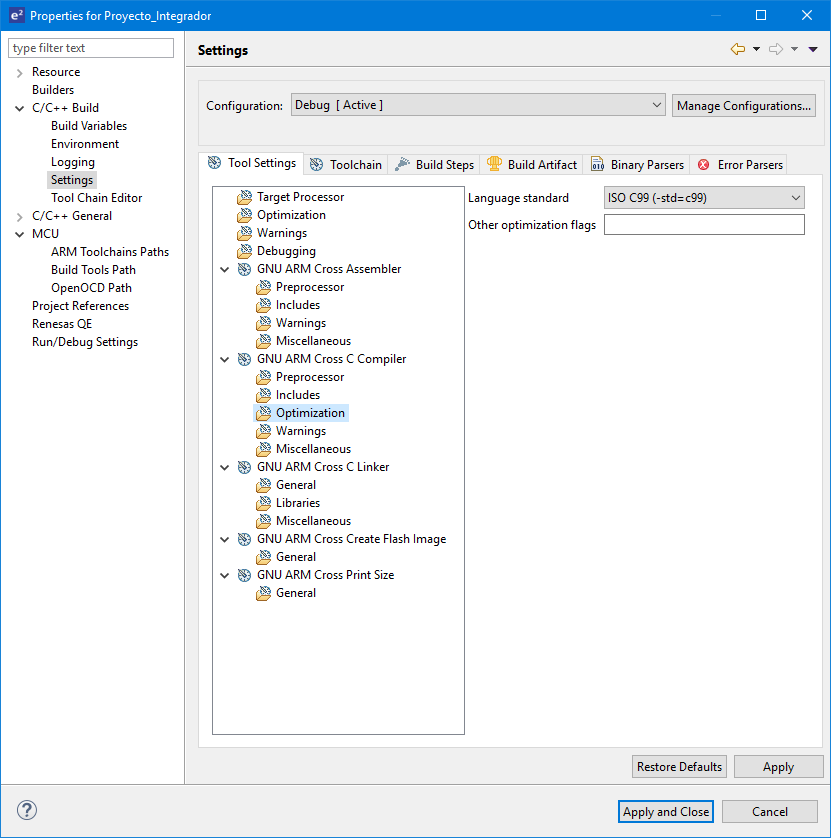
## Standards

Se siguió la norma C99 para codificar el software del proyecto integrador.

La norma puede consultarse en el siguiente enlace:

<http://www.open-std.org/jtc1/sc22/WG14/www/docs/n1256.pdf>

La herramienta usada para aplicar este estándar fue el software de entorno de desarrollo Renesas e2 studio. El software de desarrollo e2 studio de Renesas, tiene en el compilador ARM C Cross Compiler una configuración basada en la norma C99, la cual evalúa el código buscando que cumpla con tal norma.



## Naming conventions

El propósito del presente documento es presentar el estándar que se debe de utilizar para codificar un programa en lenguaje C, independientemente de la plataforma que se trate se deben de seguir ciertas reglas al momento de desarrollar software, el objetivo de estas es que el programa sea claro y explícito.

El documento se puede encontrar en la siguiente ruta, dentro de la carpeta principal del proyecto integrador.

Documentacion/3) Design/9.2. NamingConventions\_20190809.docx

# Testing

## Verification strategy (black box test)

## White box strategy

## Cyclomatic Complexity Redundance index

# Release

En esta sección se define el manejo de control de cambios y de versiones del software, así como la convención de nombrado del software y de la documentación.

Para el nombre del software se tiene el siguiente formato o convención de nombrado

<Nombre del Programa>\_<Nombre del Proyecto>\_<Tipo de Liberación>\_<Sprint><Versión>

De aquí se obtiene que el nombre del software queda como se muestra a continuación:

**DSE\_MTRCTRL\_DEV\_0100**

Donde:

DSE = Desarrollo de Software Embebido

MTRCTRL = Control Motor de CD

DEV = Desarrollo (ENG = Ingeniería, PROD = Producción)

01 -> Sprint 01

00 -> Versión 00

Durante el desarrollo del programa se ha mantenido constante la leyenda DEV mientras no se lance a producción que para nuestro caso no aplica, ya que el alcance de este proyecto es solo desarrollo, los que sí están iterando son el sprint y la versión.

Para la convención de nombrado del documento se tiene la siguiente nomenclatura:

**Software Development Plan\_YYYYMMDD\_1.0.7.docx**

Donde:

YYYY = Año

MM = Número del mes

DD = Día

1.0.7 -> Versión

En esta nomenclatura de nombrado cambia tanto la fecha como la versión

En el siguiente enlace se encuentra el repositorio virtual el cual se usó para llevar el control de versiones de la documentación.

[www.github.com/JPacho/ProyectoIntegrador\_Documentacion](http://www.github.com/JPacho/ProyectoIntegrador_Documentacion)

## Software Development Folder

En el siguiente enlace se encuentra el repositorio virtual el cual se usó para llevar el control de versiones del Proyecto integrador.

[www.github.com/JPacho/ProyectoIntegrador](http://www.github.com/JPacho/ProyectoIntegrador/tree/documentacion)

## Integration Tests Strategy

## Validation Testing / Functional Testing

## Throughput and Flash and RAM measurement

En esta sección se presentan las mediciones de uso de CPU, así como la medición de uso de memoria (RAM, flash), el stack y mapa de memoria.

La información de esta sección puede ser encontrada en la siguiente ruta, dentro de la carpeta principal del proyecto integrador.

Documentacion/4) Verification/ 11.3. ThroughputRAMFlash\_procedure

# Results

En esta sección se muestran evidencias de los resultados obtenidos con la planta del Proyecto integrador, se encuentran fotografías.

Esta sección se encuentra en la carpeta Documentacion/6) Results

# Lessons Learned

Este diplomado me sirvió para reforzar algunos temas y conceptos de software embebido, además me pareció muy interesante toda la parte de administración del proyecto, como el redactar requerimientos, llevar un control de todo el desarrollo de software embebido, así como asegurar la calidad del producto.

Francisco, Díaz

En este diplomado aprendí cosas muchas cosas nuevas. Nunca había programado para equipos embebidos, ni había visto la interacción de hardware-software de esta manera.