**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 Master: Solís, Gabriel

Developer. Pintor, Antonio

Date (YYYYMMDD): 20190724

Version: 1.0.0.

Project Version: 1.0.1.

# 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 | Modified “6. Estimates” | Solis, Gabriel |
| 1.0.2. | 20190724 | Modified “5”, “6” | Solis, Gabriel |
| 1.0.3 | 20190729 | Modified “6. Estimates” | Solis, Gabriel |
| 1.0.4. | 20190803 | Modified “5”, “6”, “10.3” | Solis, Gabriel |
| 1.0.5. | 20190805 | Added “3. Project scope”, “8. Solving Problem Strategy, “9. Design”, “9.2 Naming conventions” | Pintor, Antonio |
| 1.0.6. | 20190806 | Updated “3, 8 and 9” | Pintor, Antonio |
| 1.0.7. | 20190807 | Added section “11.4” | Pintor, Antonio |
| 1.0.8. | 20190807 | Modified “9”, “9.1”, “11.1”, “11.2” | Solis, Gabriel |
| 1.0.9 | 20190808 | Added section 10.1 and 10.2 | Pintor, Antonio |
| 1.0.10 | 20190808 | Modified “11.1”, “11.2” | Solis, Gabriel |

# Index

Table of Contents

[1. Log 2](#_Toc8215461)

[2. Index 3](#_Toc8215462)

[3. Project Scope 4](#_Toc8215463)

[4. Deliverables 4](#_Toc8215464)

[5. Development methodology 4](#_Toc8215465)

[6. Estimates 4](#_Toc8215466)

[7. Planning 5](#_Toc8215467)

[8. Solving Problem Strategy 6](#_Toc8215468)

[9. Design 6](#_Toc8215469)

[9.1. Standards 6](#_Toc8215470)

[9.2. Naming conventions 6](#_Toc8215471)

[10. Testing 7](#_Toc8215472)

[10.1. Verification strategy (black box test) 7](#_Toc8215473)

[10.2. White box strategy 7](#_Toc8215474)

[10.3. Cyclomatic Complexity Redundance index 7](#_Toc8215475)

[11. Release 8](#_Toc8215476)

[11.1. Software Development Folder 8](#_Toc8215477)

[11.2. Integration Tests Strategy 8](#_Toc8215478)

[11.3. Validation Testing / Functional Testing 8](#_Toc8215479)

[11.4. Throughput and Flash and RAM measurement 9](#_Toc8215480)

[12. Results 9](#_Toc8215481)

[13. Lessons Learned 9](#_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 **debe** ser constante.

La frecuencia de trabajo **debe** estar en un rango de *f* = 100 Hz a *f* = 1 KHz (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:

<PROJECT\_PATH>\1) Requirements\stakeholder

**Requisitos del Proyecto.**

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

<PROJECT\_PATH>\ 1) Requirements\3. SWRA\_20190405.xlsx

# Deliverables

En esta sección se menciona la lista de los entregables tanto de documentación como de código.

Estos documentos son encontrados en las siguientes carpetas para su revisión:

1) Requirements

* 3. SWRA\_20190405.xlsx

2) Planning

* 7. Planning\_20190405.xlsx
* 8. FMEA\_20190405.xlsx

3) Design

* 9. SoftwareDesignDocument\_20190405.docx
* 9.1. SoftwareStandards\_20190405.docx
* 9.2. NamingConventions \_20190405.docx

4) Verification

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

4) Verification\Results

* 10.1. BlackboxTest\_20190720.docx
* 10.2. WhiteboxTest\_20190720.docx
* 10.3. CCR\_20190720.docx
* 11.1. IntegrationTesting\_20190720.docx
* 11.2. ValidationTesting\_20190720.docx

# Development methodology

Durante el proyecto se utilizó para llevar el orden de tareas la metodología Agile Scrum, usando como herramienta la extensión diseñada para agile de GitHub, llamada ZenHub, en ella se realizó la creación de tickets, épicas y sprints.

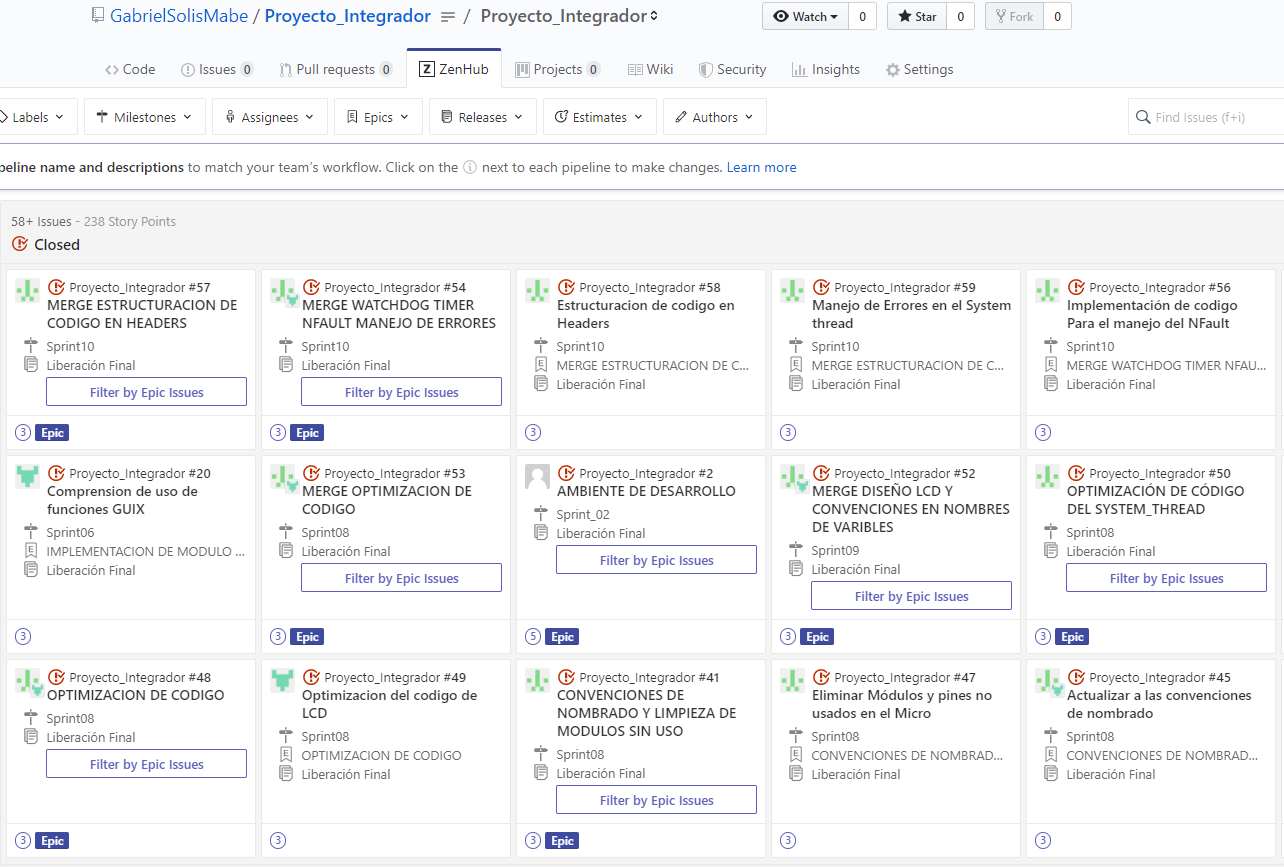
La duración de los sprints para la programación fue de pocos días, dado que la herramienta fue utilizada para llevar los cambios de software que hicimos, y dado que no comprendíamos el funcionamiento de la herramienta Git, comenzamos a usarla cuando el código estaba cercano a ser la versión final.

Posiciones: Scrum Máster: Gabriel Solís

Developers: José Antonio Pintor

Product owner: Marcos Samuel Peña

**SCRUM BOARD**

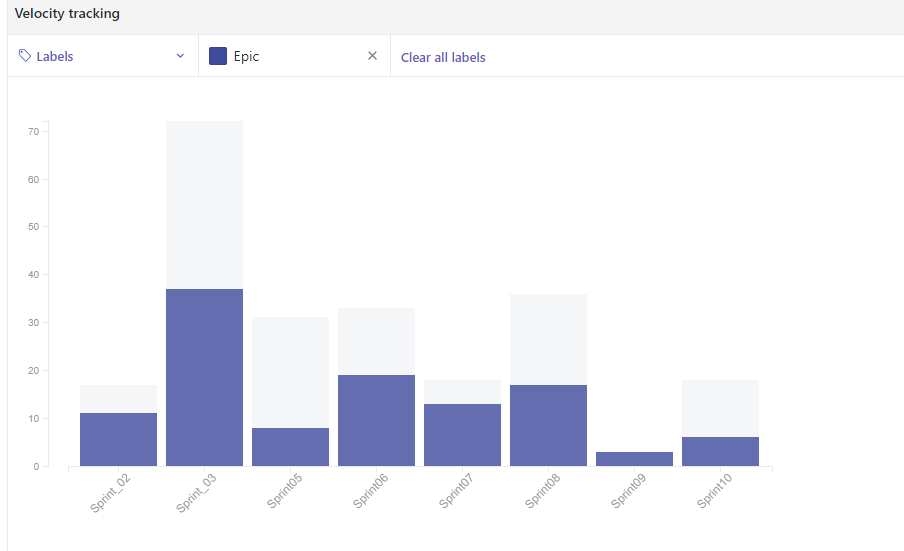


**SPRINTS**

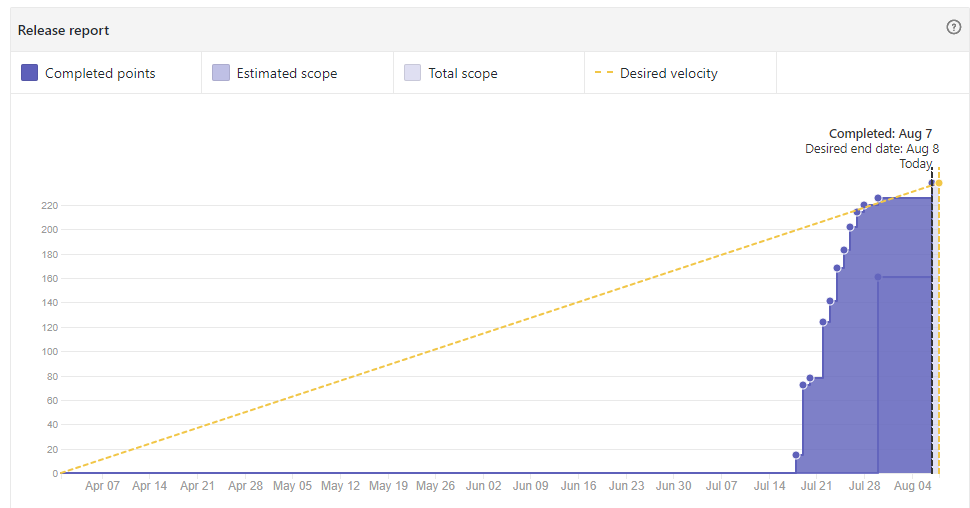
La duración del sprint fue de 2 días, ya que la mayoría del código lo teníamos implementado y listo para respaldar en la plataforma, de modo que el control de versiones se realizó en las últimas etapas de desarrollo del software, ya que no conocíamos el procedimiento de vinculación entre la plataforma GitHub y el software entorno de desarrollo Renesas e2 Studio.

Las juntas se realizaron diariamente para acelerar el proceso de la creación y asignación de tickets entre los desarrolladores, en estas también se discutía la metodología a seguir durante el proyecto.

**GRAFICA DE RASTREO DE VELOCIDAD.** Se muestran los puntos cerrados por cada sprint.



**RELEASE REPORT.**



# Estimates

**HECHOS.**

Tarjeta de desarrollo:

Se dispone de una tarjeta Renesas Synergy SK-S7G2, con su cable de datos USB para la conexión con la PC, transmisión de datos, alimentación eléctrica y el debuggeo de programas.

Planta de control:

Se dispone de una planta de control automático, la cual consta de un motor de DC, con aspas plásticas integradas como carga del motor, un sensor de efecto Hall integrado, una tarjeta de potencia y cables para su conexión con la tarjeta de control.

Computadora de escritorio:

Se cuenta con una PC Hewlett Packard, con potencia de procesamiento más que suficiente para ejecutar el software con el que trabajamos para desarrollar el software de control de velocidad requerido por el diplomado.

Dispositivos de medición y suministro de energía:

Se cuenta con dispositivos electrónicos que ayudaron a realizar las mediciones eléctricas necesarias para llevar a cabo el proyecto integrador. El equipo electrónico consta de un multímetro digital, una fuente de potencia con múltiples canales, un generador de funciones, y un osciloscopio para medir las señales eléctricas provenientes de las salidas del microcontrolador.

**SUPOSICIONES.**

Hardware dañado:

El caso de que el hardware estuviera dañado nos habría afectado de gran manera puesto que recibimos la planta de control de manera tardía y una demora más de tiempo por esta razón habría sido de gran impacto para el proyecto.

El tiempo de uso del laboratorio fue escaso las primeras veces que comenzamos a utilizarlo, ya que no contábamos con la tarjeta, la planta de control, o los cables de los dispositivos de medición, por lo tanto, no podíamos llevar a cabo actividades relacionadas a construir el proyecto. Fue semanas después que pudimos tener acceso completo al equipo al igual que trabajar en el laboratorio las tardes entre semana.

El equipo de trabajo, constituido por los dos desarrolladores anteriormente mencionados tuvieron bastante tiempo para trabajar en el proyecto, puesto que al trabajar en la misma empresa tienen comunicación todos los días laborales y pueden compartir ideas y soluciones para los problemas presentados.

El hardware que fue prestado se encontró en buen estado en general, a excepción de los cables de conexión de la tarjeta de potencia, ya que estos estaban rotos, sin terminal de conexión o incorrectamente ordenados según la hoja de datos de la tarjeta de potencia.

Software nuevo:

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 de compilación y debuggeo, será de cierta dificultad para ellos utilizarlas todas, sin embargo, las más comunes son sencillas de ocupar y muy intuitivas para el usuario.

El hardware no presenta demasiada dificultad de comprender puesto que se cuenta con una hoja de datos de conexión de la tarjeta de potencia, lo cual hará que su comprensión sea mejor.

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

**ESTIMADOS DE ACTIVIDADES.**

Recursos humanos:

Dado que seguimos una estrategia de desarrollo basada en Agile Scrum, el product owner del proyecto integrador fue el tutor de nuestro equipo el Ing. Marcos Samuel Peña, el rol de Scrum Master estuvo a cargo del ing. Gabriel Solís, quien también fungió como desarrollador al igual que el ing. Antonio Pintor.

Se contó con dos desarrolladores para el proyecto integrador aquí presentado.

Software:

Los desarrolladores cuentan con experiencia en la programación estructurada, sin embargo, no tienen conocimiento de las sentencias de código utilizadas en concreto con este microcontrolador, por lo tanto, deberán invertir tiempo en la lectura, investigación y análisis de ejemplos (notas de aplicación) de las funciones que requiere el microcontrolador para funcionar.

El uso del software de Renesas fue de utilidad a la hora de realizar la programación de la tarjeta de control, las herramientas que ofrece nos permitieron llevar a cabo el proyecto por medio de threads para realizar las tareas de control y del despliegue de información a través de la pantalla LCD con facilidad.

Documentación.

Entre los desarrolladores a cargo de este proyecto se repartieron tareas referentes a la documentación, aunque hubo tareas que debían desarrollarse por ambos ingenieros, dado que requerían de conocimientos acerca de cada parte de código que cada uno desarrolló.

**ESTIMADOS DE CÓDIGO.**

En cuanto a los módulos que se necesitan programar para el proyecto integrador, se encuentran el uso de un ADC, para la entrada de señal del potenciómetro que se usará para ajustar el valor de referencia del control, varios módulos de temporizadores, para llevar la cuenta de tiempo usado entre los flancos de subida y bajada del sensor de efecto hall, esto para obtener la frecuencia y velocidad del motor, un módulo de señal PWM, un módulo de interrupción que servirá de igual forma para la señal del sensor de velocidad, para la comunicación con la pantalla LCD se ocupara un módulo SPI y el procesamiento de gráficos estará a cargo del módulo GUIX.

# Planning

Se realizó un diagrama de Gantt con las actividades más importantes del proyecto para realizar, algunas llevaron más tiempo de lo planeado debido a dificultades con la investigación del código o la búsqueda de cómo implementarlo, o a la hora de hacer pruebas cada vez que implementábamos código.

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

<PROJECT\_PATH>\3) Planning\7. Planning\_20190405.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:

<PROJECT\_PATH>\2) Planning\8. DFMEA\_20190405.xlsx

# Design

En esta sección se muestran los diagramas de Flujo del diseño del software, así como un diagrama de estados del encendido y apagado del Motor de DC, también se muestra el diagrama a bloques del hardware que visualiza las conexiones entre la tarjeta de Renesas, el módulo de potencia del Motor, el motor, el potenciómetro, el LCD, los interruptores y los LEDs.

Dicho documento puede ser encontrado en el siguiente archivo:

<PROJECT\_PATH>\3) Design\9. SoftwareDesignDocument\_20190405.docx

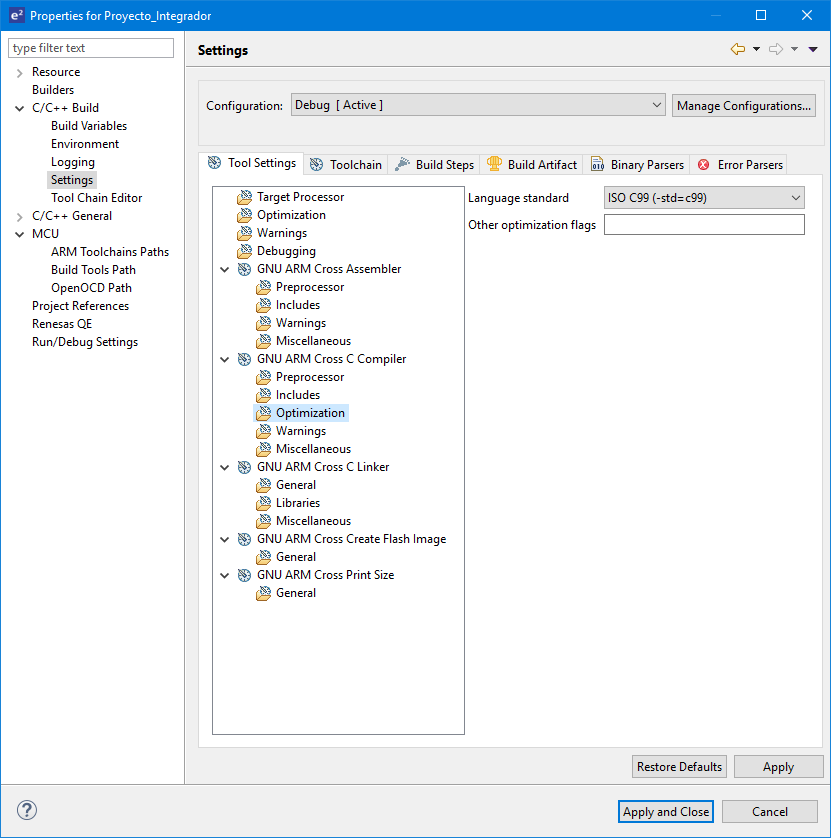
## Standards

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.

<PROJECT\_PATH>\3) Design\9.2. NamingConventions\_20190405.docx

# Testing

## Verification strategy (black box test)

En esta sección se presenta la estrategia que se eligió para la verificación mediante las pruebas del black box, el cual se encuentra en la siguiente ruta, dentro de la carpeta principal del proyecto integrador.

<PROJECT\_PATH>\4) Verification\10.1. BlackboxTest\_baseline.docx

así también como los resultados que se encuentran en la siguiente ruta:

<PROJECT\_PATH>\4) Verification\Results\10.1. BlackboxTest\_20190405.docx

## White box strategy

En esta sección se presenta la estrategia que se eligió para la verificación mediante las pruebas del black box, el cual se encuentra en la siguiente ruta, dentro de la carpeta principal del proyecto integrador.

el cual se encuentra en la siguiente ruta, dentro de la carpeta principal del proyecto integrador.

<PROJECT\_PATH>\4) Verification\10.2. WhiteboxTest\_baseline.docx

así también como los resultados que se encuentran en la siguiente ruta:

<PROJECT\_PATH>\4) Verification\Results\10.2. WhiteboxTest\_20190405.docx

## Cyclomatic Complexity Redundance index

Mediciones de complejidad de código y optimización.

Esta sección está contenida en la siguiente dirección:

<PROJECT\_PATH>\4) Verification\ 10.3. CCRI\_20190405.docx

Los resultados de esta sección están contenidos en la siguiente dirección:

<PROJECT\_PATH>\4) Verification\Results\10.3. CCRI\_20190405.docx

# 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\_1.0.9.docx**

Donde:

YYYY = Año

MM = Número del mes

DD = Día

1.0.9 -> 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.

<https://github.com/GabrielSolisMabe/Documentacion.git>

## 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.

<https://github.com/GabrielSolisMabe/Proyecto_Integrador.git>

## Integration Tests Strategy

Pruebas de integración tipo ascendente realizadas al combinar módulos de funcionalidad del software.

Esta sección está contenida en la siguiente dirección:

<PROJECT\_PATH>\4) Verification\11.1. IntegrationTesting\_baseline.docx

Los resultados de esta sección están contenidos en la siguiente dirección:

<PROJECT\_PATH>\4) Verification\Results\11.1. IntegrationTesting\_20190405.docx

## Validation Testing / Functional Testing

Esta sección se encuentra documentada en la dirección siguiente:

<PROJECT\_PATH>\4) Verification\11.2. ValidationTesting\_baseline.docx

Sus resultados se encuentran en la dirección siguiente:

<PROJECT\_PATH>\4) Verification\Results\11.2. ValidationTesting\_20190405.xls

## Throughput and Flash and RAM measurement

En esta sección se presentan las mediciones de uso de CPU (Throughtput) así como la medición de uso de memoria RAM, uso de memoria 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.

<PROJECT\_PATH>\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 y un video demostrativo.

Esta sección se encuentra en la carpeta <PROJECT PATH>\6) Results

# Lessons Learned

Durante todo el diplomado se llevaron a cabo actividades que no había recibido en mi educación académica, se impartieron materias que me enseñaron nuevas formas de trabajar en equipo como lo fue la metodología Agile Scrum, se tuvo la oportunidad de tener cátedras muy complejas y especificas acerca del funcionamiento de los microcontroladores, al igual que algunas más básicas que sirvieron para consolidar conocimientos y rellenar grietas de conocimiento o comprender mejor conceptos que no se encontraban claros. En general los conocimientos obtenidos a lo largo del diplomado son algo nuevo que agregar a mi vida, y que serán de bastante de utilidad en el futuro cuando tenga que aplicar conocimientos para resolver problemas de software.