# Portada

[Date]

oscar

[company name]

Proyecto integrador de Diplomado de Software embebido

Portada

# Revisores y aprobaciones.

|  |  |  |
| --- | --- | --- |
| Nombre | Fecha | Firma |
| Olaf Tonatiuh  Oscar Ramirez | 22/07/19  22/07/19 |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Historial de cambios.

|  |  |  |  |
| --- | --- | --- | --- |
| Numero de revisión | Fecha | Nombre | Detalles |
| V1 | 14/03/2019 | Oscar Ramirez Diaz | Agregar Titulos y subtitulos |
| V2 | 22/07/2019 | Oscar Ramirez Díaz | Actualizacion de documentación |

# Índice

Contents

[1. Portada 0](#_Toc5312866)

[2. Revisores y aprobaciones. 1](#_Toc5312867)

[3. Historial de cambios. 1](#_Toc5312868)

[4. Índice 2](#_Toc5312869)

[5. Alcance del proyecto 4](#_Toc5312870)

[5.1. Identificación del proyecto 4](#_Toc5312871)

[5.2. Descripción general del proyecto 4](#_Toc5312872)

[5.3. Entregables con el cliente. Compromisos de entrega de los módulos del software. 4](#_Toc5312873)

[6. Metodologías de desarrollo 4](#_Toc5312874)

[6.1. Metodología de desarrollo de software seleccionada. 4](#_Toc5312875)

[6.2. Plan de entregas de software al cliente. 4](#_Toc5312876)

[6.3. Procedimientos de control. 4](#_Toc5312877)

[7. Estimación 4](#_Toc5312878)

[7.1. Recursos 4](#_Toc5312879)

[7.1.1. Recursos críticos(RAM, ROM, Througput). 4](#_Toc5312880)

[7.1.2. Infraestructura y herramientas. 4](#_Toc5312881)

[7.1.3. Recursos humanos. 4](#_Toc5312882)

[7.2. Suposiciones/restricciones/Riesgos/resolución de problemas 4](#_Toc5312883)

[7.2.1. Suposiciones 4](#_Toc5312884)

[7.2.2. Restricciones 4](#_Toc5312885)

[7.2.3. Riesgos 4](#_Toc5312886)

[7.2.4. Resolución de problemas 4](#_Toc5312887)

[8. Planeación 5](#_Toc5312888)

[8.1. Roles y responsabilidades. 5](#_Toc5312889)

[8.2. Trazabilidad de entregables. 5](#_Toc5312890)

[8.3. Administración de riesgos. 6](#_Toc5312891)

[8.4. Análisis de requisitos de software 8](#_Toc5312892)

[8.5. Diseño de arquitectura de software 9](#_Toc5312893)

[8.5.1. Diagrama de Bloques de SW 9](#_Toc5312894)

[8.5.2. Diagrama de Control 9](#_Toc5312895)

[8.5.3. UML 9](#_Toc5312896)

[8.5.4. Árbol de Llamadas 9](#_Toc5312897)

[8.5.5. DiagramasdeFlujo(ActualizartodareferenciaalTema5.5.1Arquitectura) 9](#_Toc5312898)

[8.5.6. Final: % Uso de ROM, % Uso RAM, STACK ,Mapa de Memoria 9](#_Toc5312899)

[8.5.7. Throughput 9](#_Toc5312900)

[8.5.8. Complejidad de Código (índice Cyclomatic Complexity) 9](#_Toc5312901)

[9. Construcción del software 10](#_Toc5312902)

[9.1. Aplicación de Estándares de codificación y convenciones 10](#_Toc5312903)

[9.1.1. Nomenclatura de archivos 10](#_Toc5312904)

[9.1.2. Nomenclatura de funciones 10](#_Toc5312905)

[9.1.3. Nomenclatura de clases (si aplica) 10](#_Toc5312906)

[9.1.4. Nomenclatura de variables 10](#_Toc5312907)

[9.2. Metodología de revisión de software 10](#_Toc5312908)

[10. Pruebas de software. 10](#_Toc5312909)

[10.1. Pruebas de integración 10](#_Toc5312910)

[10.2. White Box Test 10](#_Toc5312911)

[10.3. GrayBox (opcional) 10](#_Toc5312912)

[10.4. BlackBox Test 10](#_Toc5312913)

[10.5. ABA Test 10](#_Toc5312914)

[11. Liberación de versiones de software. 10](#_Toc5312915)

[11.1. GIT 10](#_Toc5312916)

[12. Verificación y aseguramiento de la calidad 10](#_Toc5312917)

[12.1. Prueba funcional y validación en campo 10](#_Toc5312918)

[12.2. Calendario de revisión: documental y código 10](#_Toc5312919)

# Alcance del proyecto

## Identificación del proyecto

Este proyecto se trata de a través de la metodología de desarrollo de software en V, GSP, Agile, Ingeniería de requisitos, control de código fuente distribuido, generar un proyecto que controle la velocidad de un motor por medio de un microcontrolador y se describirá en el punto 5.2 a fondo además de evidenciar documentar todo el proceso en el que consiste el desarrollo del proyecto.

## Descripción general del proyecto

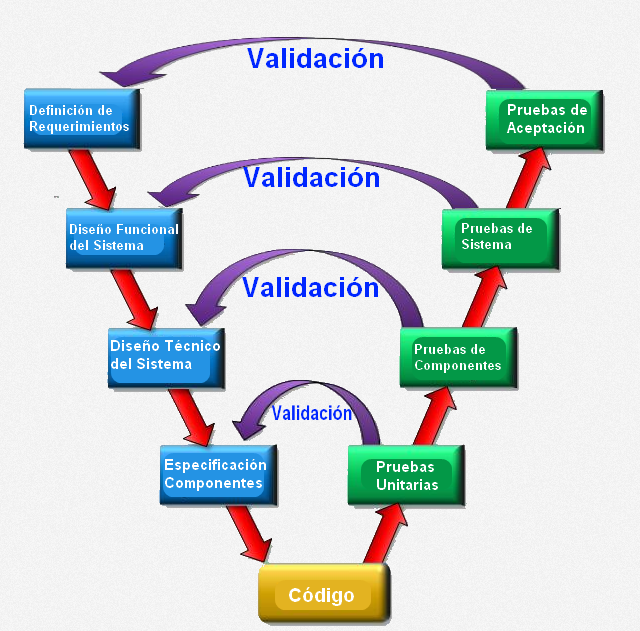
El proyecto de trata de desarrollar un control de un ventilador a través de un microcontrolador usando una interfaz de potencia la cual funciona con señal de un PWM como referencia de control, para poder visualizar la

## Entregables con el cliente. Compromisos de entrega de los módulos del software.

# Metodologías de desarrollo

## Metodología de desarrollo de software seleccionada.

La metodología de desarrollo de software a usarse será en V mesclada con agile ya que se basa en requisitos que entrega el cliente y a partir de ellos se generan requisitos hijos destinados a diferentes usos y competencias lo cual distribuye el trabajo de administración, revisión, implementación y verificación en diferentes personas, robusteciendo el proceso de generación de documentación y justificación de las decisiones técnicas del proyecto que recaen en diferentes encargados.



## Plan de entregas de software al cliente.

Refiérase a diagrama de Gantt adjunto la carpeta llamado Diagrama de gannt.mpp

## Procedimientos de control.

* Revisión cruzada de desarrollo de código
* Para cada cambio de código se analizaran los cambios por medio de juntas de revision
* Ejecución de test cases según entrega de código

# Estimación

## Recursos

### Recursos críticos(RAM, ROM, Througput).

ROM

LIbrerias del SO

Programa 273344 bytes

Constantes 8 bytes

Init data 1168 bytes

No init data 471328 bytes

Data 64 KB

Stack 8192 bytes

Others 464 Bytes

Ram 73 %

Flash 6%

ID-Codes 100%

### Infraestructura y herramientas.

* Control de Versiones de SW
  + GIT
  + Github
  + ZENHUB
* Control de Actividades y entregables
  + Microsoft Proyect
* IDE Desarrollo de SW
  + E2 studio
* Administración de requisitos
  + - Excel
  + Documentación general
    - Word

### Recursos humanos.

## Suposiciones/restricciones/Riesgos/resolución de problemas

### Suposiciones

Los ingenieros conocen del proceso de desarrollo de software ya que han estado trabajando previamente en una empresa de desarrollo de software.

Los ingenieros tienen estudios en áreas relacionadas como Mecatrónica / electrónica / software

Las fechas de los entregables se pacta al inicio del proyecto

### Restricciones

### Riesgos

* Dañar la tarjeta
* Dañar la planta con el motor
* Dañar el equipo de computo
* Dañar las instalaciones
* Enfermedad de uno de los 2 integrantes del equipo
* Renuncia o cambio de trabajo de uno de los 2 integrantes de trabajo
* Falta de conocimiento de uno de los 2 integrantes de trabajo
* Falta de tiempo para generar los entregables
* Proyectos en los trabajos con carga de trabajo fuerte que no permita que los ingenieros aporten al desarrollo del proyecto
* Que el cliente cambie la fecha de entrega al final

### Resolución de problemas

No hubo problemas

# Planeación

## Roles y responsabilidades.

* Project manager
* Desarrollador
* QA
* System Engineer
* Branch manager
* System Lead
* Sw Arquitect
* Scrum Master
* Gestor de proyecto
* Gestor de calidad
* Responsable de compras
* Project Manager
* Desarrollador de SW
* Analista de requisitos
* Scrum Master
* Integrador

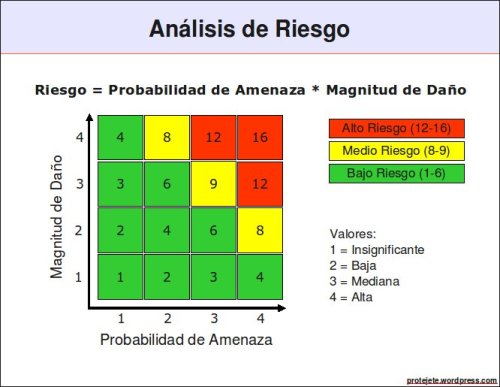
## Trazabilidad de entregables.

De acuerdo a lo planeado con el cliente se tienen diferentes entregables programados con diferentes fechas y con diferentes alcances

|  |  |  |
| --- | --- | --- |
| Fecha | Nombre de entrega | Características |
| ~~26/08/2019~~ | ~~Entrega de proyecto completo~~ | ~~Entrega de documentación completa y proyecto funcional~~ |
| 26/07/2019 | ~~Entrega de proyecto completo~~ | ~~Entrega de documentación completa y proyecto funcional~~ |
|  |  |  |

## Administración de riesgos.

Esta sección trata de análisis de riesgos en diferentes aspectos, además de en diferentes partes del cronograma del proyecto usando la teoría de análisis de riesgo que se muestra en la tabla





## Análisis de requisitos de software

Para una correcta identificación de requisitos se contempla la siguiente terminología para hacer mas entendible y además categorizar los requisitos de una manera más entendible y flexible

Requisitos de cliente

Req-cl-modulo-01(F/NF)

“Req” quiere decir quisito

“cl” quiere decir cliente

00 numero identificador del requisito

Identificación de requisitos de diseño por numero o identificador completo

Req-sys-LCD-01 (F/NF)

“Req” quiere decir quisito

“sys” quiere decir sistema

LCD- quiere decir modulo

00 numero identificador del requisito

Req-Arq-LCD-01 (F/NF)

“Req” quiere decir quisito

“arq” quiere decir arquitectura

LCD quiere decir modulo de software

00 numero identificador del requisito

Requisitos de impementacion

Req-sw-LCD-01 (F/NF)

“Req” quiere decir quisito

“sw” quiere decir software

LCD quiere decir modulo de software

00 numero identificador del requisito

Requisitos de Pruebas

Req-pr-LCD-01

Identificación de TIquets para tareas

### Requisitos de cliente

### Requisitos de sistema

### Requisitos de desempeño

### Requisitos de Diseño

### Requisitos de Arquitectura

### Requisitos Software

## Diseño de arquitectura de software

### Diagrama de Bloques de SW

### Diagrama de Control

* + - * Entradas
      * Salidas
      * Ruido
      * Sistema
      * Flujo de Datos

### UML

* + - * Diagrama de estados
      * Diagrama de Tiempos/Recursos

### Árbol de Llamadas

### DiagramasdeFlujo(ActualizartodareferenciaalTema5.5.1Arquitectura)

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

### Throughput

### Complejidad de Código (índice Cyclomatic Complexity)

# Construcción del software

## Aplicación de Estándares de codificación y convenciones

### Nomenclatura de archivos

### Nomenclatura de funciones

### Nomenclatura de clases (si aplica)

### Nomenclatura de variables

* + - * Variables locales
      * Variables globales
    1. Macros
    2. Tipos de datos
       - Estándares/primitivos
       - Definidos por el usuario
    3. Estructura de proyecto
    4. Normas aplicables

## Metodología de revisión de software

# Pruebas de software.

## Pruebas de integración

## White Box Test

## GrayBox (opcional)

## BlackBox Test

## ABA Test

# Liberación de versiones de software.

## GIT

# Verificación y aseguramiento de la calidad

## Prueba funcional y validación en campo

## Calendario de revisión: documental y código