

Requisitos del proyecto
Comité de Empresas de Software Embebido de Querétaro
CESEQ

DIPLOMADO DE SOFTWARE EMBEBIDO

REVISIÓN (2019/05/27)

REVISORES:

Uriostegui, Luis; Pérez, José; Pérez, Adbeel; Aguilar-Pereyra, Felipe
Reynoso, Daniel; Zuñiga, Gregorio; Donjuán, Eduardo; Vázquez, Germán; Santa Ana, Rafael;

Nota:

Este documento se está actualmente en fase estable, por lo que la última versión está en la siguiente dirección: <https://drive.google.com/drive/u/1/folders/0BydOs3Mey500RjJsV2IxWU1RNUk>

Se deberá enviar un correo a: faguilar@uteq.edu.mx para solicitar acceso a este repositorio.

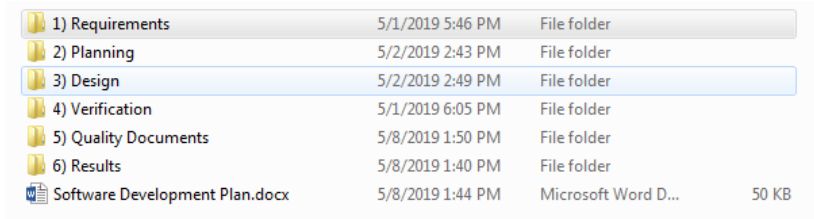
Tabla de Contenido

1	ENTREGABLES:	3
1.1.	Estructura de folders del proyecto	3
1.2.	Documentación del Proceso de desarrollo de software	3
1.3.	SW de control para Dispositivo CESEQ-001.	4
1.3.1.	Descripción del proyecto (Requisitos generales).	4
1.3.2.	Requerimientos funcionales específicos:	5
1.3.2.1.	Entradas	5
1.3.2.2.	Salidas	7
1.3.2.3.	Control	9
1.3.2.4.	Sistema Operativo/Calendarizador.	9
1.3.3.	Requerimientos no funcionales	10
1.3.3.1.	Apariencia del display.	10
1.3.3.2.	Diagramas de flujo.	11
2	INSUMOS:	11
3	INFRAESTRUCTURA:	11
4	PROCEDIMIENTOS:	11
5	REFERENCIAS BIBLIOGRÁFICAS:	13
6	CONTROL DE VERSIÓN DEL DOCUMENTO	13

1 ENTREGABLES:

1.1. Estructura de folders del proyecto

El proyecto **debe** mostrar la siguiente estructura:



1) Requirements	5/1/2019 5:46 PM	File folder	
2) Planning	5/2/2019 2:43 PM	File folder	
3) Design	5/2/2019 2:49 PM	File folder	
4) Verification	5/1/2019 6:05 PM	File folder	
5) Quality Documents	5/8/2019 1:50 PM	File folder	
6) Results	5/8/2019 1:40 PM	File folder	
Software Development Plan.docx	5/8/2019 1:44 PM	Microsoft Word D...	50 KB

Figura 1

Esta estructura **debe** ser descargada desde:

https://github.com/adpema79/ceseq_diplomado.git

1.2. Documentación del Proceso de desarrollo de software

El primer documento en el que se **debe** trabajar, está localizado en raíz:

<PATH DEL PROYECTO>\Software Development Plan.docx

Este documento, denominado Software Development Plan (SDP), **debe** tener la siguiente estructura, la cual debe ser seguida:

2. Index	
Table of Contents	
1. Log	2
2. Index.....	3
3. Project Scope.....	4
4. Deliverables.....	4
5. Development methodology	4
6. Estimates.....	4
7. Planning.....	5
8. Solving Problem Strategy	6
9. Design.....	6
9.1. Standards.....	6
9.2. Naming conventions.....	6
10. Testing.....	7
10.1. Verification strategy (black box test).....	7
10.2. White box strategy.....	7
10.3. Cyclomatic Complexity Redundance index.....	7
11. Release	8
11.1. Software Development Folder	8
11.2. Integration Tests Strategy	8
11.3. Validation Testing / Functional Testing	8
11.4. Throughput and Flash and RAM measurement	9
12. Results	9
13. Lessons Learned	9

Figura 2 Índice

El documento **debe** estar escrito en inglés o bien en español, en caso de establecer un idioma como predefinido y por cuestiones del proyecto se necesita usar otro idioma, este segundo idioma deberá estar indicado de la siguiente manera “*English text*”.

El SDP **debe** ser el documento maestro del proyecto y **debe** seguirse y cumplirse a cabalidad.

Cada sección contiene:

- **SHALL.** Actividades que deben cumplirse de manera obligatoria (mandatorios)
- **COULD/MUST.** Estas actividades deben cumplirse, pero da opciones para su entrega (estas opciones deben ser definidas en el mismo documento).
- **SUGGESTED.** Estas actividades son sugeridas, opcionales, con base a la conveniencia del proyecto.

Ver el ejemplo a continuación:

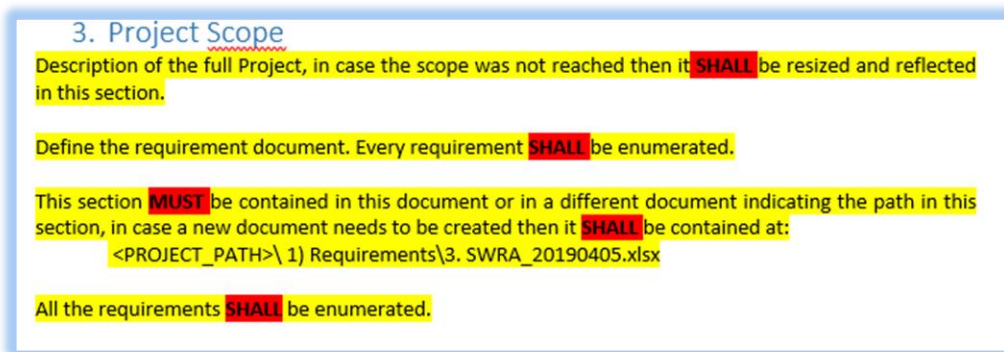


Figura 3

1.3. SW de control para Dispositivo CESEQ-001.

El software **debe** estar operando de acuerdo a requisitos definidos a continuación.

1.3.1. Descripción del proyecto (Requisitos generales).

Requisitos de operación del dispositivo CESEQ-C001 y CESEQ-P001

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.

1.3.2. Requerimientos funcionales específicos:

1.3.2.1. Entradas

Sensor de Efecto Hall

La salida del sensor debe ser un tren de pulsos de una señal cuadrada de frecuencia variable con voltaje máximo de 13.6 V descrita en la tabla 1.

Tabla 1 Valores sin carga del motor @13.6V.

Freezer Fan			
Frecuencia			
	Osciloscopio (Hz)	Interfaz (rpm)	Tacometro (rpm)
20%	28.6		427
30%	62.6	946	939
40%	94.6	1430	1427
50%	122	1830	1827
60%	147	2188	2190
70%	170.1	2575	2561
80%	187	2835	2847
90%	210	3131	3135
100%	236	3555	3549

*Columna 1 corresponde al porcentaje del ciclo de trabajo de una señal PWM cuya frecuencia es de 1Khz a un voltaje de 13.5V.

**Columna 2: Frecuencia de trabajo

***Columna 3: Revoluciones por minuto

****Columna 4: Revoluciones por minuto

Cabe señalar que, los pulsos que se generan no son simétricos. Para medir la velocidad **debe** considerarse el número de pulsos registrados, en el sensor de efecto hall, que ocurrieron en un periodo de 100 ms, promediarse con el de los siguientes 100 ms y, el resultado, será la velocidad mostrada en el display.

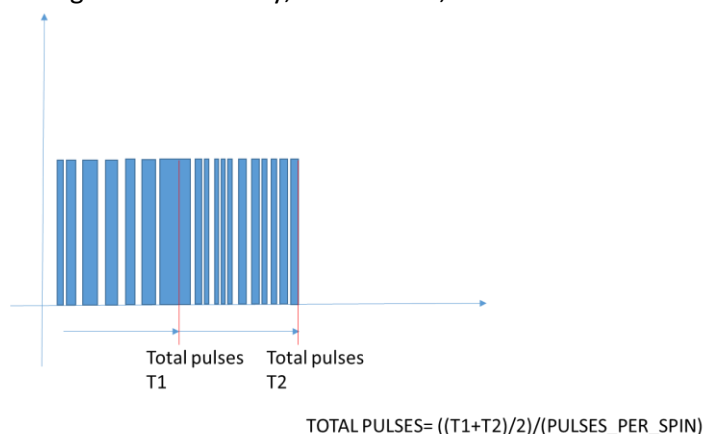


Figura 4 señal de salida de sensor de efecto Hall

Ajuste del valor de referencia o SetPoint mediante potenciómetro.

El ajuste del potenciómetro **debe** estar dado por:

Voltaje en el Potenciómetro	% PWM	Velocidad Promedio
0 V	x	0 rpm
1.5V	x	1500 rpm
3V	x	3000 rpm
3.3V	x	3000 rpm

El ajuste del potenciómetro **debe** estar conectado como se muestra en la figura 1.

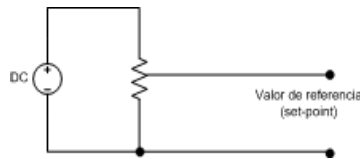


Figura 5

Para atenuar el ruido que pueda haber en la resistencia, se **deben** tomar tres muestras con un periodo de 100ms, estas **deben** de ser promediadas y el resultado será el valor del offset.

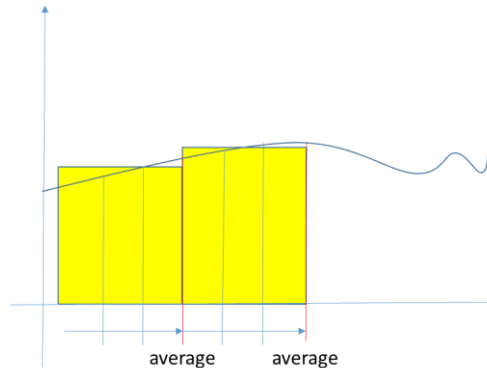


Figura 6 señal de entrada del potenciómetro

Ajuste del valor de referencia o SetPoint mediante puerto UART.

El ajuste del potenciómetro **debe** estar dado por:

Señal de control	%PWM	Velocidad Promedio
0x00	x	0 rpm
0x100	x	1500 rpm
0x200	x	3000 rpm
0x255	x	3000 rpm

El protocolo de comunicación en al UART entre la PC y la tarjeta es abierto, pero **debe** estar documentado (sección: diseño) y controlado (github).

El tiempo mínimo entre mensajes de actualización del offset **debe** ser de 200ms.

La velocidad de transmisión de al UART **debe** estar configurada a 115200 bits por segundo.

1.3.2.2. Salidas

Variación de velocidad de ventilador

Se **debe** considerar que:

- El circuito de potencia **debe** estar alimentado todo el tiempo a 12 volts
- El circuito **debe** proporcionar una salida estandarizada de 3.3V proveniente del sensor de efecto hall.

Se **debe** considerar el siguiente diagrama esquemático del circuito para el diseño de POTENCIA:

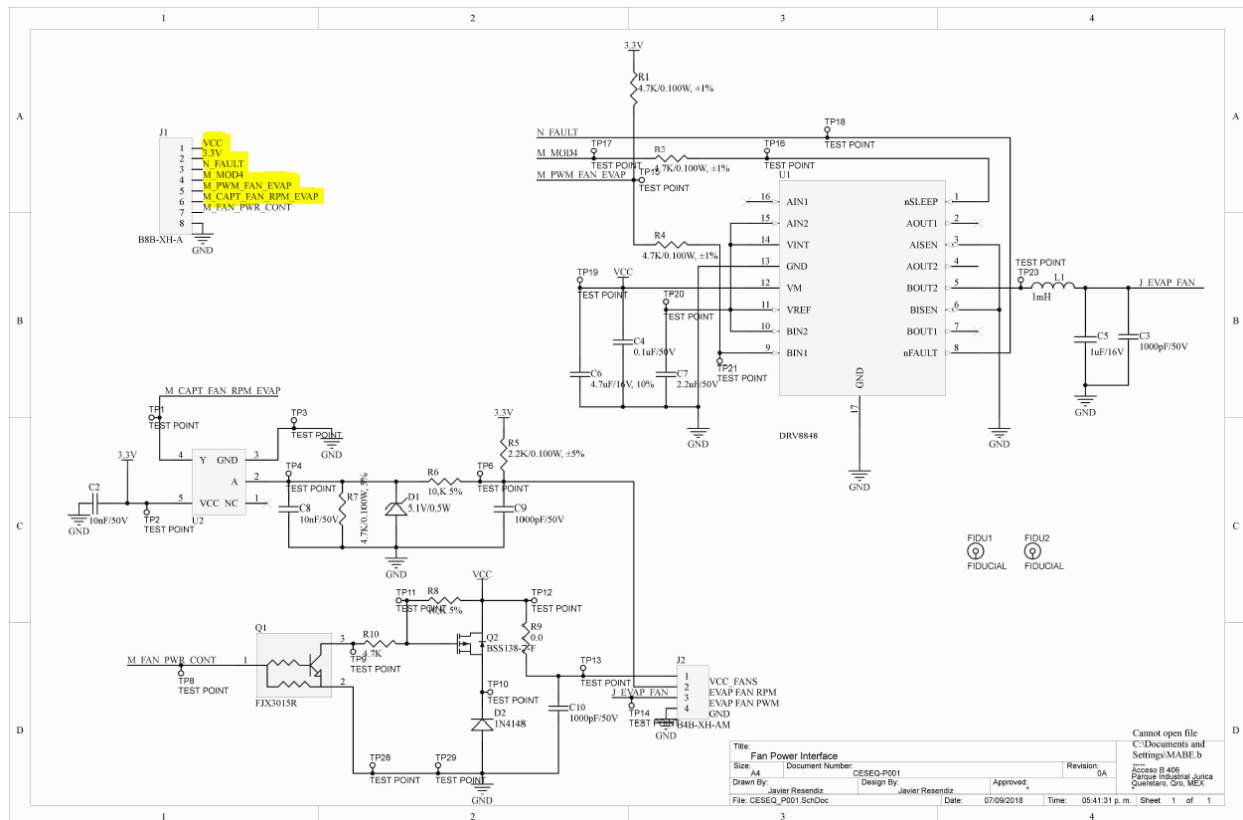


Figura 7 Diagrama esquemático

VCC	= 12 Volts
N_FAULT	= "Error condition to Uc"
M_CAPT_FAN_RPM_EVAP	= "Hall sensor condition"
M_FAN_PWR_CONT	= "Not used"

Voltaje de alimentación del motor

El voltaje de alimentación del motor del ventilador **debe** ser de 12 Vcd.

El tiempo de operación del motor **debe** determinarse basado en el tiempo de operación de la salida BOUT1, entre On/Off, a un periodo constante Δt , como se muestra en la figura 4:

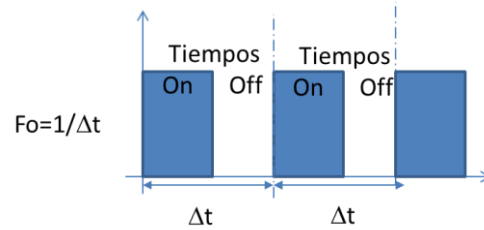


Figura 8

Así, al variar el tiempo de encendido On, se **debe** variar la velocidad del motor. El cambio en el tiempo On puede ir desde 0%, 25%... al 100%, dependiendo del ajuste del “set-point”. Ver figura 9:

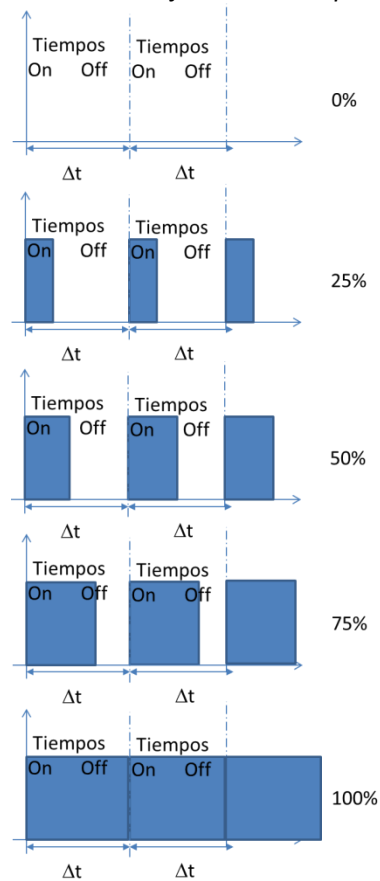


Figura 9

Este porcentaje de operación **debe** ser indicado en la pantalla del display de la siguiente forma:

Nombre del proyecto:	Control Vel. Motor CD
Duty cycle:	XXX %
Speed:	XXXX RPM
SW Versión:	X.X
HW Versión:	CESEQ-C001 / CESEQ-P001
Programadores:	Apellido1, Nombre1 Apellido2, Nombre2

1.3.2.3. Algoritmo de Control

El algoritmo de control **debe** ser en lazo cerrado y **puede** ser de tipo:

1. Proporcional
2. Proporcional-Integral
3. Proporcional-Integral-Derivativo
4. Redes neuronales
5. Lógica difusa
6. Variables de estado
7. Algoritmos genéticos
8. Otro (especificar).

Los tiempos definidos para ejercer la acción de control **deben** de realizarse cada 100 ms.

1.3.2.4. Sistema Operativo/Calendarizador.

El sistema **debe** contar con un sistema de control de tiempo para hacer determinísticos los procesos del código. Este está abierto a las siguientes opciones:

1. Máquina de estado. Deberán definir:
 - Estados
 - Eventos
2. Calendarizador (scheduler)
 - Tareas
 - Máquinas de estado (si aplica)
3. Sistema Operativo (Renesas Operative System)
 - RTOS (preemptive/non-preemptive)
 - Tareas
 - Eventos
 - Máquinas de estado (si aplica)
 - Semáforos.
 - Display handler
4. Otro
 - Especificar.

Configuraciones al Inicializar el sistema.

Durante el proceso de la inicialización del sistema operativo **debe** considerar la configuración de los siguientes registros:

1. Memoria *EEPROM*.
2. Temporizador por interrupción cada XXms (definido por el usuario).
3. Convertidores analógicos digitales
4. Modulares de ancho de pulso
5. Puertos digitales.
6. *Watchdog timer*
7. *Input capture*.
8. Pantalla *LCD*

Tareas del sistema.

El sistema **debe** considerar las siguientes tareas:

1. Puertos
 - Entradas-salidas digitales.
 - Convertidores analógicos digitales.
 - Modulación de ancho de pulso PWM.
 - Puerto SPI.
2. Control.
 - (ver sección de Control)
3. Diagnósticos.
 - Modo estático.
 - Detección de corto a batería.
 - Modo dinámico.
 - Detección de corto a tierra.
4. Actualización de la pantalla LCD.
5. Grabar valores en EEPROM.
 - Diagnóstico de cortos.
 - Controles digitales.

Análisis de *throughput* y Balanceo de tareas

Las tareas **deben** ser ejecutadas en un tiempo denominado ciclo de operación TIC's que corresponde a x mS. Todas las operaciones deberán desarrollarse en ese periodo.

Las tareas **deben** ser monitoreadas con una resolución de al menos 10x TIC's. Además, **deben** ser almacenadas en variables de tipo entero.

El criterio de rediseño **debe** ser: si el Throughput es mayor al 70% del uso del CPU. Será necesario dividir estados/tareas/subrutinas, para alcanzar el throughput propuesto.

Diagnósticos

El sistema **debe** detectar:

- A) Cortos en la etapa de Potencia.
- B) Corrupción de memoria
- D) Motor atascado.
- E) Error en la comunicación con el display.

Análisis de la complejidad ciclomática

Revisión de la Complejidad a través de la CCCC, la cual **debe** ser <19.

1.3.3. Requerimientos no funcionales

Los requerimientos no funcionales que deben estar comprendidos en el documento SDP son los siguientes:

1.3.3.1. Apariencia del display.

El display **debe** tener las siguientes características:

1. Contraste adecuado
2. Buena iluminación
3. Uso de la tipografía claro y legible.
4. Velocidad de refresco adecuado para evitar ver "*glitches*" o pantallas de transiciones.

1.3.3.2. Diagramas de flujo.

El proyecto **debe** incluir los diagramas de flujo del algoritmo de control y de las funciones empleadas.

2 INSUMOS:

El proyecto **debe** contar con:

1. Tarjeta YSSKS7G2E30 de RENESAS para control
2. Display: 320X240, 2.4"
3. Dispositivo CESEQ_P001 como dispositivo de potencia
4. Dispositivo CESEQ_C001 como planta o dispositivo a controlar

3 INFRAESTRUCTURA:

Cada equipo **debe** contar con:

1. Infraestructura
 - Mesa de trabajo,
 - Infraestructura de Red alámbrica o wifi.
 - PC con conectividad
2. Hardware
 - Osciloscopio,
 - Multímetro,
 - Fuente de alimentación,
 - Generador de funciones,
 - Tacómetro.
3. Software:
 - Renesas e2Studio IDE (Eclipse Based),
 - Compilador GCCARM.
 - GIT.
 - Microsoft Office.

4 PROCEDIMIENTOS:

1. Se **debe** utilizar una Metodología Agile, preferiblemente SCRUM.



Figura 6

2. Se **debe** controlar los requisitos, documentos y código generado:
 - Requisitos.
 - Versiones de software y liberaciones.
3. Se **debe** considerar en el diseño de código que sea:
 1. Modular.
 2. Reutilizable.
 3. Mantenible.
 4. Basado en pruebas.

5 REFERENCIAS BIBLIOGRÁFICAS:

NA.

6 CONTROL DE VERSIÓN DEL DOCUMENTO

No. Rev	Descripción del cambio	Autor	Fecha
11	<ol style="list-style-type: none">Se incluyen los diagramas de flujo como requisitos documentalesEn el punto 1.3.2.3 se especifica: Algoritmo de control	F. Aguilar-Pereyra	20190527
10	<ol style="list-style-type: none">Cambio en la señal PWMSistema Operativo /Calendarizador	L. Urióstegui	20190527
9	<ol style="list-style-type: none">Errores de redacción mínimos.	Adbeel A. Pérez	20190506
8	<ol style="list-style-type: none">Se detalló cada uno de los requisitos<ul style="list-style-type: none">Palabra clave debe y puede.Se eliminaron requisitos específicos del display.Se agregaron requisitos de las señales de entrada y salida.Se eliminó secciones que ya están contenidas en el SDP.	Adbeel Pérez	20190502
7	<ol style="list-style-type: none">Se modificó el modelo de la tarjeta de control.Se modificó la sección 1.1 en el punto de Aplicación de Estándares de codificación y convenciones.Se agregó el punto 1.2.3.2 Apariencia del displaySe modificó el punto 1.2.2.4 en la sección de diagnósticos	Eduardo Donjuán Luis Urióstegui José Rodríguez Adbeel Pérez Rafael Santa Ana Germán Vázquez	20171031

6	<ol style="list-style-type: none">5. Se modificaron la sección de documentación del proceso de desarrollo de sw.6. Captura del tren de pulsos del sensor de efecto hall.7. Variación de velocidad de ventilador.	Adbeel Pérez Eduardo Donjuan	20171024
5	<ol style="list-style-type: none">1. Se modificó formato general2. Se agregó control3. Sistema operativo<ul style="list-style-type: none">● Inicialización● Tareas	Luis Uriostegui Adbeel Perez	20171007
4	<ol style="list-style-type: none">1. Se creó tabla de contenidos2. Se agrega tabla de datos del ventilador3. Se agregó descripción del circuito potencia del ventilador.	Luis Urisotegui Adbeel Perez	20171004
3	<p>Se modificó</p> <ol style="list-style-type: none">1. Planeación<ul style="list-style-type: none">● Diseño arquitectura de SW● Prueba de Software2. Insumos3. Infraestructura	Adbeel A. Pérez Jose S. Pérez Luis H. Urióstegui	20170926
2	<p>Se modificaron los siguientes puntos:</p> <ol style="list-style-type: none">1. Documento SDP → se definieron más temas y se reorganizaron2. se restructiraron los procedimientos.	Adbeel A. Pérez	20170922
1	Revisión inicial	CESEQ, comité técnico	20170919