**UNIVERSIDAD TECNOLÓGICA DE QUERÉTARO**

**CESEQ**



**Diplomado en Software Embebido**

Proyecto Integrador

DOCUMENTO: Software Development Plan #CESEQ001

Scrum Master: Preciado, Francisco

Developer. Mendizábal, Rebeca

Velázquez, Aldo

Date: 20190919

# *Log*

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Descripción** | **Revisor** |
| 0.0.1 | 17/08/2019 | Planeación inicial | Mendizábal, Rebeca |
| 0.0.2 | 30/08/2019 | Actualización de las secciones: alcance del proyecto, entregables, metodología de desarrollo y entregables | N/A |
| 0.0.3 | 06/09/2019 | Actualización de las secciones: Estrategia de solución de problemas y Diseño | N/A |
| 1.0.0 | 13/09/2019 | Actualización de las secciones: Liberación, Pruebas y Diseño | Velázquez, Aldo |
|  |  |  |  |

# Índice

Tabla de Contenidos

[*Log*](#_heading=h.gjdgxs)***2***

[Índice](#_heading=h.30j0zll) **3**

[Alcance del proyecto](#_heading=h.1fob9te) **4**

[Identificación del proyecto](#_heading=h.pc9xkevs482s) 4

[Descripción general del proyecto](#_heading=h.vu0qtj735dt) 4

[Entregables con el cliente](#_heading=h.2et92p0) **4**

[Compromisos de entrega de los módulos de software](#_heading=h.kgam937dnz7q) 5

[Metodologías de desarrollo](#_heading=h.c8kv39o2haot) 5

[Metodología de desarrollo](#_heading=h.tyjcwt) **5**

[Estimación](#_heading=h.3dy6vkm) **7**

[Planeación](#_heading=h.1t3h5sf) **10**

[Estrategia de Solución de Problemas](#_heading=h.4d34og8) **10**

[Diseño](#_heading=h.2s8eyo1) **10**

[Estándares](#_heading=h.17dp8vu) 11

[*Naming conventions*](#_heading=h.3rdcrjn) *11*

[Pruebas](#_heading=h.26in1rg) **12**

[Pruebas de caja negra](#_heading=h.lnxbz9) 12

[Pruebas de caja blanca](#_heading=h.35nkun2) 12

[Complejidad Ciclomática](#_heading=h.1ksv4uv) 13

[Liberación](#_heading=h.44sinio) **13**

[Plan de entregas de liberaciones al cliente](#_heading=h.hhwwg297iz17) 13

[Estrategia de pruebas de integración](#_heading=h.2jxsxqh) 14

[Pruebas de validación](#_heading=h.z337ya) 15

[Procedimientos de control](#_heading=h.or25rzct1j06) 15

[Cálculo de throughput y memorias Flash y RAM](#_heading=h.3j2qqm3) 15

# Alcance del proyecto

Se controlará la velocidad de un motor de corriente directa mediante la aplicación de una señal cuadrada variable en su ancho de pulso, cuya frecuencia de trabajo deberá ser constante en un rango de 100 Hz a 1 KHz.

El valor de referencia de entrada será variable y podrá ser establecida por el usuario, a través de un potenciómetro. Se implementará un sistema de control Proporcional integral derivativo (PID) para establecer una retroalimentación al sistema. Finalmente, mediante el uso de un sensor de efecto hall, acoplado del rotor del motor, se medirá la velocidad del mismo, el cual proveerá una serie de pulsos cada que se complete una vuelta entera. De esta manera, a mayor velocidad del motor, mayor será el número de pulsos leídos y mientras menor sea la velocidad, menor será el número de pulsos.

La definición y análisis de los requisitos se podrá encontrar en el documento continuación: Proyecto\_Integrador/ESTRUCTURA DEL PROYECTO/1) Requisitos/SWRA\_20190817.xlsx

## Identificación del proyecto

## Descripción general del proyecto

## 

# Entregables con el cliente

A continuación se hará una breve descripción de los elementos a entregar al término del proyecto:

* **Código (.hex)**. Se entregará el código completo con la trazabilidad y documentación correspondiente, de manera que cualquier persona pueda interpretar y entender lo que se implemente en cada sección. De esta manera también será más sencillo el mantenimiento del mismo.
* **Documentación**. Se entregará una serie de documentos organizados en carpetas definiendo el proceso completo que se llevó a cabo para la elaboración del proyecto integrador desde su inicio hasta el final. De esta manera, quedará como evidencia y respaldo en caso de que se deseara replicar o hacer algún cambio al sistema. La documentación se dividirá en las siguientes categorías: requisitos, planeación, diseño, verificación y documentos de calidad, agregando como documento general el Plan de desarrollo de Software.

1. Requisitos. En esta carpeta se incluirá toda la información relacionada a la creación, diseño, análisis y aprobación de requisitos. Ésta se capturará en un archivo de Excel, donde se tendrá inicialmente una captura de los requisitos del cliente, posteriormente se hará un análisis validando que los requisitos son válidos y que cumplen con todas las características necesarias. Después se hará una revisión y aprobación de los mismos donde se establecerá qué competencia debe probar cada uno y se hará una vinculación tanto con las tareas de desarrollo de software como con las pruebas necesarias para verificar su correcto funcionamiento.
2. Planeación. Se incluirán todos los archivos que describen: los roles y principales responsabilidades de los miembros del equipo, siguiendo la metodología SCRUM; el calendario con las tareas asignadas a cada miembro del equipo, obtenidas de las estimaciones previas; el flujo de trabajo utilizado para el control de tareas; documento utilizado para el Análisis de Modos de Fallas y Efectos (FMEA); y las actividades contempladas durante toda la etapa del proyecto (documentos necesarios, actualizaciones, ejecución de prueba de caja blanca/negra, ejecución de prueba de integración y reuniones contempladas).
3. Diseño. Carpeta utilizada para archivar todos los diagramas de modelado estático y dinámico del proyecto, que corresponden a: diagramas de bloques, diagramas de flujo, diagramas “*call tree*”, diagramas de máquina de estados, diagramas de secuencia y otros dependiendo del paradigma de programación. Además, se incluirá el diagrama de control donde se definen: entradas, salidas, ruido y su retroalimentación. Por otro lado, se incluirá el documento relacionado al estándar utilizado para el proyecto y la herramienta para evaluarlo, así como el documento donde se define la convención de nombres.
4. Verificación. La documentación necesaria para la realización de la etapa de pruebas se incluirán en esta carpeta. Los archivos son relacionados a las pruebas que se realizarán de caja negra, caja blanca, integración, validación y CCR (“*Cyclomatic Complexity Redundance index*”). Además, se incluirá el proceso de la prueba de rendimiento y medición de memoria Flash y RAM.
5. Documentos de calidad. Las auditorias de calidad se incluirán en esta carpeta.

## Compromisos de entrega de los módulos de software

## Metodologías de desarrollo

# Metodología de desarrollo

La metodología que se siguió para el desarrollo del presente proyecto es la metodología ***SCRUM***. Para ello se designó a un *Product Owner* y a un *Scrum Master* cuyos nombres se presentan en la siguiente tabla.

|  |  |  |
| --- | --- | --- |
| **Rol** | **Nombre** | **Responsabilidades** |
| ***Product Owner (PO)*** | TBD | Maximizar el valor de los productos creados por el equipo de *scrum.* |
| ***Scrum Master (SM)*** | Preciado, Francisco | Asegurarse que el equipo sigue los valores y principios de la metodología ágil y sigue los procesos y prácticas acordadas por el equipo. Además, mantiene el código actualizado, haciendo las veces de integrador y remueve obstáculos del equipo para mantenerlo concentrado. |
| **Scrum development team member 1** | Mendizábal, Rebeca | Desarrolla apegándose a los requisitos y a los preceptos de la metodología ágil *scrum.* |
| **Scrum development team member 2** | Velázquez, Aldo | Desarrolla apegándose a los requisitos y a los preceptos de la metodología ágil *scrum.* |
| **Scrum development team member 3** | Preciado, Francisco | Desarrolla apegándose a los requisitos y a los preceptos de la metodología ágil *scrum.* |

Tabla X.X Roles del equipo

Debido a la naturaleza de la metodología elegida, se utilizó Zenhub como herramienta para la administración del proyecto debido a su fácil integración con Git, el *software* de control de versiones requerido para este proyecto. A continuación se presenta una liga al *scrum board del equipo:*

***Zenhub***

La longitud de *sprint* que se juzgó pertinente por el tiempo disponible para la implementación del proyecto fue de una semana natural. Empezando los lunes y terminando los domingos. El calendario con los sprints se presenta en la figura X.X.

Por cuestiones de tiempo no se llevarán a cabo las *Daily Scrum Meetings****.*** El estatus que normalmente se actualizaría en dichas juntas se controló con un documento interno del equipo en el que se podía consultar el estatus de cada miembro del equipo. Dicho documento era actualizado por los integrantes del equipo cada que su estatus cambiaba.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lunes** | **Martes** | **Miércoles** | **Jueves** | **Viernes** | **Sábado** | **Domingo** |
|  |  |  |  |  | **Agosto 17** | **18** |
| **19** | **20** | **21** | **22** | **23** | **24**  *Big Room Planning* | **25** |
| **26** *Sprint* 1 | **27** | **28** | **29** *Review Meeting* | **30** | **31** *Planning & Retrospective Meeting* | **Septiembre 1** |
| **2** *Sprint* 2 | **3** | **4** | **5** *Review Meeting* | **6** | **7** *Planning & Retrospective Meeting* | **8** |
| **9** *Sprint* 3 | **10** | **11** | **12** *Review Meeting* | **13** | **14** *Planning & Retrospective Meeting* | **15** |
| ***16*** *Sprint* 4 | **17** | **18** | **19** *Review Meeting* | **20** | **21** *Planning & Retrospective Meeting* | **22** |
| ***23*** *Sprint* 5 | **24** | **25** | **26** *Review Meeting* | **27** | **28** *Planning & Retrospective Meeting* | **29** |
| ***30*** *Sprint* 6 | **Octubre 1** | **2** | **3** *Review Meeting* | **4** | **5** *Planning & Retrospective Meeting* | **6** |
| ***7*** *Sprint* 7 | **8** | **9** | **10** *Review Meeting* | **11** | **12** *Planning & Retrospective Meeting* | **13** |
| ***14*** *Sprint* 8 | **15** | **16** | **17** *Review Meeting* | **18** | **19** *Planning & Retrospective Meeting* | **20** |
| ***21*** | **22** | **23** | **24** | **25** | **26** |  |

Tabla X.X Calendario de juntas.

# Estimación

## Infraestructura y herramientas

Para esta estimación se consideró que se trabajará con los insumos del laboratorio únicamente todos los viernes del 23 de agosto de 2019 al 25 de octubre de 2019. Por lo tanto, su disponibilidad medida en horas (h) es la siguiente:

|  |  |
| --- | --- |
| *Microcontrolador, planta, osciloscopio, multímetro, generador de funciones, y fuente de alimentación:* | 40h |

Tabla X.X Disponibilidad del *hardware* de laboratorio.

Esto, asumiendo que las herramientas estén completamente funcionales durante ese tiempo. Esto se considerará en el AMEF en la sección 7 del presente documento.

## Recursos humanos

La estimación de los recursos humanos se realizó bajo el acuerdo que cada miembro del equipo trabajará 8 horas por semana durante el mismo periodo antes descrito en esta misma sección. Dentro del acuerdo se consideró que el *scrum master* deberá invertir 2 horas más por semana para cubrir con sus responsabilidades.

|  |  |
| --- | --- |
| Tiempo de desarrollo: | 220 h. |
| Tiempo actividades propias del *scrum master* | 40h. |

Tabla X.X Disponibilidad de los recursos humanos

Esto asumiendo que el equipo no tenga nada que le impida cumplir con este acuerdo. Esto será incluido en el AMEF como parte de los entregables del proyecto en la siguiente dirección: Proyecto\_Integrador/ESTRUCTURA DEL PROYECTO/2) Planeación/AMEF.xlsx

**Listado de riesgos**

El cálculo del número prioritario de riesgo se realizará en la sección 8 del presente documento.

Las principales actividades de riesgo:

* Hardware dañado o faltante.
* Falta de disponibilidad temporal o permanente de los integrantes del equipo debido a asuntos personales o laborales.
* Vacaciones de la UTEQ.
* Nuevo hardware y/o microcontrolador.
* Falta de conocimiento del lenguaje de programación, SW IDE o sistema operativo.
* Error en la conexión del hardware.
* Cambios a la planeación inicial.
* Mala comunicación entre los integrantes.
* Falta de conocimiento de la metodología SCRUM.
* Errores de comunicación del microcontrolador.
* Mala estimación de recursos a las tareas.
* Error al acceder a la documentación o repositorio del proyecto.

## Recursos críticos

## Desglose de actividades

**Lectura y análisis inicial de la descripción del proyecto**

* Planeación para la documentación inicial

**Documentación inicial**

* Estimación de recursos
* Desglose de actividades
* Análisis de dependencias
* Desarrollo de Diagrama de Gantt
  + Establecer milestones
* Detección de riesgos
* Desarrollo AMEF
* Análisis de requisitos de sistema

**Configuración de herramientas**

* Instalación del ambiente de desarrollo (e2studio)
* Creación del repositorio de Git
* Ambiente de edición de documentos

**Inicialización de periféricos**

* Lectura de entrada analógica (ADC)
* Configuración del *PWM*
* Input Capture para el sensor de efecto Hall
* Configuración de protocolo SPI (para pantalla LCD)
* Configuración del protocolo de comunicación UART

**Desarrollo del algoritmo de control**

* Establecer tipo de controlador a utilizar
* Base de tiempo por medio de interrupción o temporizador
* Creación del modelo de control
* Simulaciones del sistema
* Cálculo de la velocidad
* Implementación del algoritmo de control en C

**Interfaz de usuario**

* Formato de la información para el usuario
* Configuración de las características del display
* Despliegue de los datos del sistema

**Verificación**:

* Evaluación del estándar de programación
* Pruebas de caja blanca
  + Documentación de pruebas
* Pruebas de caja negra
  + Documentación de pruebas
* Cálculo de la complejidad ciclomática
* Pruebas de integración
* Medición de *throughput*, *RAM* y *Flash*

# Planeación

## Roles y responsabilidades

Las definición de terminado de cada una de las tareas se encuentra en ZenHub, dado que es el lugar donde los desarrolladores tiene mejor acceso

Los roles y responsabilidades del los miembros del equipo se pueden encontrar en la sección 6 del presente documento, en la table X.X.

## Trazabilidad de entregables

# Estrategia de Solución de Problemas

Partiendo de la sección 11 del documento, donde se establecen los recursos y estimaciones, se desarrolló desde el inicio del proyecto y durante toda su realización, un documento de Análisis de Modo y Efecto de Falla (AMEF) ubicado en la siguiente ruta: Proyecto\_Integrador/ESTRUCTURA DEL PROYECTO/2) Planeación/AMEF.xls. En él se establecen todos los posibles modos de falla y riesgos que se pudieran dar a lo largo de la ejecución del proyecto.

# Diseño

## Diagrama de bloques de software

## Diagrama de control

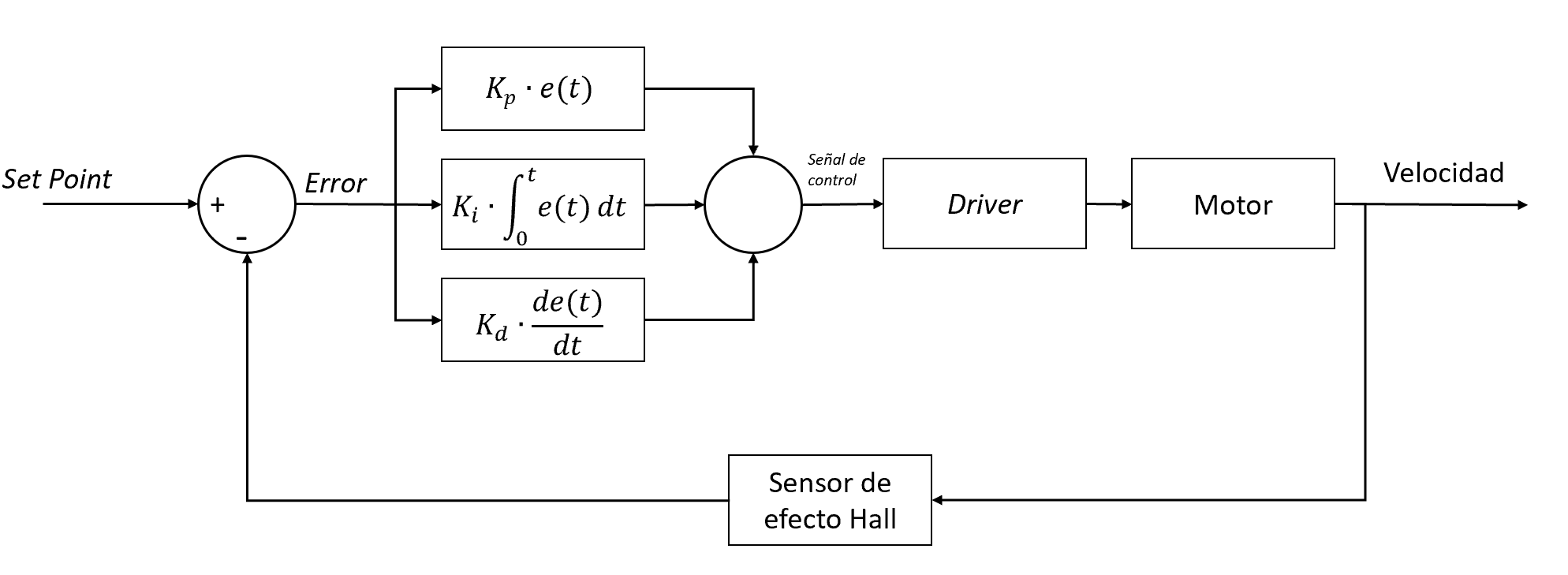
El tipo de controlador elegido fue el Proporcional Integral Derivativo por su facilidad de ser implementado en C en un microcontrolador. En la siguiente figura se puede observar el diagrama de bloques del sistema controlado en lazo cerrado. El ruido del sistema se despreció para simplificar el diseño y la implementación del controlador por lo que no se encuentra en el diagrama de bloques. 

Figura X.X Diagrama de control

## Estándares

El estándar utilizado para el desarrollo de código C en el presente proyecto fue C99.

La herramienta utilizada para evaluar que el código desarrollado sigue el estándar elegido fue Cpp Check.

## *Naming conventions*

|  |  |
| --- | --- |
| **Tipo de elemento de código** | ***Naming Convention*** |
| Variable local |  |
| Variable global | Las variables globales **deben** ir todas en mayúsculas. |
| Función local |  |
| Función global |  |
| *Macros* | Las *macros* **deben** ir todas en mayúsculas. |
| Enumeraciones |  |
| Estructuras |  |
| Archivos | Los nombres de las carpetas **deben** de ir todos en minúsculas y en lugar de espacios tener guiones bajos. |
| Carpetas | Los nombres de las carpetas **deben** de empezar con mayúscula y en lugar de espacios tener guiones bajos. |

Tabla X.XX *Naming convention* de los diferentes elementos del código

# Pruebas

Las pruebas de software contribuyen a que la implementación o producto final se comporte de manera correcta de acuerdo a los requisitos del cliente. Es importante ejecutar este tipo de actividades durante el desarrollo del producto, ya que ayudan a la calidad del mismo y al equipo de desarrollo a reducir costos de errores en el mercado final.

Existen varios tipos de pruebas, dentro de los cuales se seleccionaron las más importantes para el desarrollo del proyecto. Mostradas a continuación:

## Pruebas de caja negra

Este tipo de pruebas siguen la metodología de que la estructura o diseño internos no se prueba o no se conoce por la persona que la ejecuta. Esto es para evitar que la opinión del *tester* o el resultado final de la prueba se vean afectados por el juicio del desarrollador.

Los casos de prueba de este proyecto se hicieron basados en los requisitos finales. Se corrieron cada vez que se terminó un proceso de liberación de software (cada dos sprints) y se documentó lo correspondiente a las mismas. La evidencia se puede encontrar en la siguiente ruta: Proyecto\_Integrador/ESTRUCTURA DEL PROYECTO/4) Verificación/PruebasCajaNegra.xls.

## Pruebas de caja blanca

## Complejidad Ciclomática

# Liberación

Se estableció que las liberaciones a pruebas se realizarán cada dos semanas, al igual que al cliente, sin embargo, éstas tendrán un desfase de 1 semana para poder ejecutar las pruebas y corregir los errores encontrados en éstas, de manera que se haga una liberación con una mayor calidad al cliente. También se estableció un acuerdo para nombrar a las versiones, la cual establece a continuación. Un diagrama más detallado sobre las fechas de liberación se encuentra en la siguiente ruta: Proyecto\_Integrador/ESTRUCTURA DEL PROYECTO/2) Planeación/Liberaciones.pptx

15/09/2019 - v0.1.0 - HW001

29/09/2019 - v0.2.0 - HW001

13/10/2019 - v0.3.0 - HW001

25/10/2019 - v1.0.0 - HW001

## Plan de entregas de liberaciones al cliente

En el calendario a continuación, se establecen las fechas de liberaciones tanto a pruebas como al cliente,

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lunes** | **Martes** | **Miércoles** | **Jueves** | **Viernes** | **Sábado** | **Domingo** |
|  |  |  |  |  | **Agosto 17** | **18** |
| **19** | **20** | **21** | **22** | **23** | **24** | **25** |
| **26** *Sprint* 1 | **27** | **28** | **29** | **30** | **31** | **Septiembre 1** |
| **2** *Sprint* 2 | **3** | **4** | **5** | **6** | **7** | **8**  Liberación de la versión 0.1.0 (pruebas) |
| **9** *Sprint* 3 | **10** | **11** | **12** | **13** | **14** | **15**  Liberación de la versión 0.1.0 (cliente) |
| ***16*** *Sprint* 4 | **17** | **18** | **19** | **20** | **21** | **22**  Liberación de la versión 0.2.0 (pruebas) |
| ***23*** *Sprint* 5 | **24** | **25** | **26** | **27** | **28** | **29**  Liberación de la versión 0.2.0 (cliente) |
| ***30*** *Sprint* 6 | **Octubre 1** | **2** | **3** | **4** | **5** | **6**  Liberación de la versión 0.3.0 (pruebas) |
| ***7*** *Sprint* 7 | **8** | **9** | **10** | **11** | **12** | **13**  Liberación de la versión 0.3.0 (cliente) |
| ***14*** *Sprint* 8 | **15** | **16** | **17** | **18** | **19** | **20**  Liberación de la versión 1.0.0 (pruebas) |
| ***21*** | **22** | **23** | **24** | **25**  Liberación de la versión 1.0.0 (cliente) | **26** |  |

Tabla X.X Calendario de liberaciones de *software*.

## Estrategia de pruebas de integración

Las pruebas de integración se pueden definir como el paso inmediato anterior a que se haga cualquier liberación, ya sea a pruebas o al cliente. Éstas sirven para asegurar que la funcionalidad básica del producto no ha sido afectada por los cambios y nuevas implementaciones hechas al código.

La documentación correspondiente se puede encontrar en la siguiente ruta: Proyecto\_Integrador/ESTRUCTURA DEL PROYECTO/4) Verificación/PruebasIntegracion.docx

## Pruebas de validación

Las pruebas de validación son aquellas que se ejecutan considerando el escenario o ambiente final del producto. Generalmente son realizadas por un ingeniero de sistemas basándose tanto en los requisitos del cliente como en las acciones más frecuentes de los usuarios finales, por eso son llamadas pruebas de aceptación de usuario (UAT por sus siglas en inglés).

La documentación correspondiente se puede encontrar en la siguiente ruta: Proyecto\_Integrador/ESTRUCTURA DEL PROYECTO/4) Verificación/PruebasValidacion.docx

## Procedimientos de control

## Cálculo de *throughput* ymemorias Flash y RAM