

**Universidad Autónoma de Yucatán**

**Facultad de Matemáticas**

**Licenciatura en Ingeniería de Software (MEFI)**

**Aseguramiento de la Calidad de Software**

**MODELO DE PROCESOS DE SOFTWARE**

**Integrantes:**

**Chavez Bagundo Emanuel Rafael**

**Mendoza Vértiz Víctor Eduardo**

**Ortega Aguilar Andrea Natalí**

**Pérez Vázquez Jamart Uriel**

**Valencia Guzmán Emiliano**

**ÍNDICE**

[**Introducción 2**](#_ioscqbsh5s2d)

[**Propósito 2**](#_kvb0lpp6y44h)

[**Objetivos 2**](#_c8zj43d63kk1)

[**Indicadores 2**](#_wschyyykb16e)

[**1. Roles 3**](#_45p7xzefu5ro)

[1.1 Definición de roles. 3](#_gz0xet61tvga)

[1.2 Descripción de roles. 3](#_iikuenue2ong)

[**2. Definición de Procesos. 6**](#_tjjwnfos8j6p)

[2.1 Inicio 6](#_lgcqrc3oswna)

[2.2 Análisis y diseño 9](#_aniqa9jpht7b)

[2.3 Construcción 12](#_aajq39n6rgsk)

[2.4 Implementación y pruebas 14](#_fgzowvojrc7s)

[2.5 Cierre 17](#_9ly093dk2vx6)

[**3.0 Costos de calidad de software 19**](#_4ywd9svgozuc)

[3.1. Costos de control 19](#_x860bz20dtkd)

[3.2. Costos de fallo de control 19](#_t87hp01stfvl)

[**4.0 Unidad de SQA 20**](#_pieri81ab44h)

[4.1 Actores 20](#_jv19v055ezbm)

[4.2 SQA en las fases de desarrollo y mantenimiento. 20](#_ybtdoi6wbofl)

[4.2.1 Inicio 20](#_fbvar3pojep7)

[4.2.2 Análisis y diseño 20](#_qv37j8ag7a85)

[4.2.3 Construcción 21](#_6litwfi07ggj)

[4.2.4 Implementación y pruebas 21](#_eggil32ol668)

[4.2.5 Cierre 21](#_b10smde7ayeo)

[**5.0 Anexo 22**](#_56paqztruv7)

[5.1 Gestión de configuración. 22](#_vi459rq3bem)

[5.2 Inicio. 23](#_6q0ccu1t2xi6)

[5.3 Análisis y diseño. 23](#_rhn5eubwl0wj)

[5.3.1 Descripción de componentes 23](#_yhhsdcguxl08)

[5.3.2 Modelo de diseño 24](#_ofjtvq4kluuv)

[5.3.3 Documentos de control. 25](#_yb2ozdb15tzr)

[5.3.3.1 Modelo de diseño 25](#_j78piyo0rsz3)

[5.3.3.2 Plan de pruebas de unidad e integración 25](#_mljg8y5tlfpe)

[5.3.3.3 Registro de rastreo 26](#_lh02oyixfirp)

[5.3.3.4 Reporte de actividades 26](#_ufoifb3gbnd)

[**6.0 Control de progreso del proyecto 27**](#_jv9jw6i3r835)

[6.1 27](#_kmavexbgidfq)

[4.4 Construcción. 27](#_dzec9xkp0bm)

[4.5 Implementación y pruebas. 28](#_ce65w8mswd5s)

[4.6 Cierre. 28](#_e8lnxruh389r)

[4.7 Corrección de defectos encontrados. 28](#_jpim017xi3ln)

# 

# Introducción

El modelo de procesos de software presentado a continuación proporciona un conjunto de elementos que ayudarán al equipo de trabajo involucrado en un proyecto a llevar un orden de actividades y fases necesarias para asegurar la calidad tanto de los procesos como de los productos para poder lograr la realización y entrega de productos de software adecuados.

# 

# Propósito

El propósito de este documento es definir los procesos de desarrollo y mantenimiento mediante la orientación al equipo de trabajo para aplicar buenas y sistemáticas prácticas en cada fase de un proyecto, esto para asegurar que se cumplan los objetivos establecidos al inicio de este y el cliente tenga un producto que satisfaga todos los requerimientos de la mejor manera posible, todo esto enfocado a los ciclos de desarrollo del software.

Durante el proceso de desarrollo y mantenimiento de software se requieren de fases concretas y sistemáticas bien definidas que den como resultado un producto que cumpla con las expectativas de la organización, para alinear estas actividades y establecer un rendimiento de proceso estándar se definen las siguientes.

# Objetivos

* Mejora incremental de los procesos que corresponden a cada fase del desarrollo.
* Mejora continua del sistema de gestión de desarrollo y calidad del software.
* Responsabilidades, autoridades y objetivos claros en cada proyecto.

# Indicadores

* Cada fase se realiza conforme a los establecido y se clasifica al menos con el nivel de madurez 2: Gestionado del modelo de procesos CMMI v1.3
* Los productos de entrada y salida son coherentes entre sí al finalizar cada fase.
* Se hicieron los procesos de verificación y validación correspondientes.

# 1. Roles

## **1.1 Definición de roles.**

| **ROL** | **ABREVIATURA** |
| --- | --- |
| **Administrador de Proyecto Específico** | **ADMIN** |
| **Responsable de Desarrollo y Mantenimiento de Software** | **RDM** |
| **Analista** | **AN** |
| **Diseñador** | **DI** |
| **Programador** | **PR** |
| **Responsable de pruebas** | **RPU** |
| **Revisor** | **RE** |
| **Responsable de Manuales** | **RM** |
| **Equipo de Trabajo** | **ET** |
| **Cliente** | **CL** |
| **Usuario** | **US** |

## **1.2 Descripción de roles.**

**Administrador de proyecto.** Es el responsable directo de los procesos y de que los productos resultantes de estos cumplan con los requisitos. Es aquel que lleva toda la relación formal con el cliente, administra y controla los recursos asignados a un proyecto.

Entre sus objetivos se encuentran terminar el producto en tiempo y con el presupuesto antes estimado, no descuidando la calidad y asegurarse de cumplir con éxito cada una de las fases del proyecto. De igual manera debe asegurarse de la capacitación del grupo en las metodologías y estándares a utilizar

**Responsable de desarrollo y mantenimiento de software.** Responsable de observar cada actividad para detectar y resolver inconvenientes, realizar estimaciones de horas-hombre, este debe tener una comunicación fluida con cada miembro del equipo, definir un plan y protocolo para desarrollo de reuniones. Coordinar a los integrantes para cumplir los objetivos particulares (un plan de trabajo específico y flujo de actividades).

**Analista.** Participará en reuniones con el administrador, RPM y diseñadores para estudiar el problema y descomponerlo en subproblemas de menor complejidad para realizar la especificación de requisitos, utilizando el estándar de la ESA para el análisis, se dividirá en dos: requisitos de usuario y requisitos de software.

Deberá mantener reuniones con el cliente y ayudarle a identificar sus necesidades además de verificar que los requisitos sean los correctos.

**Diseñador.** Con el documento de especificación de requisitos deberá definir una estructura del sistema que incluya módulos y resultados esperados y generar los diagramas de la arquitectura de manera clara para los demás miembros del equipo, a su vez, debe considerar una interfaz que reduzca complejidad, la cual esté revisada y aprobada por el cliente; además, interactúa de manera continua con ellos para asegurarse de mantener la organización y abstracción previamente establecida. Debe generar el diseño arquitectónico del sistema y mantenerlo actualizado durante el proyecto es parte de sus responsabilidades también.

También debe tener relación con los responsables de prueba para apoyarlos en la verificación y mediciones que se realizarán para la validación del sistema.

**Programador.** Estos son aquellos que, basados en la documentación que reciban, deben transformarlo en código ejecutable, utilizando uno o más lenguajes de programación y herramientas de software de apoyo para la programación.

Deben reducir el tiempo de codificación, la complejidad y errores que podrían entorpecer las siguientes fases. Para esto es vital la evaluación personal y en equipo de los integrantes por lo que estos deberán ser evaluados con el modelo PSP y TSP para detectar errores y corregirlos en el presente y futuro. Deberán realizar la elaboración de documentación como son diagramas de su codificación y versiones del proyecto, esto de manera sencilla pero efectiva a la hora de presentarla al administrador de proyecto, diseñador, tester, etc.

La elección del lenguaje, estilo y ambiente de programación es su responsabilidad y esta es aprobada por el (los) analista(s) y diseñador(es).

**Responsable de pruebas.** Debe construir y aplicar los planes de prueba unitarios, de módulo, de sistema, y aceptación parcial, manteniéndolos actualizados durante el proyecto, debe coordinar las inspecciones. De igual manera, este participa en la especificación del diseño para prevenir errores en etapas tempranas del desarrollo.

Debe asegurarse de construir buenos casos de test que tengan altas probabilidades de encontrar defectos aún no descubiertos, para esto debe tener un basto conocimiento de la metodología utilizada en el proyecto

**Revisor**. Proveer de información acerca de los resultados para correcciones y posteriormente realizar nuevas revisiones. Debe asegurarse de visitar todas las decisiones lógicas, loops y estructuras de datos para asegurar su validez. Los test de caja negra son parte de sus tareas para verificar el rendimiento del sistema y nivel de satisfacción de los requerimientos de este mismo.

Para validar debe demostrar que las funciones del sistema parecen estar funcionando de acuerdo a sus especificaciones y proveer una buena indicación de la confiabilidad del software y algunas indicaciones de la calidad del software.

**Responsable de Manuales.** Debe procurar el almacenamiento y recuperación de manera sencilla la información de los procesos y productos más recientes, mantener la apariencia y estructura de los manuales no permitiendo el almacenamiento de estos con formatos diferentes. Debe definir una estructura para cada uno de los manuales como pueden ser, capítulos, secciones, subsecciones, índices, etc.

Debe asegurar la calidad de estos y reflejar los cambios en estos de ser necesario. para cada manual debe analizar las características deseadas y validarlas con aquellos que usen estos documentos.

**Equipo de trabajo.** Son cada una de las personas que se ve involucrada de manera directa en la elaboración o mantenimiento del proyecto, todos deben tener responsabilidades y objetivos asignados.

**Cliente.** Es aquel que paga por la elaboración del proyecto y tiene alta importancia ya que es el que aprueba requisitos y funcionalidades que deberá tener el producto final.

**Usuario.** Es a quién va dirigido el producto final y se debe tener en cuenta el perfil (general) de este para el desarrollo y su opinión (feedback) para posterior mantenimiento.

# 2. Definición de Procesos.

## **2.1 Inicio**

**Propósito**

El propósito es conocer y entender el Plan de Desarrollo para que, a partir de ello, se pueda elaborar un documento de Especificación de Requerimientos adecuado y completo, proponiendo un producto de Software que satisfaga las necesidades del cliente.

**Descripción**

A lo largo de esta fase se realizarán reuniones con el administrador del proyecto, el analista y diseñador, de manera que se analice a detalle el Plan de Desarrollo y se pueda establecer el producto de software que se espera conseguir, por medio de un documento de Especificación de Requerimientos. Para ello, también se necesitará establecer un prototipo de la interfaz y una reunión con el cliente que ayudarán a ajustar detalles y asegurar que el producto cumple con las funcionalidades solicitadas. Adicionalmente, se elaborará el Plan de Pruebas del Sistema.

**Objetivos**

1. Conocer y comprender el plan de desarrollo.
2. Obtener un documento de Especificación de Requerimientos acorde al Plan de Desarrollo.

**Indicadores**

1. Se obtiene la especificación de requerimientos y es aprobado por el cliente.
2. Se tiene la documentación del prototipo de interfaz del proyecto.

**Metas**

* El cliente verifica y aprueba los requisitos especificados y se siente satisfecho con las soluciones propuestas por el equipo de trabajo.

**Responsable**

* Administrador de proyectos y Analista.

**Entradas**

* Plan de Desarrollo

**Salidas**

* Especificación de requerimientos
* Prototipo de la interfaz
* Plan de Pruebas del Sistema
* Manual de Usuario
* Reporte de Actividades

**Actividades**

| **No.** | **Actividad** | **Roles involucrados** |
| --- | --- | --- |
| A1. | Revisar con los miembros del equipo de trabajo el Plan de Desarrollo actual para lograr un entendimiento común y obtener su compromiso con el proyecto. | ET |
| A2. | Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas. | RDM |
| A3. | Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual. | RDM  AN |
| A4. | Documentar o modificar la *Especificación de Requerimientos.*  • Identificar y consultar fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) para obtener nuevos requerimientos.  • Analizar los requerimientos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio del cliente o del proyecto.  • Elaborar o modificar el prototipo de la interfaz con el usuario.  • Generar o actualizar la Especificación de Requerimientos. | AN  CL  US  DI |
| A5. | Verificar la Especificación de Requerimientos (Ver1). | RE |
| A6. | Corregir los defectos encontrados en la Especificación de Requerimientos con base en el Reporte de Verificación y obtener la aprobación de las correcciones. | AN  DI |
| A7. | Validar la Especificación de Requerimientos (Val1). | CL  US  RPU |
| A8. | Corregir los defectos encontrados en la Especificación de Requerimientos con base en el Reporte de Validación y obtener la aprobación de las correcciones. | AN  DI |
| A9. | Elaborar el Plan de Pruebas de Sistema. | RPU  AN |
| A10. | Verificar el Plan de Pruebas de Sistema (Ver2). | RE |
| A11. | Corregir los defectos encontrados en el Plan de Pruebas de Sistema con base en el Reporte de Verificación y obtener la aprobación de las correcciones. | RPU |
| A12. | Documentar la versión preliminar del Manual de Usuario o modificar el manual existente. | RM |
| A13. | Verificar el Manual de Usuario (Ver3) | RE |
| A14. | Corregir los defectos encontrados en el Manual de Usuario con base en el Reporte de Verificación y obtener la aprobación de las correcciones. | RM |
| A15. | Incorporar Especificación de Requerimientos, Plan de Pruebas de Sistema y Manual de Usuario como líneas base a la Configuración de Software. | RDM |
| A16. | Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas. | RDM |

**Verificaciones y validaciones**

| **Verificación o validación** | **Actividad** | **Producto** | **Rol** | **Descripción** |
| --- | --- | --- | --- | --- |
| **ver1** | A5 | Especificación de Requerimientos | RE | Verificar la claridad de redacción de la *Especificación de Requerimientos* y su consistencia con el *Plan de Desarrollo* y con el estándar de documentación requerido. Adicionalmente revisar que los requerimientos sean completos y no ambiguos o contradictorios. Documentar los defectos encontrados en un *Reporte de de Verificación.* |
| **val1** | A7 | Especificación de Requerimientos | CL  US  RPU | Validar que la *Especificación de Requerimientos* cumple con las necesidades y expectativas acordadas incluyendo la realización de la prueba de usabilidad de la interfaz de usuario. Documentar los defectos encontrados en un *Reporte de Validación.* |
| **ver2** | A10 | Plan de Pruebas del Sistema | RE | Verificar la consistencia del *Plan de Pruebas del Sistema* con la *Especificación de Requerimientos* y con el estándar de documentación requerido. Los defectos encontrados se documentan en un *Reporte de Verificación* |
| **ver3** | A11 | Manual de Usuario | RE | Verificar la consistencia del *Manual de Usuario* con la *Especificación de Requerimientos* y con el estándar de documentación requerido en el *Proceso Específico*. Los defectos encontrados se documentan en un *Reporte de Verificación.* |

## 

## **2.2 Análisis y diseño**

**Propósito**

El propósito es trazar metas de desarrollo de software y establecer los componentes esperados para el producto de Software, obteniendo el Diseño del sistema a elaborar.

**Descripción**

Durante esta fase, se recibirá el diseño preliminar de la interfaz y el documento de Especificación de Requerimientos, para poder descomponer el sistema de Software y definir los componentes que tendrá. Se tendrán reuniones para poder discutir cuál es la mejor manera de desarrollarlos, las prácticas y técnicas de construcción que se utilizarán. Así, se obtendrá un Modelo del Diseño del Sistema, en donde igual se describirán las prácticas seleccionadas. Por último, se establecerán las métricas que medirán el funcionamiento del Software y el registro de rastreo, todo esto se documentará.

**Objetivos**

1. Elaborar el Modelo del Diseño del Sistema de acuerdo al documento de Especificación de Requerimientos.
2. Especificación de las métricas de software y describirlas dentro del documento “Métricas del Proyecto”.

**Indicadores**

1. Se realizan las verificaciones y validaciones correspondientes del diseño de los componentes del sistema.
2. Se documenta correctamente las métricas que se utilizarán para las fases finales.

**Metas**

* Se elabora y valida el diseño del sistema de software.

**Responsable**

* Analista y Diseñador.

**Entradas**

* Plan de Desarrollo
* Especificación de Requerimientos

**Salidas**

* Análisis y Diseño
* Plan de Pruebas de Integración
* Registro de rastreo
* Reporte de Actividades

**Actividades.**

| **No.** | **Actividad** | **Roles involucrados** |
| --- | --- | --- |
| A1. | Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual. Definición de CI’s (Anexo 3.1). | RDM  AN  DI |
| A2. | Documentar o modificar el Análisis y Diseño:  • Analizar la Especificación de Requerimientos para generar la descripción de la estructura interna del sistema y su descomposición en subsistemas, y éstos a su vez en componentes, definiendo las interfaces entre ellos. (IEEE 1016-2009).  • Describir el detalle de la apariencia y el comportamiento de la interfaz con base en la Especificación de Requerimientos de forma que se puedan prever los recursos para su implementación.  • Describir el detalle de los componentes que permita su construcción de manera evidente. (Anexo 3.3.1)  • Generar o actualizar el Análisis y Diseño. (Anexo 3.3.2)  • Generar o modificar el Registro de Rastreo. (Anexo 3.3.3.1) | AN  DI |
| A3. | Verificar el Análisis y Diseño y el Registro de Rastreo. (ver4) (Anexo 3.3.3.1, 3.3.3.3) | RE |
| A4. | Corregir los defectos encontrados en el Análisis y Diseño y en el Registro de Rastreo con base en el Reporte de verificación y obtener la aprobación de las correcciones. | AN  DI |
| A5. | Validar el Análisis y Diseño. (val2) | CL  RPU |
| A6. | Corregir los defectos encontrados en el Análisis y Diseño con base en el Reporte de validación y obtener la aprobación de las correcciones. | AN  DI |
| A7. | Elaborar o modificar Plan de Pruebas de Integración.(Anexo 3.3.3.2) | RPU |
| A8. | Verificar el Plan de Pruebas de Integración. (ver5) (Anexo 3.3.3.2) | RE |
| A9. | Corregir los defectos encontrados en el Plan de Pruebas de Integración con base en el Reporte de Verificación y obtener la aprobación de las correcciones. | RPU |
| A10. | Incorporar Análisis y Diseño, Registro de Rastreo y Plan de Pruebas de Integración como líneas base a la Configuración de Software. (Anexo 3.1) | RDM |
| A11. | Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas. (Anexo 3.3.3.4) | RDM |
| A12 | Verificaciones del progreso del proyecto (Anexo 6.0) | RDM |

## 

En caso de necesitar apoyo de organizaciones o productos externos, se seguirá un proceso basado en el IEEE 1062 - 1998, en el cual a partir del apartado 5 se establece el proceso a seguir, especificando la planeación, la estrategia organizacional, las prácticas generales, las prácticas de contratación, las responsabilidades y las definiciones. Además, expone que es necesaria una buena definición de lo que se necesita, cómo será utilizado y de qué manera lo aprovechará el equipo destinado a manejarlo. Describe las estrategias para escoger a un proveedor que sea de provecho y que se alinee con nuestras necesidades. Así como a establecer el contrato, monitorear el producto y los criterios de aceptación que deben tomarse en cuenta. Por último, incluye checklist que ayudan a evaluar cada una de las fases antes descritas por el mismo.

**Verificaciones y Validaciones**

| **Verificación o validación** | **Actividad** | **Producto** | **Rol** | **Descripción** |
| --- | --- | --- | --- | --- |
| **ver4** | A3 | Análisis y Diseño  Registro de Rastreo | RE | Verificar la claridad de la documentación del *Análisis y Diseño,* su factibilidad y la consistencia con la *Especificación de Requerimientos* y con el estándar de documentación requerido.  Verificar que el *Registro de Rastreo* contenga las relaciones adecuadas entre los requerimientos y los elementos de *Análisis y Diseño.* Documentar los defectos encontrados en un *Reporte de de Verificación. (Anexo 3.3.4.1)* |
| **val2** | A5 | Análisis y Diseño | CL  RPU | Validar que el *Análisis y Diseño* cumple con las necesidades y expectativas acordadas con el cliente. Documentar los defectos encontrados en un *Reporte de Validación.* |
| **ver5** | A8 | Plan de Pruebas de Integración | RE | Verificar la consistencia del *Plan de Pruebas de Integración* con el *Análisis y Diseño* y con el estándar de documentación requerido. Los defectos encontrados se documentan en un *Reporte de Verificación. (Anexo 3.3.4.2)* |

## **2.3 Construcción**

**Propósito**

El propósito es desarrollar el producto de Software delimitado en las dos fases anteriores, con base en el Modelo del Diseño de Software, para lograr producir componentes de software que correspondan al Análisis y al Diseño, así como la realización de pruebas unitarias.

**Descripción**

En esta fase se codificará el sistema, utilizando el Modelo del Diseño del Software, cuidando utilizar las prácticas especificadas y asegurando la lógica y sencillez del código. También, durante esta fase se asignan las tareas a los miembros de trabajo de acuerdo a su rol. El documento de Especificación de Requerimientos se tendrá como guía durante la elaboración.

Otra tarea importante que se realizará durante esta fase es la revisión del Registro de Rastreo y su posterior corrección, en caso de encontrar deficiencias.

Al concluir obtendremos el sistema de software con sus componentes, previamente establecidos, completos y validados.

**Objetivos**

1. Realizar el código de manera completa y lógica, siguiendo el Modelo del Diseño del Sistema y la Especificación de Requerimientos.
2. Obtener un código y componentes sencillos, adecuados y completos.
3. Desarrollar los componentes del sistema con las funcionalidades correspondientes.
4. Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual.

**Indicadores**

* Se realiza cada componente solicitado.
* Las actividades son realizadas en tiempo y orden.
* Se cumple con la calendarización de desarrollo.

**Metas**

* Componentes de software estables y con una buena usabilidad, código correcto

**Responsable**

* Programador y Diseñador

**Entradas**

* Plan de Desarrollo
* Análisis y Diseño
* Registro de Rastreo

**Salidas**

* Componentes de software
* Reporte de Actividades

**Actividades**

| **No.** | **Actividad** | **Roles involucrados** |
| --- | --- | --- |
| A1. | Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual. | RDM |
| A2. | Construir o modificar los Componentes de software:  • Implementar o modificar Componente(s) con base a la parte detallada del Análisis y Diseño.  • Definir y aplicar pruebas unitarias para verificar que el funcionamiento de cada componente esté acorde con la parte detallada del Análisis y Diseño.  • Corregir los defectos encontrados hasta lograr pruebas unitarias exitosas (sin defectos).  • Actualizar el Registro de Rastreo, incorporando los componentes construidos o modificados | PR |
| A3. | Verificar el Registro de Rastreo (Ver6). | RE |
| A4. | Corregir los defectos encontrados en el Registro de Rastreo con base en el Reporte de Verificación y obtener la aprobación de las correcciones. | PR |
| A5. | Incorporar Componentes y Registro de Rastreo como líneas base a la Configuración de Software. | RDM |
| A6. | Elaborar el Reporte de Actividades, registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas. | RDM |

**Verificaciones y Validaciones**

| **Verificación o validación** | **Actividad** | **Producto** | **Rol** | **Descripción** |
| --- | --- | --- | --- | --- |
| **ver6** | A3 | Registro de Rastreo | RE | Verificar que el *Registro de Rastreo* contenga las relaciones adecuadas entre los elementos de *Análisis y Diseño* y los componentes*.* Documentar los defectos encontrados en un *Reporte de de Verificación.* |

## **2.4 Implementación y pruebas**

**Propósito**

El propósito es identificar todos los defectos y los errores antes de que el producto sea entregado.

**Descripción**

En esta etapa buscamos realizar el proceso de actividades bien definidas que los testers del software necesitan para asegurar la calidad del software, cada actividad realizada en este proceso tendrá un proceso y una secuencia definida.

**Objetivos**

1. Verificar el trabajo de los productos como los requerimientos, diseño y código.
2. Verificar que se hayan cumplido con todos los requerimientos.
3. Comprobar que cumple las expectativas de los stakeholders.
4. Prevenir defectos en el producto de software.
5. Reducir el riesgo de una calidad de software insuficiente.
6. Arquitectura de clases extensible.
7. Software bien documentado.

**Indicadores**

* Defectos arreglados
* Requerimientos aprobados
* Defectos rechazados
* Revisión de requerimientos

**Metas**

* Identificar errores, fallas y defectos.
* Producto de alta calidad

**Responsable**

* Testers

**Entradas**

* Plan de Desarrollo
* Plan de Pruebas de Integración
* Registro de Rastreo
* Plan de Pruebas del Sistema
* Manual de Usuario
* Componentes de Software

**Salidas**

* Reporte de Pruebas de Integración
* Manual de Operación
* Reporte de Pruebas de Sistema
* Reporte de Actividades
* Software

**Actividades**

| **No.** | **Actividad** | **Roles involucrados** |
| --- | --- | --- |
| A1. | Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual. | RDM |
| A2. | Realizar integración y pruebas.  • Integrar los componentes en subsistemas o en el sistema del Software y aplicar las pruebas siguiendo el Plan de Pruebas de Integración, documentando los resultados en un Reporte de Pruebas de Integración.  • Corregir los defectos encontrados, con base en Reporte de Pruebas de Integración, hasta lograr una prueba de integración exitosa (sin defectos).  • Actualizar el Registro de Rastreo. | PR  RPU |
| A3. | Documentar el Manual de Operación o modificar el manual existente. | RM |
| A4. | Verificar el Manual de Operación (Ver7). | RE |
| A5. | Corregir los defectos encontrados en el Manual de Operación con base en el Reporte de Verificación y obtener la aprobación de las correcciones. | RM |
| A6. | Realizar las pruebas de sistema siguiendo el Plan de Pruebas de Sistema, documentando los resultados en un Reporte de Pruebas de Sistema. | RPU |
| A7. | Corregir los defectos encontrados en las pruebas de sistema con base en el Reporte de Pruebas de Sistema y obtener la aprobación de las correcciones. | PR |
| A8. | Documentar el Manual de Usuario o modificar el existente. | RM |
| A9. | Verificar el Manual de Usuario (Ver8). | RE |
| A10. | Corregir los defectos encontrados en el Manual de Usuario con base en el Reporte de Verificación y obtener la aprobación de las correcciones. | RM |
| A11. | Incorporar Software, Reporte de Pruebas de Integración, Registro de Rastreo, Manual de Operación y Manual de Usuario como líneas base a la Configuración de Software. | RDM |
| A12. | Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas. | RDM |

**Verificaciones y Validaciones**

| **Verificación o validación** | **Actividad** | **Producto** | **Rol** | **Descripción** |
| --- | --- | --- | --- | --- |
| **ver7** | A4 | Manual de Operación | RE | Verificar consistencia del *Manual de Operación* con el *Software* y con el estándar de documentación requerido. Documentar los defectos encontrados en un *Reporte de de Verificación.* |
| **ver8** | A9 | Manual de Usuario | RE | Verificar consistencia del *Manual de Usuario* con el *Software* y con el estándar de documentación requerido. Documentar los defectos encontrados en un *Reporte de de Verificación.* |

## **2.5 Cierre**

**Propósito**

El propósito de esta fase es concluir la realización de la documentación correspondiente al sistema de software desarrollado y que esta documentación cuente con todos los contenidos necesarios para disponer de un mantenimiento eficiente.

**Descripción**

A lo largo de esta fase se realizarán el Manual de Usuario y el Manual de Operación, que servirán como guía para el cliente en caso de un futuro mantenimiento al sistema de software o para que los futuros usuarios conozcan las funcionalidades del sistema y cómo aprovecharlas. Igualmente, se registrarán las actividades realizadas a lo largo de la elaboración del proyecto en un Reporte de Actividades, con el fin de tener un seguimiento de los proyectos elaborados en la empresa. Por último, se realizarán las mediciones correspondientes, según las Métricas del Proyecto, recabando la información obtenida en un Reporte de Mediciones, que también se entregará al cliente.

**Objetivos**

1. Incorporar de forma satisfactoria el manual de mantenimiento.
2. Generar un reporte de mediciones y sugerencias con el fin de mejorar en los siguientes proyectos.
3. Disponer de un panorama general de todo el proceso de desarrollo por medio del reporte de actividades.

**Indicadores**

* Se efectuará el reporte de verificación del manual de mantenimiento.
* Se revisará el reporte de mediciones y sugerencias de mejora.
* Las actividades concluirán en el tiempo establecido en el Plan de Desarrollo.

**Metas**

* Reporte final de actividades con fecha de inicio y fin de cada actividad.
* Cierre de proyecto.

**Responsable**

Administrador del Proyecto

**Entradas**

* Software

**Salidas**

* Manual de Mantenimiento
* Reporte de Actividades
* Reporte de Mediciones y Sugerencias de mejora

**Actividades**

| **No.** | **Actividad** | **Roles involucrados** |
| --- | --- | --- |
| A1. | Generar el reporte de mediciones y sugerencias de mejora. | RDM  ET |
| A2. | Documentar manual de mantenimiento o modificar el existente | RM |
| A3. | Verificar el manual de mantenimiento (Ver9). | RE |
| A4. | Corregir los defectos encontrados en el Manual de Mantenimiento con base en el Reporte de Verificación y obtener la aprobación de las correcciones. | RM |
| A5. | Incorporar Manual de Mantenimiento como línea base a la Configuración de Software. | RDM |
| A6. | Realizar reporte final de actividades registrando responsable, fecha de inicio y fin de cada una. | RDM |
| A7. | Formalizar la entrega del proyecto de acuerdo al protocolo establecido en el contrato de este. | ADMIN  RDM |

**Verificaciones y Validaciones**

| **Verificación o validación** | **Actividad** | **Producto** | **Rol** | **Descripción** |
| --- | --- | --- | --- | --- |
| **ver9** | A3 | Manual de Mantenimiento | RE | Verificar consistencia del *Manual de Mantenimiento* con la Configuración de *Software* y con el estándar de documentación requerido. Documentar los defectos encontrados en un *Reporte de de Verificación.* |

# 

# 3.0 Costos de calidad de software

En la norma ISO 9001-2015, se establece que las organizaciones deben evaluar los costos asociados con la calidad de sus productos y servicios, y que estas evaluaciones deben incluir los costos de prevención, los costos de evaluación y los costos de falla.

El proceso de control de calidad implica la implementación de medidas y actividades que buscan asegurar que los productos y servicios de una organización cumplan con los requisitos de calidad establecidos. Para esto, se adaptará el modelo de calidad de Feigenbaum, el cual es un modelo que busca garantizar que los productos o servicios ofrecidos cumplan con los estándares de calidad establecidos por las empresas. El control de calidad se realiza durante todo el ciclo de vida del producto o servicio, desde la fase de diseño hasta las fases de implementación y cierre.

Los costos se categorizan en “costos de control” y “costos de falla de control”, aplicando la metodología del Costo de la Calidad, que está respaldada por la norma ISO 9001:2015 y otros estándares relacionados con la calidad.

## 3.1. Costos de control

Los costos de control incluyen costos que se realizan para prevenir y detectar errores de software con el objetivo de reducirlos a un nivel aceptado. Estos costos son necesarios para garantizar que los productos y servicios cumplan con los estándares de calidad y las expectativas de los clientes.

Los costos de control pueden ser categorizados como costos de prevención y costos de evaluación.

Algunos ejemplos de costos de prevención son:

* Entrenamiento de personal
* Inversión en desarrollo de infraestructura
* Implementación de SQA preventiva.
* Realización de pruebas de software.

Algunos ejemplos de costos de evaluación son:

* Revisión de código
* Costos de pruebas de software
* Costos de aseguramiento de calidad externos

## 3.2. Costos de falla de control

Los costos de falla de control son los costos asociados con los productos o servicios que no cumplen con los estándares de calidad y, como resultado, deben ser reparados o reemplazados.

Los costos de falla de control pueden ser categorizados en costos de fallas internas y costos de fallas externas.

Algunos ejemplos de costos de fallas internas son:

* Costos de corrección de la documentación de procesos
* Costos de rediseño o de correcciones de diseño subsecuentes a la revisión del diseño
* Costos de reprogramar o corrección de programas en respuesta a los resultados de las pruebas
* Costos de repetidas revisiones de diseño y pruebas de regresión

Algunos ejemplos de costos de fallas externas son:

* Corrección de errores de software detectados durante el funcionamiento regular
* Daños pagados a los clientes en caso de una grave falla del software

# 4.0 Unidad de SQA

La unidad de SQA (Aseguramiento de Calidad de Software) debe asegurar que los productos y procesos de software se ajusten a los estándares, procedimientos, especificaciones y planes aplicables (ISO/IEC 12207:2017).

## **4.1 Actores**

Algunos de los actores que pueden formar parte de la unidad de SQA son:

**Gerente de SQA**: Es responsable de la gestión y supervisión de la unidad de SQA, asegurando que se cumplan los estándares, procedimientos, especificaciones y planes aplicables.

**Auditor de SQA**: Es responsable de realizar auditorías y revisiones de los productos y procesos de software para identificar posibles problemas y asegurar que se cumplan los estándares y procedimientos de calidad establecidos.

**Ingeniero de SQA**: Es responsable de participar en la revisión de los documentos de análisis y diseño del software, identificar posibles problemas y hacer recomendaciones para su corrección. También es responsable de realizar pruebas de verificación y validación para confirmar que se han cumplido los requisitos y objetivos aplicables.

**Especialista en normas y procedimientos**: Es responsable de mantener actualizados los estándares y procedimientos de calidad aplicables, asegurando que se cumplan los requisitos de calidad del software.

## **4.2 SQA en las fases de desarrollo y mantenimiento.**

### 4.2.1 Inicio

—-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 4.2.2 Análisis y diseño

Los miembros de la unidad de SQA deben realizar diversas actividades durante la fase de análisis y diseño de software para garantizar que se cumplan los estándares y procedimientos de calidad aplicables. Las actividades que se deben llevar a cabo son:

**Revisión de los documentos de análisis y diseño:** Los miembros de la unidad de SQA deben revisar los documentos de análisis y diseño para identificar posibles problemas y asegurarse de que se cumplan los estándares y procedimientos de calidad establecidos. También deben asegurarse de que los documentos estén completos, precisos y coherentes.

**Identificación y documentación de problemas:** Si se encuentran problemas durante la revisión de los documentos de análisis y diseño, los miembros de la unidad de SQA deben documentarlos y asegurarse de que se tomen medidas para corregirlos.

**Verificación de requisitos y objetivos:** Los miembros de la unidad de SQA deben verificar que los requisitos y objetivos especificados en los documentos de análisis y diseño se hayan cumplido correctamente. Deben realizar pruebas de verificación y validación para confirmar que el software se ajusta a los requisitos y objetivos aplicables.

**Auditorías y revisiones:** Los miembros de la unidad de SQA deben realizar auditorías y revisiones periódicas de los documentos de análisis y diseño para asegurarse de que se cumplan los estándares y procedimientos de calidad aplicables.

**Capacitación y orientación:** Los miembros de la unidad de SQA también deben proporcionar capacitación y orientación a otros miembros del equipo de desarrollo de software en cuanto a los estándares y procedimientos de calidad aplicables.

Template: <https://drive.google.com/file/d/1Y_6bA2qWIJer4nTjt6Z1snVSYIqGVUXO/view?usp=sharing>

### 4.2.3 Construcción

—-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 4.2.4 Implementación y pruebas

—-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 4.2.5 Cierre

—-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

# 5.0 Anexo

En este anexo se presenta la relación entre los procesos y productos de este documento y los estándares y modelos más importantes que deberán ser utilizados para mantener el nivel de calidad deseado.

## **5.1 Gestión de configuración.**

ISO/IEC 20000 - Service Management

8 - Funcionamiento del sistema de gestión de servicios

**8.2.6 - Gestión de la configuración.**

En este estándar se menciona el cómo se deberán clasificar (nomenclatura basada en los ítems de configuración (CI). Esto para asegurar que la selección CI cumple con el orden necesario al momento de solicitar la información que se necesite de estos.

IEEE 828-2012 - “Standard for configuration management in systems and software engineering”

8- identificación de configuración

**8.2.2 - Identificación de ítems de configuración**

La identificación y selección los ítems de configuración será basada en lo que nos presenta esta sección, como es el hecho que la determinante para seleccionar algún CI es si se necesitará este en caso de que el proyecto necesite reiniciar a una versión base anterior en el ciclo de vida para así, poder avanzar en el ciclo de construcción

IEEE 828-2012 - “Standard for configuration management in systems and software engineering”

8- identificación de configuración

**8.2.5.1 - Establecimiento de repositorios controlados**

Indica qué debe contener (como mínimo) cada repositorio que vaya a ser utilizado por el equipo.

Plantillas CI:

<https://advisera.com/20000academy/es/documentation/proceso-de-gestion-de-configuracion/>

<https://advisera.com/20000academy/es/documentation/base-de-datos-de-gestion-de-la-configuracion-iso-20000/#:~:text=El%20objetivo%20de%20este%20documento,problemas%20y%20errores%20conocidos%20asociados>.

## **5.2 Inicio.**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

## **5.3 Análisis y diseño.**

En esta fase se debe realizar un completo análisis de la información obtenida del documento de requerimientos para posteriormente realizar el diseño del sistema aplicando la metodología, estándares y estrategias del IEEE 1016-2009.

Que primeramente explica que es un SDD (Descripción del Diseño de Software), la cual es la representación de una parte del software que se utiliza para comunicar información a las partes interesadas.

También se especifican los requisitos de los lenguajes de diseño (notaciones y otros esquemas de representación) que deben utilizarse para los SDD. Esta norma es aplicable a los lenguajes de descripción del diseño, pero puede utilizarse para documentos en papel y otros medios de descripción.

### 5.3.1 Descripción de componentes

IEEE 1016-2009 - Systems design

4- Descripción del diseño Contenido de la información

* **4.2 Identificación de SDD**

En esta sección se explica qué datos son los que deberá tener la Descripción de diseño de software, como lo son: Autoría, contexto, uno o más lenguajes (diagramas, imágenes, descripción) para cada punto de vista del diseño, Ámbito de aplicación, glosario, etc.

Plantillas (basadas en IEEE 1016-2009) : <https://www.slideshare.net/peny_mg/sdd-software-des-sample>

<https://slideplayer.com/slide/6560772/>

<https://github.com/jam01/SDD-Template/blob/master/template.md>

<https://cengproject.cankaya.edu.tr/wp-content/uploads/sites/10/2017/12/SDD-ieee-1016-2009.pdf> (17-18)

### 5.3.2 Modelo de diseño

IEEE 1016-2009 - Systems design

4- Descripción del diseño Contenido de la información

* **4.5 Diseño de puntos de vista**

IEEE 1016-2009 - Systems design

4- Descripción del diseño Contenido de la información

* **4.6 Diseño de elementos**

IEEE 1016-2009 - Systems design

4- Descripción del diseño Contenido de la información

* **4.9 Diseño de Lenguajes**

Estas secciones abarcan la explicación de qué hacer con toda esa información que se analizará del documento de requerimientos, en el diseño de elementos se explica de qué manera se puede ir descomponiendo el sistema en pequeños subsistemas y a su vez, en componentes (clases, data stores, módulos, procesos, etc) , marcando cada una de sus interrelaciones, esto se debe realizar desde distintos puntos de vista (diseño puntos de vista y de lenguajes), es decir que al momento de diseñar cada parte del proyecto se debe analizar desde el punto de vista de distintos roles del equipo, como es tester, programador, responsable de manuales, y deberá ser descrito en distintos lenguajes (diagramas de clase, diagramas de flujo, descripciones, imágenes, mapas conceptuales, etc.) de manera que sean lo más comprensible posible para cada uno de los integrantes involucrados en el desarrollo del sistema.

IEEE 1016-2009 - Systems design

4- Descripción del diseño Contenido de la información

* **3.2.3 Verificación del diseño y papel del diseño en la validación.**

Los planes de prueba se verán relacionados por el SDD, al proporcionar la visión general necesaria para comprender la implementación, justificación de decisiones y trazabilidad hasta los requisitos previamente establecidos (Registro de rastreo).

Plantilla registro de Rastreo: <https://repositorio.unne.edu.ar/bitstream/handle/123456789/1617/RIUNNE_AC_Ferraro_MdelosA_.pdf?sequence=1&isAllowed=y>

Se menciona de igual manera que el SDD estará sujeto a verificaciones de diseño para determinar si este aborda los requisitos establecidos, logra cualidades previstas (seguridad, mantenibilidad). IEEE Std 1028 - 2008 -**B17.**

### 5.3.3 Documentos de control.

Las actividades de control de estado son necesarias para reunir información y reportar el estado de los elementos de configuración. Para cada producto se debe definir el estándar de documentación cumpliendo con las características mencionadas en la descripción del producto.

ISO 9001-2015 - Quality Management Systems - Requirements.

7- Apoyo

**7.5.3 Control de la información documentada.**

Esta sección explica cómo deben resguardarse los documentos de control y que seguimiento se les debe dar para que puedan ser controlados, modificados y accesados adecuadamente.

#### 5.3.3.1 Modelo de diseño

* Modelo de Diseño: Documento que explica cómo se conforma el sistema de Software.

Plantilla:

<https://drive.google.com/file/d/1qATKz-Nxv5x4IyPSWx_21WJw8BfNOmOX/view?usp=sharing>

Checklist:

IEEE 1012-2016 - System, Software and Hardware Verification and Validation

[**https://drive.google.com/file/d/1Z8bHzP2Ig1L7Sqof4G0rDq7cBoE-41mj/view?usp=sharing**](https://drive.google.com/file/d/1Z8bHzP2Ig1L7Sqof4G0rDq7cBoE-41mj/view?usp=sharing)

#### 5.3.3.2 Plan de pruebas de unidad e integración

* Plan de pruebas de integración: Documento que identifica las pruebas requeridas para el cumplimiento de los requerimientos especificados.

Plantilla:

IEEE 829-2008 - IEEE Standard for Software and System Test Documentation

[**https://drive.google.com/file/d/17kKnJ3FdKdBrlPVwWALxHtr9RtusMLFh/view?usp=sharing**](https://drive.google.com/file/d/17kKnJ3FdKdBrlPVwWALxHtr9RtusMLFh/view?usp=sharing)

Checklist:

IEEE 1012-2016 - System, Software and Hardware Verification and Validation

**Table 1c - V&V tasks, inputs and outputs (pg. 114)**

Las pruebas de integración se realizarán con base en el proceso descrito en el mismo estándar de la plantilla.

IEEE 829-2008 - IEEE Standard for Software and System Test Documentation

Anexo C - Testing tasks.

**5.4.5 Testing tasks during the development process: Test activity.**

#### 5.3.3.3 Registro de rastreo

* Registro de Rastreo: Documento que describe la relación entre los requerimientos, elementos análisis y diseño, componentes y planes de pruebas.

Plantilla:

ISO- 9001-2015

<https://drive.google.com/file/d/1pCs2c9nyjHr3EhfeakWiBQz4X11ArZHs/view?usp=sharing>

Checklist:

IEEE 1012-2016 - System, Software and Hardware Verification and Validation

**Table 1c - V&V tasks, inputs and outputs (pg. 113)**

#### 5.3.3.4 Reporte de actividades

* Reporte de actividades: Documento que captura el registro periódico de actividades

Plantilla:

IEEE 1058 - 1998 - Software Project Management Plan

<https://drive.google.com/file/d/1YkR4ViVW4FNG0UFfWC6wOBe_ckif4OjM/view?usp=sharing>

5.3.4 Revisiones

3.3.4.1 ver4: Verificación del Análisis y Diseño y Registro de Rastreo.

IEEE 1028 - 2008 - Standard for Software Reviews and Audits.

5. Technical reviews.

Las revisiones técnicas se encargan de evaluar el producto de software por un equipo de personal calificado. En el caso del Análisis y Diseño, se espera determinar su concordancia con el documento de Especificación de Requisitos, es decir, que se pueda identificar una concordancia entre los requerimientos definidos y los componentes expuestos en el Análisis y Diseño, de manera que el diseño propuesto sea la manera eficiente de llevarlo a cabo. Por otro lado, se espera comprobar que el Registro de Rastreo esté completo, conteniendo una relación entre los requisitos y los componentes identificados.

El estándar propone un líder de revisión, un anotador, el o las personas que se encargan de hacer dicha revisión y el coordinador. Se tendrán como entradas ambos documentos a revisar, el documento de Especificación de Requerimientos y los objetivos que se deben lograr, en nuestro caso utilizaremos la checklist de cada documento. Como resultado, se tendrá un reporte con las observaciones, correcciones y adecuaciones comentadas durante la reunión de revisión.

4.3.4.1 ver5: Verificación del Plan de Pruebas de Integración.

IEEE 1028 - 2008 - Standard for Software Reviews and Audits.

5. Technical reviews.

Las revisiones técnicas se encargan de evaluar el producto de software por un equipo de personal calificado. En el caso del Plan de Pruebas de Integración se busca evaluar su calidad, asegurar que se hayan incluido todos los casos de prueba necesarios y que las prácticas a seguir tengan concordancia con los estándares establecidos y las prácticas óptimas para ello.

El estándar propone un líder de revisión, un coordinador, un anotador y el o las personas encargadas de realizar dicha revisión. Como entradas, se entregará el Plan de Pruebas de Integración, los estándares y las plantillas, además de la checklist que ayude a verificar su estructura y contenido. Como salidas, se esperará un reporte con las observaciones, correcciones y adecuaciones comentadas durante la reunión de revisión.

# 6.0 Control de progreso del proyecto

IEEE 12007-1996 Software life cycle processes

Estándar internacional que describe el ciclo de vida del software y proporciona una forma de guiarse para el desarrollo, mantenimiento y operación del software.

Nos brinda información como las directrices de diversos procesos que conforman el ciclo de vida de un proyecto de software en diversas secciones como verificaciones, validaciones, actividades de control de riesgos, seguimiento y control de riesgos.

IEEE 1058-1998 Standard for Software Project Management Plans

Proporciona la preparación de un plan de gestión de proyectos de software que aplicado de la manera correcta puede ser aplicado a proyectos de cualquier tamaño y complejidad.

Este estándar establece los requisitos para la revisión y aprobación del plan, esto asegura que todos los miembros del equipo del proyecto estén alineados con los objetivos del proyecto y se comprometan con los plazos y entregables definidos en el plan de gestión de proyectos de software.

IEEE 1490 Adoption of PMI Standard

El estándar IEEE 1490 también proporciona una forma detallada sobre cómo hacer un seguimiento del progreso del proyecto de software. Esto incluye la medición del progreso del proyecto a lo largo del tiempo, la comparación del progreso real con el plan de proyecto original y la identificación de las desviaciones del plan original.

## **4.4 Construcción.**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

## **4.5 Implementación y pruebas.**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

## **4.6 Cierre.**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

**—----------------------------------------------------------------------------------------------------**

## **4.7 Corrección de defectos encontrados.**

ISO-9001:2015

6.

**Sección 6.1 - Acciones para abordar riesgos y oportunidades**

Este estándar menciona la forma en la que se pueden gestionar las acciones para corregir errores que se presenten en las diferentes etapas.

ISO/IEC 12207 - procesos del ciclo de vida de software

6. procesos de apoyo de ciclo de vida

**6.7 proceso de auditoría**

**6.7 Proceso de solución de problemas.**

Se emplea para definir, controlar y mejorar los procesos del ciclo de vida de software, en este podemos obtener la forma en la que hacemos los procesos de auditorías, revisión y validación.

ISO 13485:2016

Este estándar será utilizado con algunos ajustes para adaptarlo a nuestro detallado de corrección de defectos de software, dado que en este estándar se especifican la forma de llevar muchas de las actividades en esta sección.

ISO 9001 - sistemas de gestión de calidad

4.

**4.14 acciones preventivas y correctivas**

Este estándar afirma que la información sobre los problemas encontrados es un motor de mejora del proceso de desarrollo. Si se detecta un punto débil en el proceso, se emprenderá una acción de mejora lo antes posible. Cuando se detecten errores en el software, aparte de la corrección inmediata de los mismos, se llevará a cabo un análisis de las causas raíz. Si es posible, se tomarán medidas para mejorar el proceso y evitar que se repitan estos tipos específicos de errores.

ISO 14001:2015

10

**10.2 no conformidad y acciones correctivas**

Nos dice como organización que se necesita para reaccionar ante los defectos y se puede tener nuestros procedimientos de acciones correctivas.

Plantillas:

<https://cmsmedtech.com/free-iso-13485-software-validation-template/>

<https://www.iso-9001-checklist.co.uk/download/ISO-14001-2015-environmental-management-system-template-sample.pdf>

<https://www.whittingtonassociates.com/resources/iso-140012015-gap-checklist/>