Contents

[1. INTRODUCCIÓN 3](#_Toc389753570)

[1.1 ANTECEDENTES 4](#_Toc389753571)

[1.2. OBJETIVOS. 5](#_Toc389753572)

[1.2.1. OBJETIVO GENERAL. 5](#_Toc389753573)

[1.2.2. OBJETIVO ESPECIFICO. 5](#_Toc389753574)

[1.3 MISIÓN. 6](#_Toc389753575)

[1.4. VISION 7](#_Toc389753576)

[1.5. POLITICAS DE CALIDAD. 8](#_Toc389753577)

[1.6 SLOGAN 8](#_Toc389753578)

[2. PLAN DE ASEGURAMIENTO DE CALIDAD DE SOFTWARE (SQAP). 10](#_Toc389753579)

[2.1. PROPOSITO. 10](#_Toc389753580)

[2.1.1. OBJETIVO. 10](#_Toc389753581)

[2.1.2. DESCRIPCIÓN. 10](#_Toc389753582)

[2.1.3. ALCANCE. 11](#_Toc389753583)

[2.2. DOCUMENTOS DE REFERENCIA. 12](#_Toc389753584)

[2.3 GESTIÓN. 13](#_Toc389753585)

[2.3.1. ORGANIZACIÓN. 13](#_Toc389753586)

[2.3.2. TAREAS. 14](#_Toc389753587)

[2.3.3 ROLES Y RESPONSABILIDADES. 15](#_Toc389753588)

[2.4. DOCUMENTACIÓN. 17](#_Toc389753589)

[2.4.1. Especificación de Requisitos de Software.- 17](#_Toc389753590)

[2.4.2 Descripción del Diseño del Software (SDD).- 19](#_Toc389753591)

[2.4.3 Plan de Verificación y Validación.- 22](#_Toc389753592)

[Proceso de gestión 23](#_Toc389753593)

[# De Reporte: 29](#_Toc389753594)

[# De Reporte: 31](#_Toc389753595)

[2.5 ESTÁNDARES, PRÁCTICAS Y CONVENCIONES. 33](#_Toc389753596)

[2.5.1 ESTANDAR DE CODIFICACION. 33](#_Toc389753597)

[2.5.2 ESTANDAR DE COMENTARIOS. 34](#_Toc389753598)

[2.5.3. RESPONSABLES DE VERIFICAR EL CUMPLIMIENTO. 35](#_Toc389753599)

[2.6. REVISIONES Y AUDITORIAS. 35](#_Toc389753600)

[Evaluación de la calidad de los productos: 41](#_Toc389753601)

[Revisar el ajuste al proceso: 42](#_Toc389753602)

[Revisión Técnica Formal (RTF): 44](#_Toc389753603)

[Requerimientos Mínimos: 44](#_Toc389753604)

[Los elementos mínimos que deberán ser revisados son: 44](#_Toc389753605)

[Agenda 45](#_Toc389753606)

[Fase I – Inicial 45](#_Toc389753607)

[2.7. GESTION DE CONFIGURACIÓN. 45](#_Toc389753608)

[2.8 GESTIÓN DE PROBLEMAS Y ACCIONES CORRELATIVAS. 46](#_Toc389753609)

[# De Reporte: 47](#_Toc389753610)

[2.9. HERRAMIENTAS, TECNICAS Y METODOLOGÍAS. 49](#_Toc389753611)

[2.10. CONTROL DE CODIGO. 50](#_Toc389753612)

[2.11. CONTROL DE MEDIOS. 50](#_Toc389753613)

[2.12. CONTROL DE SUMINISTROS Y SUBCONTRATOS. 51](#_Toc389753614)

[2.13 RECOLECCIÓN, MANTENIMIENTO Y RETENCION DE REGISTROS. 51](#_Toc389753615)

**INTRODUCCIÓN**

**A LA**

**EMPRESA**

# 1. INTRODUCCIÓN

El software cada vez está más presente en nuestra actividad y en los objetos que nos rodean y que usamos. La calidad del producto software es una preocupación cada vez mayor en el ámbito informático y cuyos resultados inmediatos se aprecian en todas las actividades en donde se utilicen computadoras.

Las necesidades de calidad del usuario sobre el software, contribuyen a especificar los requerimientos de calidad externa y estos a su vez los requerimientos de calidad interna. El cumplimiento de los requerimientos de calidad interna, externa y en uso se deben de comprobar en un proceso que permita evaluar la calidad a través de las métricas. Con este enfoque de tres niveles se intenta cubrir las perspectivas del usuario, desarrollador y el producto mismo. El desarrollo de productos software no está ausente de ofrecer calidad. Dicho nivel de calidad, incluido en los productos, considera muchas actividades dentro del desarrollo de los proyectos software, la gestión de la calidad dentro de este tipo de proyectos puede estandarizarse dentro de la organización y certificarse a la comunidad de clientes. La calidad en el desarrollo de software es alcanzable si la organización elige su norma de referencia y define y desarrolla su Plan de Calidad.

La Calidad cuesta, pero resulta más costoso el no tenerla en un ambiente competitivo como el actual. La calidad es el rasgo diferenciador entre las organizaciones capaces de destacarse en el mercado y aquellas que simplemente sobreviven o desaparecen.

# 1.1 ANTECEDENTES

La empresa de desarrollo "HAJIMESOFT" es una organización dedicada al desarrollo de sistemas, fundada en la gestión 2013 por unos jóvenes visionarios capaces de destacarse en el mercado de software.

Actualmente se cuenta con personal especializados en las áreas de:

* Jefe Comercial.
* Jefe de Proyectos.
* Análisis.
* Diseñador.
* Implementadores.

Se cuenta con una organización descentralizada; es decir, cada uno de los integrantes del grupo de desarrollo participa libremente en las reuniones, brindando opiniones y generando ideas. El director del grupo de desarrollo es el jefe de proyectos.

Jefe de Comercial

Implementador

Diseñador

Analista

Jefe de Proyectos

# 1.2. OBJETIVOS.

## 1.2.1. OBJETIVO GENERAL.

"HAJIMESOFT" tiene como objetivo desarrollar software de alta calidad que satisfagan las necesidades del cliente.

## 1.2.2. OBJETIVO ESPECIFICO.

* Desarrollar software a través de rigurosos análisis con el cliente.
* Diseñar e implementar con estándares de calidad.
* Definir los requerimientos de calidad a ser verificados.
* Describir las tareas de calidad a realizar.

# 1.3 MISIÓN.

“HAJIMESOFT” está comprometida con la satisfacción del cliente, diseñando, fabricando y entregando productos de software de la más alta calidad.

Se establecen, controlan y alcanzan los objetivos establecidos en cada uno de los productos de software a desarrollar con el fin de encontrar o exceder las expectativas de nuestros clientes.

Para alcanzar la calidad total de los productos y la mejora continua, se utilizan los siguientes estándares:

* + IEEE STD-730: "El aseguramiento de la calidad de software es un modelo planificado y sistemático de todas las acciones necesarias a fin de asegurar que el ítem o producto cumpla con los requerimientos técnicos establecidos".
  + IEEE STD-729: Análisis de requerimientos de software.
  + IEEE STD-828: Estándar para planes del manejo de las configuraciones de software.
  + IEEE STD-829: Estándar para documentación de pruebas de software.
  + IEEE STD-830: Estándar para especificación de requerimientos de software.  
    IEEE STD-1012: Estándar para la planificación de verificación y validación de software.
  + IEEE STD-1063: Estándar para los manuales de usuarios de software.

# 1.4. VISION

Actualmente, la satisfacción hacia el uso de un producto puede marcar una gran diferencia en el mercado de productos similares. Es así como el desarrollo de productos de software que satisfacen las expectativas de los clientes y usuarios harán la diferencia entre dos organizaciones que desarrollan productos que compiten en el mercado.

La preocupación por ofrecer productos acompañados de altos niveles de calidad no es una actividad nueva.

La empresa reconoce la necesidad de lograr que el personal que conforma a la misma, invierta todo su talento en el desarrollo de su trabajo, con un alto nivel de participación, además de reconocer el valor del conocimiento y pretender promoverlo, estructurarlo y hacerlo operativo, para obtener mayor eficiencia y eficacia en el desarrollo de los productos, aumentando su productividad.

Por esta razón la misión principal de Hajimesoft es el desarrollo de productos de software de calidad, para satisfacer plenamente las necesidades de los clientes y promover el desarrollo del personal adquiriendo mayor conocimiento y experiencia, de esta forma se pretende establecer una organización altamente competitiva en el campo laboral.

# 1.5. POLITICAS DE CALIDAD.

HAJIMESOFT, asume el compromiso formal de desarrollar su actividad con los mayores estándares de calidad, adoptando un Sistema de Gestión de la Calidad basado en la Norma ISO 9001:2008 para todos sus procesos, con el propósito de mejorar su competitividad, satisfacer plenamente los requerimientos de sus clientes, afianzar su participación en el mercado nacional y que permitan actuar en el mercado internacional

La dirección considera esta política como elemento integral de sus negocios y se encarga de su difusión, compresión y cumplimiento, fijando los siguientes lineamientos básicos:

* Cumplir con las fechas de entrega pactadas con los clientes.
* Mejorar los procesos y el sistema de gestión de calidad en forma continua.
* Capacitar a todo su personal en base a sus necesidades y a las nuevas tecnologías, incentivando su integración.
* Interpretar los requerimientos y expectativas del cliente, generando soluciones que aporten valor.
* Establecer una estrategia de mejora continua cuyos pilares son la planificación, ejecución, verificación en todos sus procesos.
* Se deberá buscar la certificación en los estándares para lograr una mejor operación, calidad, productividad, innovación  y eficiencia en la operación y en los productos que generamos o distribuimos; no se deberán buscar la certificación por el simple documento.

# 1.6 SLOGAN

"Calidad y Responsabilidad"

**PLAN SQA**

# 2. PLAN DE ASEGURAMIENTO DE CALIDAD DE SOFTWARE (SQAP).

## 2.1. PROPOSITO.

### 2.1.1. OBJETIVO.

Definir un conjunto de normas y actividades con el fin de asegurar la calidad en el desarrollo de software.

### 2.1.2. DESCRIPCIÓN.

A través de la implantación del SQAP se pretende cumplir con los elementos de calidad de software, los cuales son:

* Correcto
* Eficiente
* Fiable
* Facilidad de uso
* Facilidad de mantenimiento
* Seguridad e integridad
* Portabilidad

Para obtener productos de software con gran competitividad en el mercado, y poder satisfacer plenamente los requerimientos de los clientes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Modelo de Calidad** | | | |
| **Calidad Externa e Interna** |  | | |
| **Característica** | **Ponderación** | **Subcaracterística** | **Ponderación** |
| **Funcionalidad** |  | Aplicabilidad |  |
|  |  | Precisión |  |
|  |  | Interoperabilidad |  |
|  |  | Seguridad |  |
|  |  | Conformidad de Funcionalidad |  |
| **Fiabilidad** |  | Madurez (hardware/software/datos) |  |
|  |  | Tolerancia a fallos |  |
|  |  | Recuperabilidad (datos, proceso, tecnología) |  |
|  |  | Conformidad de fiabilidad |  |
| **Usabilidad** |  | Entendibilidad |  |
|  |  | Facilidad de aprendizaje |  |
|  |  | Operabilidad |  |
|  |  | Atractividad |  |
|  |  | Conformidad de usabilidad |  |
| **Eficiencia** |  | Comportamiento en el tiempo |  |
|  |  | Utilización de recursos |  |
|  |  | Conformidad de eficiencia |  |
| **Facilidad de**  **Mantenimiento** |  | Analizabilidad |  |
|  |  | Cambiabilidad |  |
|  |  | Estabilidad |  |
|  |  | Testeabilidad  Conformidad de facilidad de mantenimiento |  |
| **Portabilidad** |  | Adaptabilidad |  |
|  |  | Instabilidad |  |
|  |  | Co-existencia |  |
|  |  | Reemplazabilidad |  |
|  |  | Conformidad de portabilidad |  |

Se pretende la aplicación del SQAP para cualquier proyecto de software a desarrollarse por la empresa.

### 2.1.3. ALCANCE.

El SQAP cubre las fases del ciclo de vida de desarrollo de software

**Ciclo de Vida de Desarrollo de Software PUDS (Proceso Unificado de Desarrollo de Software)**

* Requerimientos
* Diseño
* Implementación
* Pruebas
* Instalación

Los componentes del software se presentarán de acuerdo al campo de aplicación del mismo, a sus especificaciones y requerimientos.

Se pretende implementar productos de software capaz de responder a sus objetivos en cualquier condición de funcionamiento y operación, tener una documentación completa acerca del desarrollo del mismo con el fin de facilitar su mantenimiento.

## 2.2. DOCUMENTOS DE REFERENCIA.

* IEEE STD 730-1998, IEEE Standard for Software Quality Assurance Plans.
* IEEE STD 730.1-1995, IEEE Guide for Software Quality Assurance Planning.
* IS-1(2001) – Proyecto de Ingeniería de Software
* IS-2 (2001) - Modelo de Calidad

## 2.3 GESTIÓN.

### 2.3.1. ORGANIZACIÓN.

Las organizaciones involucradas en la implementación del producto de software son:

* Organización de Grupo de Desarrollo.
* Organización del Consultor o Especialistas en SQA.
* Organización del Cliente.
* Organización del grupo SQA.
* Un Fiscal (en caso de que sea necesario).

**2.3.1.1. ORGANIZACIÓN DE GRUPO DE DESARROLLO**

Es la organización encargada del desarrollo del software, su trabajo está regido de acuerdo a las especificaciones y contratos establecidos por el cliente. Debe informar sobre sus actividades al consultor.

**2.3.1.2. ORGANIZACIÓN DEL CONSULTOR O ESPECIALISTAS EN SQA**

Es la organización fiscalizadora del producto de software, teniendo la potestad delegada por el cliente, para establecer normas, supervisar el desarrollo y hacer cumplir con los contratos establecidos.

Generalmente el consultor toma el papel de intermediario entre el cliente y el grupo de desarrollo.

**2.3.1.3. ORGANIZACIÓN DEL CLIENTE.**

La organización del cliente depende de la estructura de su empresa o de la función que realice. Varía de acuerdo al proyecto de software que se esté desarrollando.

**2.3.1.4. ORGANIZACIÓN DEL GRUPO SQA**

Es la organización que discute las normas y sugerencias generadas por el consultor, para luego aceptarlas y liberar versiones sucesivas del SQAP para el desarrollo e implementación del software, esta organización se obtiene de las tres organizaciones anteriores.

### 2.3.2. TAREAS.

La relación de tareas asociadas con el ciclo de vida de desarrollo de software y las actividades de la SQA son las siguientes, las cuales se ejecutarán durante el desarrollo del producto de software:

|  |  |
| --- | --- |
| **CICLO DE VIDA DEL SOFTWARE (PUDS)** | **TAREAS Y ACTIVIDADES ASOCIADAS AL CICLO DE VIDA DEL SOFTWARE** |
| Requerimientos | Análisis de requisitos de software  Generación de especificaciones  Revisión de especificaciones  Revisión de las especificaciones de software |
| Diseño | Diseño preliminar  Generación de especificaciones de diseño preliminar  Revisión del diseño preliminar  Diseño detallado  Generación de especificaciones de diseño detallado  Revisión del diseño detallado  Revisión del diseño preliminar  Revisión del diseño detallado del software  (Ambos versus las especificaciones) |
| Implementación | 1. Codificación 2. Generación de código 3. Revisión de código   Revisión de código versus la documentación generada |
| Pruebas | Elaboración de pruebas de unidad y generación de resultados  Revisión de resultados  Elaboración de las pruebas de unidad y de integración del software  Revisión de los resultados de las pruebas  Revisión de las pruebas funcionales y evaluación de los resultados |
| Instalación y Prueba Final | Instalación del producto software  Prueba final bajo ambiente real  Generación de resultados de prueba  Revisión de resultados  Revisión de la instalación del software y evaluación de  Resultados |

**ACTIVIDAD DE GRUPO SQA.**

|  |  |
| --- | --- |
| **Actividad** | **Entregable Asociado** |
| Elaboración del Plan de SQA | Plan de SQA |
| Identificar propiedades de Calidad | Plan de SQA |
| Evaluación de la calidad de los productos | Informe de revisión de SQA |
| Revisar el ajuste al proceso | Informe de revisión de SQA |
| Realizar Revisión Técnica Formal | Informe de Revisión Técnica Formal |
| Evaluar y ajustar el Plan de SQA | Documento de Evaluación y Ajustes al Plan de SQA |
| Evaluación final de SQA | Informe final de SQA |
| Revisar la entrega semanal | Entrega semanal de SQA |

### 2.3.3 ROLES Y RESPONSABILIDADES.

**2.3.3.1. RESPONSABILIDADES DEL GRUPO DE DESARROLLO.**

Esta organización es responsable de:

* Desarrollar un producto de software en base a lo de finido en el SQAP y los contratos establecidos con el cliente.
* Generar la debida documentación definida en la SQAP acerca de cada una de sus actividades con el fin de llevar un control de las mismas.
* Entregar la documentación de desarrollo que se exige en el plan (SQAP).

**2.3.3.2. RESPONSABILIDADES DEL CLIENTE.**

Esta organización es responsable de:

* Proveer la información necesaria para el desarrollo del software con el fin de satisfacer sus necesidades.
* Brindar los recursos y condiciones necesarias para elaborar el software.
* Participar activamente en la organización del SQA para obtener óptimos resultados.

**2.3.3.3. RESPONSABILIDAD DE LA SQA**

Actividades del grupo SQA:

* Establecimiento del plan SQA para el proyecto.
* Participar en el desarrollo de la descripción del proceso de software.
* Revisión de las actividades de ingeniería del software para verificar su ajuste al proceso del software.
* Auditoria de los productos de software designados para verificar el ajuste con los definidos como parte del proceso de software.
* Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.
* Registrar lo que no se ajuste a los requisitos e informar a sus superiores.
* Coordinar el control y la gestión de cambios.
* Analizar las métricas del software.

Esta organización es responsable de:

* Garantizar la calidad del producto de software desarrollado.
* Implantar normas y actividades para el desarrollo del software.
* Realizar reuniones para resolver los posibles conflictos durante el desarrollo del software.
* Aprobar y publicar el SQAP.
* Observar las deficiencias en el SQAP.
* Mejorar el SQAP, recomendando modificaciones o correcciones con el fin de obtener resultados óptimos.
* Autorizar la implantación del software.
* Enfoque de gestión de calidad.
* Tecnologías (métodos y herramientas).
* Revisiones Técnicas Formales.
* Estrategia de pruebas.
* Control de la documentación y de cambios.
* Procedimientos que aseguren ajustes a los estándares.
* Mecanismos de medición y generación de informes.

## 2.4. DOCUMENTACIÓN.

Se identifica toda la documentación que gobernará el desarrollo, validación y verificación, mantenimiento y uso del software.

La documentación mínima que exige el estándar para garantizar que la implementación del software satisface los requisitos es la siguiente:

* Especificación de Requisitos de Software
* Descripción del Diseño del Software
* Plan de Verificación y Validación
* Informe de Verificación y Validación
* Documentación de Usuario

### 2.4.1. Especificación de Requisitos de Software.-

Esta documentación es elaborada por el desarrollador, y se basa en el estándar ANSI /

IEEE-Std 830 “Guía para especificaciones de requerimientos de software” (**SRS**).

La SRS deberá describir claramente y de forma precisa cada uno de los requerimientos del

Software, tal como: funciones, rendimiento, restricciones de diseño y atributos.

**MODELO A USAR PARA EL CONTENIDO DEL SRS**

* 1. INTRODUCION
  2. Objetivo
  3. Alcance
  4. Definiciones, acrónimos y abreviaciones
  5. Referencias
  6. Revisión
  7. DESCRIPCION GENERAL

2.1 Perspectiva del producto

2.2 Funciones del producto

2.3 Características de los usuarios

2.4 Restricciones generales

2.5 Asunciones y dependencias

* 1. ESPECIFICACION DE REQUERIMIENTOS

3.1 Requerimiento Funcional

3.1.1. Introducción

3.1.2. Entradas

3.1.3. Procesos

3.1.4. Salidas

3.1.5. Interfaces externas

3.1.5.1. Interfaces del usuario

3.1.5.2. Interfaces del hardware

3.1.5.3. Interfaces del software

3.1.6 Requerimientos de rendimiento

3.1.7 Representación del diseño

3.1.8 Cumplimientos con estándares

3.1.9 Limitaciones del hardware

3.1.10 Atributos

3.1.10.1. Disponibilidad

3.1.10.2. Seguridad

3.1.10.3. Mantenibilidad

3.1.10.4. Transferencia / Conversión

3.1.10.5 Prevenciones

3.1.11 Otros requerimientos

3.1.11.1. Base de datos

3.1.11.2. Operaciones

3.1.11.3. Adaptaciones

APENDICES

INDICE

ANEXOS

### 2.4.2 Descripción del Diseño del Software (SDD).-

La generación y documentación de la descripción del diseño de Software se basa en el estándar ANSI / IEEE – Std 1016 “RECOMMENDED PRACTICE FOR SOFTWARE DESCRIPTIONS”

|  |  |  |  |
| --- | --- | --- | --- |
| **Organización de la SDD dentro de Vistas de Diseño** | | | |
| **VISTA DE DISEÑO** | **ALCANCE** | **ATRIBUTOS DE ENTIDAD** | **EJEMPLOS DE REPRESENTACIONES** |
| Descripción de descomposición  Descripción de dependencia  Descripción de interfaces  Descripción de detalle | Partición del sistema dentro de entidades de diseño.  Descripción de las relaciones entre entidades y recursos del sistema.  Lista de cada interfaz de diseñador, programador, o pruebas necesarias para conocer el uso de la entidad de diseño que componen el sistema.  Descripción de los detalles de diseño internos en una entidad. | Identificación, tipo, objetivo, función, subordinación.  Identificación, tipo, objetivo, dependencias, recursos.  Identificación, función, interfaces.  Identificación, procesamiento, datos. | Diagrama de descomposición, jerarquía y lenguaje natural.  Diagrama de estructura.  Tablas de parámetros.  Diagrama de flujos. |

**MODELO A USAR PARA EL CONTENIDO DEL SDD**

1. INTRODUCION
   1. Objetivo
   2. Alcance
   3. Definiciones, acrónimos y abreviaciones
2. REFERENCIAS
3. DESCRIPCION DE DESCOMPOSICION

3.1. Descomposición de módulo

3.1.1. Descripción del módulo 1

3.1.2. Descripción del módulo 2

3.1.n. Descripción del módulo n

3.2. Descomposición de procesos concurrentes

3.2.1. Descripción del proceso 1

3.2.2. Descripción del proceso 2

3.2.n. Descripción del proceso n

3.3. Descomposición de datos

3.3.1. Descripción de la entidad de datos 1

3.3.2. Descripción de la entidad de datos 2

3.3.n. Descripción de la entidad de datos n

1. DESCRIPCION DE DEPENDENCIA
   1. Dependencia entre módulos
   2. Dependencia entre procesos
   3. Dependencia entre datos
2. DESCRIPCION DE INTERFACES
   1. Interfaces de módulo

5.1.1. Descripción del módulo 1

5.1.2. Descripción del módulo 2

5.1.n. Descripción del módulo n

5.2 Interfaces de procesos

5.2.1. Descripción del proceso 1

5.2.2. Descripción del proceso 2

5.2.n. Descripción del proceso n

6. DISEÑO DETALLADO

6.1. Diseño detallado del módulo

6.1.1. Detalle del módulo 1

6.1.2. Detalle del módulo 2

6.1.n. Detalle del módulo n

6.2. Diseño detallado de datos

6.2.1. Detalle de entidad de datos 1

6.2.2. Detalle de entidad de datos 2

6.2.n. Detalle de entidad de datos n

APENDICES

INDICE

ANEXOS

### 2.4.3 Plan de Verificación y Validación.-

La generación y documentación de la descripción del Plan de Verificación y Validación (SWP) es la siguiente:

**MODELO A USAR PARA EL CONTENIDO DEL SWP**

1. OBJETIVO
2. ALCANCE
3. DEFINICIONES, ACRONIMOS Y ABREVIACIONES
4. ORGANIZACIÓN RESPONSABLES
5. CICLO DE VIDA DE VERIFICACION Y VALIDACION

APENDICE

INDICE

La organización responsable por las tareas de verificación y validación del software es la organización de SQA comandada por la organización del consultor, la cual interactúa con la organización de desarrollo para alcanzar los objetivos del plan. En casos necesarios de conflictos extremos entre el consultor y el desarrollador se recurrirá al cliente.

**2.4.3.1. CICLO DE VIDA DE VERIFICACIÓN Y VALIDACIÓN**

El plan se basa en los procesos en los que interviene la Verificación y Validación.

* Proceso de gestión.
* Proceso de Adquisición.
* Proceso de Suministro.
* Proceso de Desarrollo.
* Proceso de Operación.
* Proceso de Mantenimiento.

|  |  |  |
| --- | --- | --- |
| 1 Fase de Concepto V&V | | |
| Procesos | Actividades | Tareas |
| Proceso de gestión El proceso de gestión implica la administración del proyecto. | **Gestión del esfuerzo de la verificación y validación**  La actividad de Gestión implica:  -Una revisión continua del esfuerzo de V&V  - Seguimiento de los SVVP según sea necesario basándose en programaciones del proyecto y su estado actual.  -Establecimiento de auditorías y evaluaciones. | **Tareas de Gestión**  1) Generación SVVP 2) Evaluación del cambio propuesto/base 3) Examen de la gestión del esfuerzo de V & V  4) Apoyo a las técnicas de gestión 5) Interfaz con orden de procesos 6) Identificación de oportunidades de mejora. |
| **Proceso de Adquisición**  El segundo proceso a realizar es el de adquisición este proceso inicia con definir una necesidad existente, posteriormente se plantea una solicitud, luego se selecciona un proveedor y finaliza con la adquisición de la propuesta por medio de una aceptación de la misma. Este se compone de una sola actividad: | **Apoyo de adquisición V & V**  Hace posible la preparación de contratos, el monitoreo, la aceptación y conclusión del proyecto. | **Tareas en la Adquisición**  1) Determinación del alcance. 2) Planificación de la interfaz. 3) Evaluación de los requisitos del sistema 4) Apoyo para la aceptación de adquisición. |
| **3.Proceso de Suministro**  En este proceso se prepara la propuesta para responder a determinada solicitud de un cliente para proveer un sistema, posteriormente se organizan los recursos necesarios para el desarrollo y gestión del proyecto. | **Planificación de V & V**  La actividad de planificación V & V aborda la iniciación, preparación de actividades de respuesta, contrato, planificación, ejecución, control, revisión, evaluación, entrega y conclusión. | **Tareas para Planificación de V&V**  1) Planificación de la interfaz entre el esfuerzo de V&V y proveedor 2) Contrato de verificación |
| **Proceso de Desarrollo.**  Es uno de los más complejos y largos en el proceso de V&V ya que conforma todas las actividades que realiza el desarrollador, el análisis de requerimientos, el diseño, la codificación de software, integración, pruebas, instalación y aceptación del producto de software. | **1. Actividad 1: V&V de Concepto**  El objetivo de la V&V del concepto es verificar la asignación de requisitos del sistema, validar la solución seleccionada y garantizar que ninguna hipótesis falsas se hallan incorporado en la solución. | **Tareas de Concepto V&V**  1) Evaluación de documentación de concepto. 2) Análisis de criticidad. 3) Análisis de asignación de requisitos. 4) Análisis de trazabilidad. 5) Análisis de riesgo. 6) Análisis de la seguridad. |
| **V&V de Requisitos**  El objetivo de los requisitos de V & V es asegurar la exactitud, integridad, precisión, capacidad de prueba y consistencia de los requisitos de software de sistema. | **Tareas para V& V de requisitos**  1) Análisis de trazabilidad  2) Evaluación de los requisitos de Software 3) Análisis de la interfaz 4) Análisis de criticidad 5) Identificación plan de V&V. 6) Generación de plan de prueba de aceptación de V & V 7) Evaluación de la gestión de configuración 8) Análisis de posibles fallos. 9) Análisis de la seguridad |
| **V & V Del Diseño**  El objetivo de diseño V & V es demostrar que el diseño es una transformación correcta, exacta y completa de los requisitos de software y que no se introducen funciones no deseadas. | **Tareas para la V&V del Diseño**  1)Análisis de trazabilidad 2)Evaluación de diseño de Software 3)Análisis de la interfaz 4) Análisis de criticidad 5) Componente V & V plan de generación de prueba 6) Generación de plan de prueba de integración V & V 7) Generación de diseño de prueba componente V & V 8) Generación de diseño de prueba de integración V & V 9) Generación de diseño de prueba de sistema de V & V 10) Generación de diseño de la prueba de aceptación de V & V 11) Análisis de riesgo 12) Análisis de la seguridad |
| **V & V de Test**  El objetivo es evaluar las pruebas de software, pruebas de integración de software, pruebas de calificación de software, pruebas de integración de sistema y sistema de pruebas de clasificación | **Tareas para la V&V del Test**  1) Análisis de trazabilidad  2) Generación de procedimiento de la prueba de aceptación de V & V 3) Ejecución de prueba de integración V & V 4) Ejecución de prueba de sistema de V & V 5) Ejecución de prueba de aceptación V & V 6) Análisis de riesgo 7) Análisis de la seguridad |
| **V&V de Instalación y entrega**  El objetivo de la V&V de Instalación es verificar y validar la exactitud de la instalación del software en el entorno de destino. | **Tareas para la Instalación y Carga**  1)Auditoría de configuración de instalación 2)Verificación de la instalación 3)Análisis de riesgo 4)Análisis de la seguridad 5)Análisis de riesgo 6) Final de la generación de informes |
| **Proceso de Operación.**  El proceso de operación consiste en llevar a cabo la operación y uso del sistema por el usuario final en un ambiente operacional. | **Operación V & V**  El objetivo de la operación V & V es evaluar nuevas restricciones en el sistema, evaluar cambios en el sistema propuesto y su impacto en el software y evaluar los procedimientos operativos para la exactitud y facilidad de uso. | **Tareas de la Operación V & V**  1) Evaluación de nuevas restricciones 2) Evaluación de los procedimientos de funcionamiento 3) Análisis de riesgo 4) Análisis de la seguridad |
| **Proceso de Mantenimiento.**  Se evidencia cuando se presentan cambios en el software y deben tenerse los recursos y planes necesarios para realizar modificaciones, migraciones y retiros o remplazos en el software. | **V&V de Mantenimiento**  El objetivo de mantenimiento V & V es evaluar los cambios en el sistema propuesto de software y su impacto en el software, evaluar las anomalías se descubren durante la operación, evaluar los requerimientos de migración, evaluar los requisitos de jubilación. | **Tareas de V&V Mantenimiento**  1) Revisión SVVP 2) Evaluación de anomalías 3) Análisis de criticidad 4) Evaluación de la migración 5) Evaluación de jubilación 6) Análisis de riesgo 7) Análisis de la seguridad 8) Análisis de riesgo 9) Tarea iteración |

**2.4.3.2. Reporte sumario de fase V&V.-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **REPORTE SUMARIO DE FASE V&V**  **FASE:** | | | **Pág.**  …...... | |
| **# De Reporte:**  **Lugar:** | | **Fecha: / /**  **Hora:** | | |
| 1. Descripción de las tareas de V&V realizadas:        1. Sumario de resultados de tareas:        1. Sumario de anomalías y resolución:        1. Evaluación de calidad del software:        1. Recomendaciones: | | | | |
|  |  | | |  |
| **Equipo de Trabajo:**  **Nombre: Firma** | | |
|  | | | | |

**2.4.3.3. Reporte de Anomalías.-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **REPORTE DE ANOMALIAS**  **FASE:** | | | **Pág.**  …...... | |
| # De Reporte: **Lugar:** | | **Fecha: / /**  **Hora:** | | |
| a) Descripción y ubicación:      b) Impacto:      c) Causa:       1. Critibilidad:        1. Recomendaciones: | | | | |
|  |  | | |  |
| **Equipo de Trabajo:**  **Nombre: Firma** | | |
|  | | | | |

**2.4.3.4. Reporte Final de V&V.-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **REPORTE FINAL DE V&V**  **FASE:** | | | **Pág.**  …...... | |
| # De Reporte: **Lugar:** | | **Fecha: / /**  **Hora:** | | |
| a) Resumen de todas las tareas V&V, durante el ciclo de vida del software:      b) Resumen de resultados de tareas:      c) Resumen de anomalías, resolución:       1. Evaluación total de la calidad del software:        1. Recomendaciones: | | | | |
|  |  | | |  |
| **Equipo de Trabajo:**  **Nombre: Firma** | | |
|  | | | | |

## 2.5 ESTÁNDARES, PRÁCTICAS Y CONVENCIONES.

### 2.5.1 ESTANDAR DE CODIFICACION.

Las normas de codificación se definen de la siguiente forma:

* El software debe ser subdividido en módulos independientes, de acuerdo al diseño establecido.
* La documentación de un programa debe tener el siguiente formato:

- Nombre del programa

Objetivo

- Nombre de las entradas:

Base de Datos

Archivos

Registros

Formatos de pantalla

- Nombre de las salidas:

Base de Datos

Archivos

Registros

Formatos de pantalla

Reportes

- Nombre de los archivos de actualización:

Base de Datos

Archivos

Registros

- Nombre del autor

Fecha de creación

- Historial de actualizaciones

Versión

Fecha de cambio

Objetivo de cambio

* Cada módulo debe explicar sus funciones
* La declaración de cualquier variable debe estar comentada, explicando su función.
* Debe existir una sola instrucción por cada línea de código.
* Cada función debe de estar debidamente documentada, explicar la funcionalidad, la función de cada parámetro.
* Cada mensaje de error o excepciones deben de indicar el lugar donde se originó y la función o procedimiento en el cual se produjo.
* Los nombres de las funciones deben de indicar su funcionalidad.
* Cada clase implementada debe de estar comentada de la siguiente forma:

Nombre

Fecha y hora de creación

Autor

Nombre del módulo al que pertenece

Funcionalidad

### 2.5.2 ESTANDAR DE COMENTARIOS.

* Un comentario debe explicar porque se realiza alguna acción.
* Los comentarios dentro de un módulo deben estar separados del código.
* Utilizar comentarios de más de una línea para realizar descripciones, y comentarios de una línea para realizar especificaciones.

### 2.5.3. RESPONSABLES DE VERIFICAR EL CUMPLIMIENTO.

Los responsables de realizar la verificación del cumplimiento con los estándares definidos son:

* El jefe del equipo de desarrollo.
* La organización del SQA.

## 2.6. REVISIONES Y AUDITORIAS.

Los responsables de estas revisiones es la organización del SQA, con la participación de todo elemento de la organización que tengan que ver con los requerimientos, tales como: los diseñadores del software, agentes de pruebas.

Las revisiones y auditorias de los resultados del desarrollo se realizan a medida que se terminan cada una de las fases del ciclo de vida de desarrollo de software, con el fin de:

* Conocer el progreso alcanzado en el desarrollo.
* Evaluar el ajuste a los requerimientos del sistema.
* Evaluar la eficiencia en el trabajo.

Se deben llevar a cabo, al menos, las siguientes revisiones y auditorias:

**Revisión de los Requisitos de Software (SRR):**

La SRR se genera para:

* Evaluar las especificaciones de requerimientos del software (SRS).
* Asegurar que los requerimientos establecidos en la SRS, sean los correctos y estén completos.
* Garantizar la calidad, viabilidad e integridad de los requerimientos establecidos.

Los requerimientos de revisiones de SRS en la SRR son los siguientes:

1. Fiable
2. Completo
3. Depurable
4. Modificable
5. Consistente
6. Libre de ambigüedades
7. Utilizable durante la fase de operación y mantenimiento.
8. Inspeccionar que la relación entre los requerimientos y sus derivados sea la adecuada.

**Revisión del Diseño Preliminar (PDR):**

La PDR es realizada para evaluar la suficiencia técnica del SDD preliminar, antes de comenzar con el diseño detallado, define los siguientes puntos:

* Evaluar el progreso, consistencia y suficiencia técnica del alcance de diseño con los requerimientos funcionales de la SRS.
* Verificar la existencia y compatibilidad de las interfaces entre el software, el hardware y los usuarios finales.
* Determinar un diseño de software que cumpla con los requerimientos.

Para la PDR se toman como requerimientos de revisiones los siguientes puntos:

* Revisar que se detallen todas las interfaces con otro software, sistemas de comunicación, etc. Para una adecuada identificación de interfaces y de un diseño óptimo.
* Revisar que exista un análisis del diseño para verificar la compatibilidad con los requerimientos críticos.
* Revisar que se establece los requerimientos del factor humano.

**Revisión del Diseño Crítico (CDR):**

La CDR es generada para determinar la aceptabilidad de cómo la SDD cumple con la SRS. Evalúa la suficiencia técnica, integridad del diseño detallado del software, antes de comenzar a codificar para establecer que el diseño detallado satisface los requerimientos de la SRS.

Para la CDR se toman como requerimientos de revisión los siguientes puntos:

* Evaluar la compatibilidad del diseño detallado con la SRS.
* Examinar la representación de datos en forma de diagramas lógicos, algoritmos, almacenamiento y representación de datos.
* Determinar la compatibilidad e integridad de requerimientos de interfaces.
* Establecer que todas las interfaces internas y externas incluyendo interacciones con la base de datos sean expresadas.

**Revisión del Plan de Verificación y Validación (SVVPR):**

La SVVPR es generado para la evaluación de:

* Los métodos de Verificación y Validación definidos en el SVVP.
* El cumplimiento durante el desarrollo del software con el SVVP.

Se realizan revisiones incrementales para asegurar que los métodos de verificación y validación del software sean los adecuados para los datos del software que se está desarrollando.

Se toman como criterios de requerimientos de la SVVPR los siguientes puntos:

* Reportes para una adecuada documentación de resultados de todas las revisiones, verificaciones y pruebas basadas en los requerimientos listados en el SVVP.
* Descripciones adecuadas de la configuración del software, para ser examinado, incluyendo pruebas de soporte de software y hardware.
* Planes de pruebas y diseño de pruebas para asegurar que todos los requerimientos son examinados.
* Procedimientos y situaciones de pruebas para asegurar que las entradas sean las adecuadas para el software.
* Programación de pruebas identificando que pruebas serán realizadas, cuando y por quien van a ser realizadas.

**Auditoria Funcional:**

Esta verificación es realizada antes de la entrega del software, para verificar que todos los requerimientos especificados en la SRS fueron alcanzados.

La verificación funcional compara el código con los requerimientos documentados del software, como se estableció en SRS. Su propósito es asegurar que el código hace todo y solo lo que se indica en la documentación establecida por la SRS.

Se definen los siguientes puntos:

* Nomenclatura
* Número de identificación de la especificación
* Número de ítem de configuración
* La especificación de requerimientos de software
* Copia de código objeto
* Listado actualizado de ítems de configuración especificados
* El reporte de verificación y validación del software
* Listado del cumplimiento exitoso de pruebas funcionales
* Listado de todo lo planificado y pruebas que no fueron ejecutadas
* Actualizaciones para la documentación previamente liberada deberá ser revisada para asegurar su exactitud y consistencia.

**Auditoria Física (PA):**

Esta verificación es realizada para verificar que el software y su documentación son internamente consistentes y están listas para su entrega.

La verificación física compara el código con su documentación de soporte, su propósito es asegurar que la documentación a ser entregada describa correctamente el código.

La documentación necesaria para realizar la verificación física es la siguiente:

* Descripción del diseño del software SDD
* Productos de software
* Documentación asociada

Para proveer evidencia de un adecuado control del contenido del sistema y consistencia del equipo de auditoria se examina lo siguiente:

* Los documentos de especificación del sistema para formatos y cumplimientos
* Reportes funcionales para discrepancia y acciones tomadas
* Descripción del diseño para símbolos, etiquetas, referencias y descripción de datos
* Los manuales para formatos de completitud y cumplimiento con la descripción de datos.
* Los elementos de Sw liberados en el medio son óptimos para transferir y transmitir
* Identificar los cambios en los ítems de configuración de datos

**Auditorías del Proceso (IPA):**

Estas verificaciones serán desarrolladas dentro de los procesos de desarrollo del software, como ser el diseño para verificar la consistencia del diseño incluyendo:

* Código versus documentación del diseño
* Especificaciones de interfaces hardware / software.
* Implementación de diseño versus requerimientos funcionales

El objetivo es verificar la consistencia del producto a través del proceso del desarrollo para determinar que:

* Las interfaces hardware/software sean consistentes con el diseño de requerimientos en la SRS.
* Los requerimientos funcionales de la SRS, sean completamente probados por el SVVP.
* El diseño del producto especificado en la SDD, satisface los requerimientos funcionales del SRS.
* El código es consistente con la SDD.

**Revisiones de gestión:**

Estas revisiones son realizadas periódicamente para evaluar la ejecución del SQAP. Estas revisiones deberán ser realizadas por el elemento organizacional del consultor.

Se podrían planificar otras revisiones, como por ejemplo la revisión de la documentación de usuario.

Para cada tipo de revisión, se debe explicar:

* Su objetivo.
* Qué producto es el que se evalúa.
* Sus propósitos.
* Cuál es el elemento organizativo responsable de llevar a cabo la revisión.
* Cuáles son los elementos organizativos que deben tomar parte de la revisión.
* Cuáles son los requisitos de revisión.
* Dónde deben documentarse los resultados de la revisión.

Para cada tipo de auditoria se debe explicar:

* Su objetivo.
* Cuál es el elemento organizativo responsable de llevar a cabo la auditoria.
* Dónde deben documentarse los resultados de la auditoria.
* Cuáles son las entradas para la auditoria.

Se definen los tres tipos de revisiones (Evaluación de la calidad de los productos, Revisar el ajuste al proceso y Revisión Técnica Formal – RTF –), sus objetivos y mecanismos.

### Evaluación de la calidad de los productos:

**Objetivo:**

Revisar los productos que se definieron como claves para asegurar la calidad.

Detectar desviaciones en los objetivos de calidad definidos e informar a los responsables para que sean corregidas.

**Mecanismo:**

Se revisan los productos para verificar que cumplan con los estándares y con los objetivos de calidad definidas para el producto.

Se debe verificar que no queden correcciones sin resolver en los informes de revisión previos, si se encuentra alguna no resuelta, debe ser incluida en la siguiente revisión. Se debe identificar, documentar y seguir la pista a las desviaciones encontradas y verificar que se hayan realizado las correcciones.

Como salida se obtiene el Informe de revisión de SQA, que contiene todas las desviaciones o defectos encontrados durante la revisión. Este informe debe ser distribuido a los responsables del producto y se debe asegurar que ellos son conscientes de las desviaciones o discrepancias encontradas y de las acciones correctivas que deben realizar.

### Revisar el ajuste al proceso:

**Objetivo**:

Revisar si los productos se obtuvieron realizando las actividades que se indican en el Modelo de Proceso.

**Mecanismo**:

Se revisan los productos que se definen como claves para verificar el cumplimiento de las actividades definidas en el proceso, durante todo el ciclo de vida del software.

Se debe recoger la información necesaria de cada producto, buscando hacia atrás los productos previos que deberían haberse generado y son entrada para el producto objeto de revisión, para poder establecer los criterios de revisión y evaluar si el producto cumple con las especificaciones.

Esta información se obtiene de los siguientes documentos:

* Plan del Proyecto
* Plan de la iteración
* Plan de Verificación

Se debe verificar si todos los pasos del proceso de desarrollo son seguidos apropiadamente.

Antes de comenzar, se debe verificar en los informes de revisión previos que todas las desviaciones fueron corregidas, si no es así, las faltantes se incluyen para ser evaluadas.

Como salida se obtiene el Informe de revisión de SQA correspondiente a la evaluación de ajuste al Proceso, que contiene todas las desviaciones o defectos encontrados durante la revisión. Este informe debe ser distribuido a los responsables de las actividades y se debe asegurar que ellos son conscientes de las desviaciones o discrepancias encontradas y de las acciones correctivas que deben realizar.

### Revisión Técnica Formal (RTF):

**Objetivo:**

Descubrir errores en la función, la lógica ó la implementación de cualquier producto del software, verificar que satisface sus especificaciones, que se ajusta a los estándares establecidos, señalando las posibles desviaciones detectadas.

**Mecanismo:**

Es un proceso de revisión riguroso, su objetivo es llegar a detectar lo antes posible, los posibles defectos o desviaciones en los productos que se van generando a lo largo del desarrollo. Por esta característica se adopta esta práctica para productos que son de especial importancia.

En la reunión participan el responsable de SQA e integrantes del equipo de desarrollo.

Se debe convocar a la reunión formalmente a los involucrados, informar del material que ellos deben preparar por adelantado, llevar una lista de preguntas y dudas que surgen del estudio del producto a ser revisado.

Como salida se obtiene el Informe de RTF.

## Requerimientos Mínimos:

## Los elementos mínimos que deberán ser revisados son:

* Especificación de Requerimientos
* Modelo de Diseño y Descripción de la Arquitectura
* Plan de Verificación y Validación
* Plan de Gestión del Proyecto
* Plan de Gestión de Configuración
* Diseño vs. Especificación de requerimientos
* Implementación vs. Diseño
* Verificación vs. Especificación de requerimientos

## Agenda

En esta sección se detallan todas las revisiones de calidad que se realizarán durante todo el proyecto, organizadas por fase e iteración.

### Fase I – Inicial

#### Iteración I

|  |  |  |  |
| --- | --- | --- | --- |
| **Entregable** | **Realizado** | **Revisión** | **Tipo de revisión** |
| Nombre del entregable o producto a revisar | Fase, iteración y semana en que se debe realizar la versión del producto a revisar | Semana, si se quiere también la fecha, en la que se realizará la revisión del entregable o producto | Tipo de revisión que se realizará: Evaluación de la calidad de los productos, Revisar el ajuste al proceso o Revisión Técnica Formal |

Y así sucesivamente para cada una de las fases del ciclo de vida de desarrollo de software.

## 2.7. GESTION DE CONFIGURACIÓN.

El objetivo del SQA en esta área es asegurar que se realizan las actividades de gestión de configuración establecidas en el Plan de Configuración y que se realizan según lo establecido en el proceso. Se pueden definir las siguientes actividades mínimas que se deberían realizar:

* Asegurar que se generó la Línea Base del proyecto en el momento establecido en el modelo de proceso.
* Asegurar que la Línea Base del proyecto generada es correcta.
* Se verifica periódicamente que el Responsable de SCM mantiene apropiadamente el control de la línea base, así como el registro completo de cambios para requerimientos, diseño, código, verificación y documentación.
* Se monitorean los procedimientos del Comité de Control de Cambios para verificar que son efectivamente realizados como se especificaron en el Plan de configuración.

## 2.8 GESTIÓN DE PROBLEMAS Y ACCIONES CORRELATIVAS.

En esta sección se describen las prácticas y procedimientos que se van a utilizar para la notificación, seguimiento y resolución de problemas de software, así como las responsabilidades organizativas. El propósito de un sistema de Gestión de Problemas y Acciones Correlativas es:

* Asegurar que todos los problemas de documentan, se corrigen y no caen en el olvido.
* Asegurar que se evalúa la validez de los informes de problemas.
* Realimentar al desarrollador y el usuario sobre el estado de los problemas.
* Proporcionar datos para medir y predecir la calidad y fiabilidad del software.

Cualquier problema en el producto de software que sea encontrado durante el ciclo de vida de desarrollo de software, debe ser reportado a través de un reporte en el cual se detalla la fecha de cuando fue encontrado el problema, una identificación preliminar del mismo, descripción, etc., este reporte debe ser firmado por los que identificaron el problema, debe ser entregado a la organización responsable de los problemas.

La organización responsable de los problemas del software, es la organización del SQA, comandada por la organización del consultor, estas organizaciones son las encargadas de determinar el cronograma, lugar y temario, para llevar a fijar la acción correctiva del problema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **REPORTE FINAL DE PROBLEMAS** | | | **Pág.**  …...... | |
| # De Reporte: **Lugar:** | | **Fecha: / /**  **Hora:** | | |
| a) Identificación del problema:      b) Descripción:      c) El evento ejecutado cuando se presentó el problema es:       1. Posibles orígenes del problema: | | | | |
|  |  | | |  |
| **Equipo de Trabajo:**  **Nombre: Firma** | | |
|  | | | | |

Las acciones a seguir para corregir los problemas presentados se describen de la siguiente manera:

* Antes de la identificación de la presencia de un problema se debe buscar los posibles orígenes del mismo sin desechar ninguna de las posibilidades, para esto se debe tomar dos rutas para generar el reporte de problema completo y consistente.
  1. Registrar las causas sospechosas del origen del problema, esto asegura que no se descarta ninguna situación posible.
  2. Registrar causas adicionales
  3. Registrar causas plenamente identificadas, esto es, causas verificadas por detectores del problema
  4. Registrar otras causas externas o relacionadas a las identificadas anteriormente
* Generar el reporte del problema, este debe estar completamente detallado y de acuerdo a los puntos que contiene el mismo, este reporte debe ser entregado a la organización responsable por los problemas.
* La organización responsable por los problemas, convocan a una reunión técnica, en la cual participarán además de la organización responsable, los elementos de las organizaciones afectadas por el problema, quienes describirán el problema y darán recomendaciones necesarias para solucionar el problema.
* La especificación de acciones correctivas generada en la reunión técnica, será entregada a los elementos organizacionales afectados por el problema para que estos implementen las acciones correctivas respectivas.

## 2.9. HERRAMIENTAS, TECNICAS Y METODOLOGÍAS.

En esta sección se identifican todas las herramientas, técnicas y metodologías que se van a utilizar en el desarrollo que apoyan el Aseguramiento de Calidad.

Algunas de las herramientas son:

* Utilidades del sistema operativo WINDOWS 7.
* Utilidades del sistema operativo WINDOWS 7.
* Documentación de ayuda.
* Instaladores.

Las técnicas que ayudan a la evaluación o mejora de la calidad son:

* ANSI / IEEE – STD 830 Guide for Software Requirements Specifications
* ANSI / IEEE – STD 1016 Recommended Practice for Software Design Descriptions
* ANSI / IEEE – STD 1008 Standard for Software Unit Testing
* ANSI / IEEE – STD 1063 Standard for Software User Documentation
* ANSI / IEEE – STD 1028 Standard for Software Reviews and Audits

Las metodologías de Aseguramiento de Calidad serán conjuntos integrados de técnicas, de entre los anteriores.

## 2.10. CONTROL DE CODIGO.

En esta sección se definen los métodos, técnicas y facilidades que se van a utilizar para controlar el almacenamiento y mantenimiento de versiones del código.

Se especifica un procedimiento de control del Código que:

* Defina cuál es el software que se va a controlar.
* Describa un método estándar para identificar, etiquetar y catalogar el software.
* Liste la localización física del software bajo control
* Describa la localización, forma de mantenimiento y de uso de las copias de
* seguridad.
* Describa los procedimientos para distribución de copias.
* Identifique la documentación que se verá afectada por los cambios.
* Describa los procedimientos para la construcción de una nueva versión

## 2.11. CONTROL DE MEDIOS.

En esta sección se definen los métodos y facilidades que se van a utilizar para proteger el medio físico de accesos no autorizados y daños y degradaciones inesperadas, y las organizaciones responsables para realizar este control.

La organización responsable por esta tarea es la organización de desarrollo, con la supervisión de la organización de la SQA.

Se debería asegurar que:

* Está garantizado el almacenamiento y recuperación de software.
* El software está accesible únicamente para aquellos que lo necesitan.
* Se controla el entorno para que no se degrade el medio físico en el que se almacena el software.
* Se almacenan copias del software crítico y del código en línea base fuera de las instalaciones de la organización.

**Medio de Almacenamiento.-**

El medio del programa de computadora se define como aquellos medios sobre los cuales los datos son almacenados.

Se utilizarán los siguientes medios:

* Los discos duros como dispositivos primarios.
* Los CD´s como almacenamiento secundario, para guardar las copias de seguridad.
* La documentación respectiva sobre el desarrollo de software (papel).

**Proceso de copias de seguridad**

Las copias de seguridad serán realizadas a la finalización de cada sesión de trabajo, registrándose la fecha y hora de copia de seguridad.

**Puntos de Control**

Para el acceso no autorizado, se debe asignar cuentas privilegiadas, cada usuario que interactúa con el software tendrá su propia cuenta de acuerdo al cargo que desempeñe.

Se cuenta con la integridad de la Base de Datos para la protección de los datos.

Se realizará una revisión periódica del software con el fin de que funcione óptimamente

## 2.12. CONTROL DE SUMINISTROS Y SUBCONTRATOS.

En esta sección se explica de qué forma se va a asegurar que el software comprado o subcontratado cumple los requisitos técnicos.

## 2.13 RECOLECCIÓN, MANTENIMIENTO Y RETENCION DE REGISTROS.

Las organizaciones responsables por las tareas de esta sección, es la organización del consultor, en coordinación con la organización de la SQA.

Se identifica aquella documentación que se debe retener, y se especifican los métodos y facilidades que se utilizarán para recolectar, proteger y mantener esta documentación.

También se especificará el período de retención para cada tipo de registro.

Se puede registrar no sólo documentación, sino también los medios físicos que contienen las versiones de los programas y los materiales utilizados en las pruebas, para asegurar la repetición de los tests en el futuro.

Los documentos que son requeridos son los siguientes:

* Plan de garantía de calidad del software.
* Especificación de requerimientos del software.
* Descripción del diseño del software.
* Plan de verificación y validación del software.
* Documentación del usuario.

El mantenimiento de los registros del software, será realizado por versiones sucesivas de actualizaciones de las mismas, para esto se lleva un registro de actualizaciones de la documentación.

Los documentos verificados y validados, deben ser documentados en libros impresos, con tres copias de cada documento y almacenado en lugares diferentes y ambiente adecuados.

La retención de registros se realizará en cada finalización de las fases del ciclo de vida de desarrollo de software y según los puntos de verificación y validación.