

PLAN DE SQA

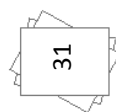
EMPRESA TERASOFT

Contenido

1.	INTRODUCCION.....	3
1.1.	ANTECEDENTES.....	3
1.2.	OBJETIVOS.....	3
1.2.1.	OBJETIVO GENERAL.....	3
1.2.2.	OBJETIVOS ESPECIFICOS.....	4
1.3.	MISION.....	4
1.4.	VISION.....	4
1.5.	POLITICAS DE CALIDAD.....	4
1.6.	SLOGAN.....	5
2.	PLAN DE ASEGURAMIENTO DE CALIDAD DE SOFTWARE.....	6
2.1.	PROPOSITO.....	6
2.1.1.	OBJETIVO.....	6
2.1.2.	DESCRIPCIÓN.....	6
2.1.3.	ALCANCE.....	6
3.	GESTION.....	7
3.1.	ORGANIZACIÓN.....	7
3.1.1.1.	ORGANIZACIÓN DE LA SQA.....	8
3.2.	ORGANIGRAMA DE SQA.....	8
3.3.	TAREAS.....	8
3.4.	RESPONSABILIDADES.....	10
3.4.1.	RESPONSABILIDADES DEL GRUPO DE DESARROLLO.....	10
3.4.2.	RESPONSABILIDADES DEL CLIENTE.....	11
3.4.3.	RESPONSABILIDADES DE LA SQA.....	11
3.5.	EVOLUCIÓN DEL SQAP.....	12
3.5.1.	Control De Cambio.....	12
3.5.2.	CUANDO HACER MODIFICACIONES.....	13
3.5.3.	IMPLANTACIÓN DE MODIFICACIONES.....	13
3.5.4.	PUBLICACIÓN DEL SQAP.....	13
4.	DOCUMENTACION.....	14
4.1.2.	DESCRIPCIÓN DEL DISEÑO DEL SOFTWARE (DDS).....	15



4.1.3.	PLAN DE VERIFICACIÓN Y VALIDACIÓN.....	16
4.1.4.	INFORMACIÓN DE VERIFICACIÓN Y VALIDACIÓN.....	18
4.1.5.	DOCUMENTACIÓN DE USUARIO.....	22
5.	ESTANDARES, PRACTICAS Y CONVENCIONES.....	22
5.1.	ESTÁNDAR DE CODIFICACIÓN.....	22
5.2.	ESTÁNDAR DE COMENTARIO.....	24
5.3.	RESPONSABLES DE VERIFICAR EL CUMPLIMIENTO.....	24
6.	REVISIONES Y AUDITORIAS.....	24
7.	GESTION DE CONFIGURACION.....	27
8.	INFORMACION DE PROBLEMAS Y ACCIONES CORRELATIVAS.....	27
10.	HERRAMIENTAS, TECINAS Y METODOLOGIAS.....	29
11.	CONTROL DEL CODIGO.....	30
11.1.	CONTROL DE MEDIOS.....	30
11.2.	CONTROL DE SUMINISTRADORES Y SUBCONTRATOS.....	31
12.	RECOLECCION, MANTENIMIENTO Y RETENCION DE REGISTROS.....	31
13.	REFERENCIAS.....	32



1. INTRODUCCION

Una de las principales fases dentro de la elaboración de un proyecto es el Aseguramiento de la Calidad del Software (SQA), es decir, un modelo sistemático y planeado de todas las acciones necesarias para proveer la confianza adecuada, según los requerimientos técnicos establecidos, de cada producto e ítem del proyecto. Un sinónimo del aseguramiento de la calidad del software es aseguramiento del producto de software.

El plan de aseguramiento de la calidad del software (SQAP) define cuan adherido a estos estándares se debe monitorear. El SQAP contiene una lista de comprobación para las actividades que se deben llevar a cabo para asegurar la calidad del producto. Para cada actividad, en las que tiene responsabilidad el SQA, se debe crear un plan para su monitoreo.

En este documento se describen todos los planes y roles que tendrá cada elemento de la organización en el proceso de aseguramiento de la calidad del software.

1.1. ANTECEDENTES

Todo producto de software busca satisfacer a cabalidad el propósito para el cual fue creado, estar en sintonía con el ambiente donde se va a aplicar, cumplir con los requerimientos que se le imponen, además de ser mantenible, confiable y estable; el logro de estos factores permite decir que se trata de un software de calidad, las diversas características con las que se desea que cumpla un software varían ampliamente.

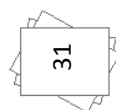
Algunas tienen que ver con el usuario que interactúa con el sistema, otra con el líder del proyecto y diseñadores, y otras que parecen ser muy abstractas y hasta indefinidas. En definitiva toda empresa debe complacer a los clientes e ir más allá para entender a fondo sus necesidades presentes y futuras a fin de sorprenderlos con productos y servicios que ni siquiera imaginen; una vez que son conocidas estas necesidades deben ser traducidas en términos cuantitativos y tangibles, este proceso no es sencillo y requiere de la integración de conocimientos de marketing, ingeniería de software, documentación y auditoría informática, para que las expectativas y las necesidades del consumidor puedan ser satisfechas completamente.

El desarrollo de todos estos lineamientos recae en manos de un proceso denominado Aseguramiento de la Calidad del Software (SQA) el cual se desempeña mediante un plan (plan SQA) que le permitirá estructurarse por medio de actividades que conlleven al éxito de la calidad del proyecto del software, situación en la que radica la importancia de la participación de este plan en el proceso.

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

El desarrollo de un plan de aseguramiento de la calidad de software para la empresa.



1.2.2. OBJETIVOS ESPECIFICOS

Como empresa, se tienen los siguientes objetivos:

- Definir los requerimientos de calidad a ser verificados.
- Indicar los roles y responsabilidades de cada integrante del equipo y director del plan SQA
- Indicar las partes del ciclo vida cubierto por el plan SQA así como las líneas de trabajo que serán contempladas en el mismo.
- Describir las tareas de calidad a realizar.
- Especificar los documentos involucrados en el desarrollo del plan.

1.3. MISION

- Ser una empresa de desarrollo de software de excelente calidad, basada en la utilización de estándares de desarrollo y planificación de software, herramientas de última tecnología informática.
- Convertirnos en los socios estratégicos preferidos en tecnologías de la información de nuestros clientes, apuntando a los sectores más pujantes del país.

1.4. VISION

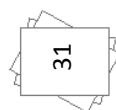
Es ser una empresa de reconocido prestigio nacional e internacional, con autonomía administrativa, con excelencia en el diseño, fabricación y entrega productos de software de la más alta calidad y en donde el mejoramiento continuo en todas las áreas sean de agrado a nuestros consumidores, de eficiente gestión, competitiva, comprometida con el servicio al cliente, la formación integral de su recurso humano y tecnológico.

1.5. POLITICAS DE CALIDAD

TERASOFT, como empresa de ingeniería de software, 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, comprensión y cumplimiento, fijando los siguientes lineamientos básicos:

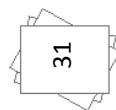
- Suministrar productos tendientes al cero defecto.
- Cumplir con las fechas de entrega pactadas con los clientes.
- Mejorar los procesos y el sistema de gestión de calidad en forma continua.
- Capacitar a todo su personal en base a sus necesidades y a las nuevas tecnologías, incentivando su integración.



- 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

“La calidad del producto refleja la calidad de la empresa”.



2. PLAN DE ASEGURAMIENTO DE CALIDAD DE SOFTWARE

2.1. PROPOSITO

El propósito de este plan es especificar las actividades que se realizarán para asegurar la calidad del software a construir. En él se detalla el producto que se va a revisar y los estándares, normas o métodos a aplicar, los métodos y procedimientos que se utilizarán para revisar que la elaboración del producto se realice como lo establece el modelo de ciclo de vida del proyecto, y procedimientos para informar a los responsables del producto los defectos encontrados y realizar un seguimiento de dichos defectos hasta su corrección.

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

Calidad del software es el cumplimiento con los requisitos explícitamente establecidos y documentados, la concordancia con los estándares de desarrollo aplicados y la agregación de requisitos implícitos que se espera de todo producto hecho por profesionales.

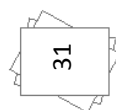
A través de la implantación del SQAP se pretende cumplir con los elementos de calidad de software, los cuales son:

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

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

2.1.3. ALCANCE

Este documento presenta enfoques a seguir de manera de asegurar al cliente la calidad deseada. El alcance de este plan cubre todas las actividades involucradas en el proceso de desarrollo dejando de lado la etapa de mantenimiento, la cual no será parte del mismo. La meta del plan de aseguramiento de la calidad es verificar que todo software y documentación liberados cumplan con todos los requerimientos técnicos establecidos.



El SQAP cubre las fases del ciclo de vida de desarrollo de software. Ciclo de Vida de Desarrollo de Software XP (Programación Extrema)

FASES DE XP:

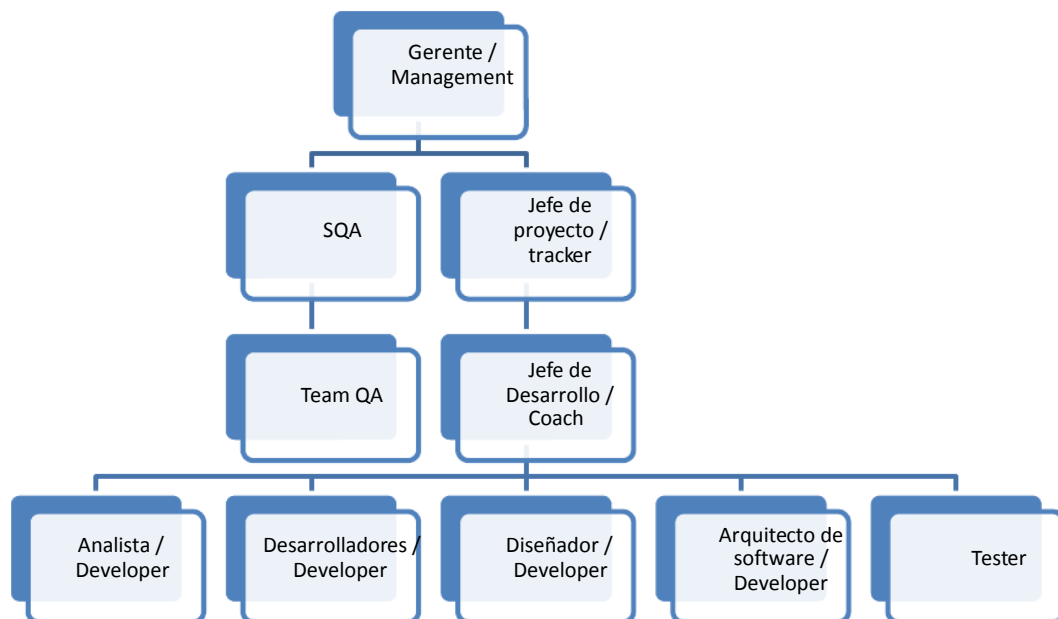
- [1ª Fase: Planificación del proyecto.](#)
- [2ª Fase: Diseño.](#)
- [3ª Fase: Codificación.](#)
- [4ª Fase: Pruebas.](#)

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.

3. GESTION

Seguidamente se especifican los elementos que tienen influencia sobre la calidad del software, como está conformada la línea de gestión de calidad, de quien es la autoridad y responsabilidad por la calidad del software.

3.1. ORGANIZACIÓN



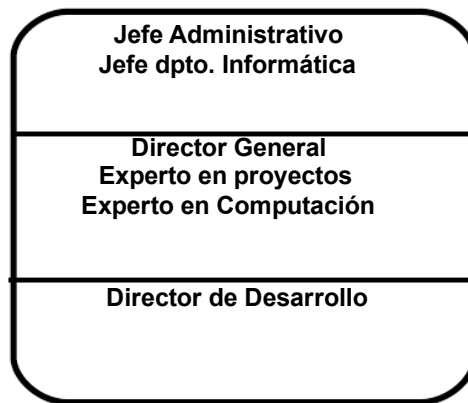
3.1.1.1. ORGANIZACIÓN DE LA SQA

Es la organización que se obtiene de las anteriores organizaciones. En la que se toman decisiones según las normas establecidas para luego liberar versiones sucesivas del SQAP para el desarrollo del software.

Tenemos la lista de la organización de SQA:

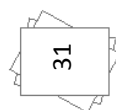
- Gerente Administrativo
- Jefe de Departamento de Informática
- Director General
- Experto en Proyectos
- Director de equipo de Desarrollo

3.2. ORGANIGRAMA DE SQA

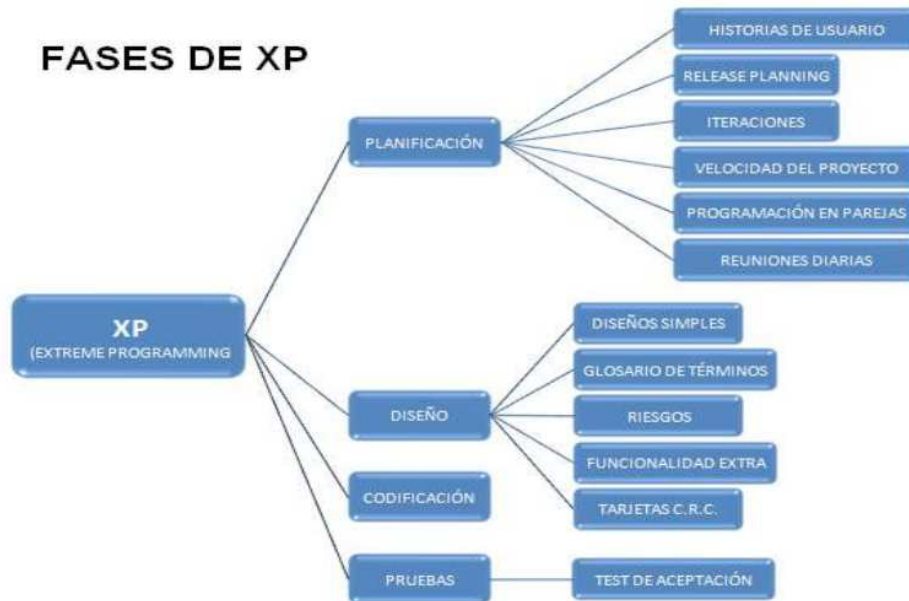


3.3. TAREAS

La relación de tareas asociadas con el ciclo de vida de desarrollo de software (se encuentra basadas en las 4 fases de XP) y las actividades de la SQA son las siguientes, las cuales se ejecutarán durante el desarrollo del producto de software:



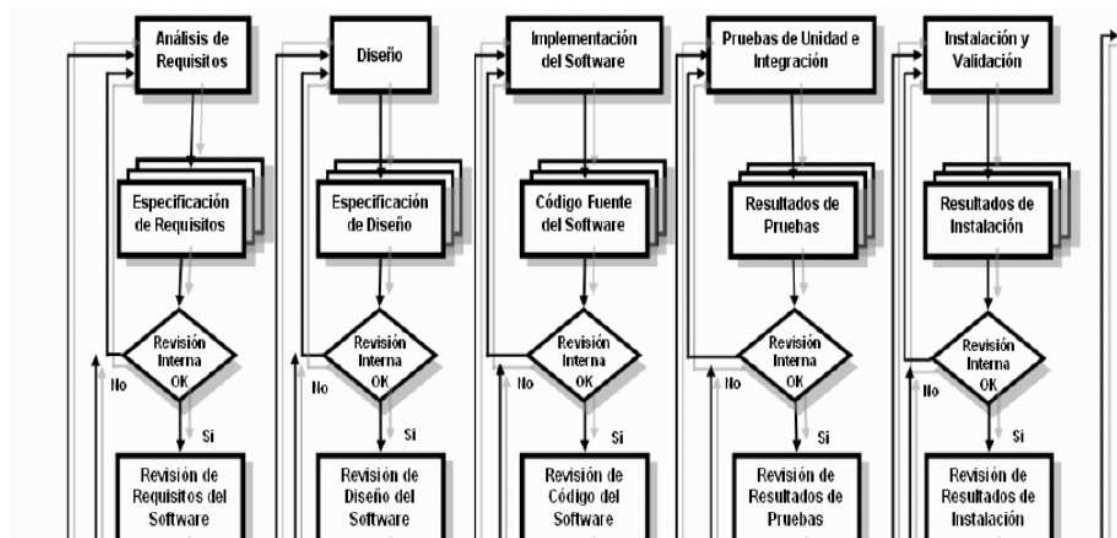
FASES DE XP

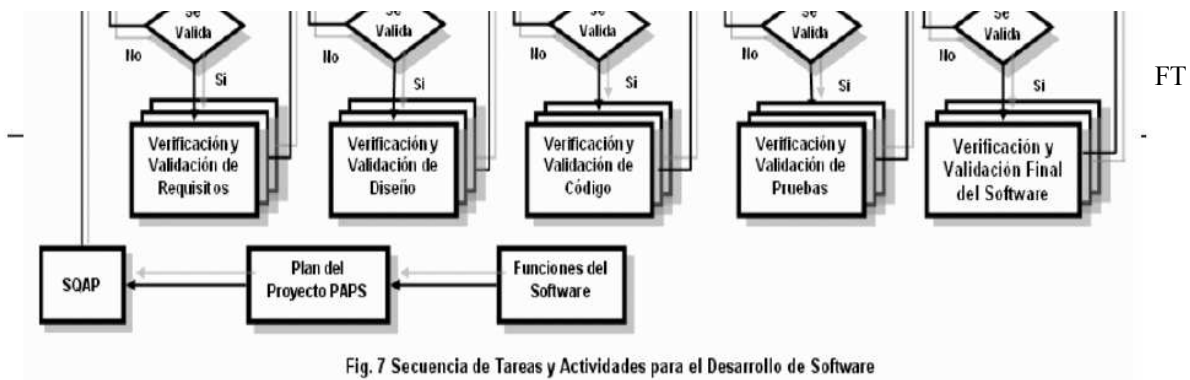


Las actividades de SQA definidas en el modelo de proceso son:

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

Tabla 1 Actividades de Grupo SQA





3.4. RESPONSABILIDADES

Las responsabilidades de cada una de las organizaciones involucradas se definen de la siguiente forma:

3.4.1. RESPONSABILIDADES DEL GRUPO DE DESARROLLO

Esta organización es responsable de:

- Desarrollar un producto de software en base a lo definido 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).

3.4.2. 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.

3.4.3. RESPONSABILIDADES DE LA SQA

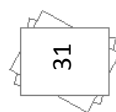
Actividades del grupo SQA:

- | | |
|---|--|
| • | 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. |

Como parte de las actividades del Responsable de SQA se revisarán los productos que se consideren relevantes para la calidad del producto y del proceso. A continuación se identifican esos productos y el responsable de las acciones correctivas para eliminar los defectos de cada producto.



Producto	Rol Responsable	Responsable
Documento de Requerimientos		
Modelo de Casos de Uso		
Alcance del Sistema		
Descripción de la Arquitectura		
Modelo de Diseño		
Modelo de Datos		
Estándar de Implementación		
Estándar de documentación técnica		
Documento de Estimaciones		
Documento de Riesgos		
Plan del Proyecto		
Plan de Verificación y Validación		
Reporte de pruebas unitarias, de integración y del Sistema		
Plan de Implantación		
Estándar de Documentación de Usuario		
Documentación de Usuario		
Plan de Gestión de Configuración		

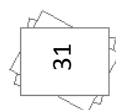
3.5. EVOLUCIÓN DEL SQAP

La organización responsable del SQAP, es la organización SQA. El SQAP, será dado a conocer a todas las organizaciones, cualquier modificación durante la implementación del SQAP debido a circunstancias o necesidades, se ejecutará a del siguiente Plan, para la implementación e evolución del SQAP.

3.5.1. Control De Cambio

El control de cambio del SQAP, será realizado utilizando versiones evolutivas así por ejemplo al presente SQAP, se denomina como la versión 1.0, los siguientes puntos describen el criterio a seguir para la asignación de versión a las modificaciones del SQAP.

1. Si los cambios solo afectan a secciones específicas del SQAP, y las mismas no modifican la línea base del SQAP, la versión asignada será la suma de más 0.1, a la versión actual, los cambios se realizan insertando y/o eliminando hojas en el SQAP vigente.
2. Si los cambios son significativos la versión asignada será la suma de cualquier valor entre [0.1, 0.2, ..., 1.1, 1.2, ..., 1.9, 2, 2.1] para alcanzar el próximo número entero, a la versión actual del SQAP y se imprimirá completamente el nuevo SQAP.



3. Si se quiere con frecuencia el punto (1), y no se imprimió completamente el SQAP modificado ya varias veces, entonces para contar con un SQAP consistente y mantenible se debe imprimir completamente el plan, con todas las modificaciones realizadas previamente, para la asignación de versión se utilizará el criterio descrito en el punto 2.

3.5.2. CUANDO HACER MODIFICACIONES

Las modificaciones al SQAP serán realizadas cuando se presenten las siguientes situaciones:

- En las relaciones y evaluaciones establecidas en este Plan.
- Otras situaciones pueden ser:
- Surgen nuevos cambios necesarios en los requerimientos.
- Se detectan problemas que modifican la línea base del SQAP, en alguna de las fases de la Implementación del SQAP.
- Incumplimiento con alguna de las secciones establecidas en el SQAP.
- Por observación de cualquiera de las organizaciones.

3.5.3. IMPLANTACIÓN DE MODIFICACIONES

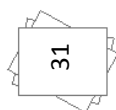
Las modificaciones al SQAP, se harán en función al problema presentado de la siguiente manera:

- Detectar uno de los puntos especificados.
- Realizar un análisis del problema.
- Realizar reuniones técnicas con las áreas involucradas en la modificación del SQAP, estas áreas además de exponer problemas, darán recomendaciones para la modificación correspondiente.
- Hacer la modificación.
- Implementar la modificación correspondiente.
- Una vez finalizada la implementación se evaluará si fue correcta la modificación.
- Promulgar la nueva versión del SQAP.

La modificación al SQAP, solo se realiza en las secciones del SQAP que sean afectadas por la observación, este trabajo estará a cargo de la organización de la SQA.

3.5.4. PUBLICACIÓN DEL SQAP

La publicación de la versión 1.0 del SQAP, se lo hará solo a las áreas afectadas, las cuales recibirán una copia de la nueva versión del SQAP, este trabajo se lo realizará en coordinación entre el



consultor y el jefe de área. Si se presenta el control de cambio (3), se imprimirá todo el contenido del SQAP, actualizado y publicado a través de los representantes de las organizaciones.

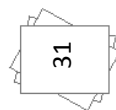
4. DOCUMENTACION

Esta documentación es elaborada por el desarrollador, y se basa en el estándar ANSI / IEEE – Std 830 “Guía par especificaciones de requerimientos de software” (ERS).

La ERS deberá describir claramente y de forma precisa cada uno de los requerimientos del software, tal como: funciones, rendimiento, restricciones de diseño y atributos.

4.1.1. MODELO A USAR PARA EL CONTENIDO DEL ERS

1. INTRODUCCION
 - 1.1.1. Objetivo
 - 1.1.2. Alcance
 - 1.1.3. Definiciones, acrónimos y abreviaciones
 - 1.1.4. Referencias
 - 1.1.5. Revisión
2. 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
3. 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

4.1.2. DESCRIPCIÓN DEL DISEÑO DEL SOFTWARE (DDS)

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”

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

Tabla de Organización de la DDS dentro de Vistas de Diseño

MODELO A USAR PARA EL CONTENIDO DEL DDS

1. INTRODUCCION
 - 1.1. Objetivo
 - 1.2. Alcance
 - 1.3. Definiciones, acrónimos y abreviaciones
2. REFERENCIAS
3. DESCRIPCION DE DESCOMPOSICION
 - 3.1. Descomposición de módulo
 - 3.1.1. Descripción del módulo 1
 - 3.1.n. Descripción del módulo n

-
- 3.2. Descomposición de procesos concurrentes
 - 3.2.1. Descripción del proceso 1
 - 3.2.n. Descripción del proceso n
 - 3.3. Descomposición de datos
 - 3.3.1. Descripción de la entidad de datos 1
 - 3.3.n. Descripción de la entidad de datos n
 - 4. DESCRIPCION DE DEPENDENCIA
 - 4.1. Dependencia entre módulos
 - 4.2. Dependencia entre procesos
 - 4.3. Dependencia entre datos
 - 5. DESCRIPCION DE INTERFACES
 - 5.1. Interfaces de módulo
 - 5.1.1. Descripción del módulo 1
 - 5.1.n. Descripción del módulo n
 - 5.2 Interfaces de procesos
 - 5.2.1. Descripción del proceso 1
 - 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.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

4.1.3. PLAN DE VERIFICACIÓN Y VALIDACIÓN

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

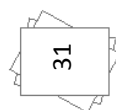
MODELO A USAR PARA EL CONTENIDO DEL V & V

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

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.

4.1.3.1. CICLO DE VIDA DE VERIFICACIÓN Y VALIDACIÓN

El plan se basa en el siguiente ciclo de vida del software:



➤	Fase de concepto
➤	Fase de requerimientos
➤	Fase de diseño
➤	Fase de implementación
➤	Fase de prueba
➤	Fase de instalación y prueba
➤	Fase de operación y mantenimiento

1 Fase de Concepto V&V

Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas
<p>Evaluación de concepto de documentación</p> <p>Evaluar el concepto de documentación para determinar si el concepto propuesto satisface las necesidades del usuario y el objetivo del proyecto. Identificar las restricciones principales de interfaces del sistema y limitaciones de objetivos propuestos. Fijar credibilidad de cada elemento de software</p>	<p>Concepto de documentación:</p> <p>Planificación del proyecto, Memo de iniciación del proyecto</p>	<p>Reporte de tareas</p> <p>Reporte de anomalías</p>

Tabla de Tareas, entradas y salidas de V&V (Fase 1)

2 Fase de Diseño V&V

Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas
<p>Análisis de seguimiento a los requerimientos de software.</p> <p>Seguir los requerimientos ERS hacia los requerimientos del sistema dentro del concepto de documentación. Analizar identificando relaciones para correcciones, consistencia, completitud y optimización.</p>	<p>Concepto de documentación ERS.</p> <p>Documentación de requerimientos de interfaces.</p>	<p>Reporte de tareas.</p> <p>Reporte de anomalías.</p>
<p>Evaluación de requerimientos de software.</p> <p>Evaluar los requerimientos ERS para: precisión, cumplimiento, confiabilidad, y capacidad de ser probado. Fijar la credibilidad de requerimientos para identificar rendimiento o área críticas de software.</p>	<p>Concepto de documentación ERS.</p> <p>Documentación de requerimientos de interfaces.</p>	<p>Reporte de tareas.</p> <p>Reporte de anomalías.</p>
	<p>Concepto de documentación ERS.</p> <p>Documentación de requerimientos de interfaces.</p>	<p>Reporte de tareas.</p> <p>Reporte de anomalías.</p>
	<p>Concepto de documentación ERS.</p> <p>Documentación de requerimientos de interfaces.</p> <p>Documentación de usuario.</p>	<p>Plan de pruebas del sistema.</p> <p>Reporte de anomalías.</p>
	<p>Concepto de documentación ERS.</p> <p>Documentación de</p>	<p>Plan de pruebas de aceptación.</p> <p>Reporte de anomalías.</p>

	requerimientos de interfaces. Documentación de usuario.	
Tabla de Tareas, entradas y salidas de V&V (Fase 2)		

3 Fase de Implementación V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas
Tabla de Tareas, entradas y salidas de V&V (Fase 3)		

4 Fase de Pruebas V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas
Tabla de Tareas, entradas y salidas de V&V (Fase 4)		

5 Fase de Concepto V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas
Tabla de Tareas, entradas y salidas de V&V (Fase 5)		

6 Fase de Instalación y Validación V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas
Tabla de Tareas, entradas y salidas de V&V (Fase 6)		

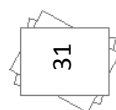
7 Fase de Operación y Mantenimiento V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas
Tabla de Tareas, entradas y salidas de V&V (Fase 7)		

4.1.4. INFORMACIÓN DE VERIFICACIÓN Y VALIDACIÓN

El formato para la documentación de los resultados de la implementación del plan de verificación y validación del software (V & V) es la siguiente:

4.1.4.1. REPORTE SUMARIO DE FASE V&V

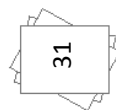
REPORTE SUMARIO DE FASE V&V	Pág.
-----------------------------	------



FASE:		*****
# De Reporte:	Fecha: / /	
Lugar:	Hora:	
a) Descripción de las tareas de V&V realizadas:		
b) Sumario de resultados de tareas:		
c) Sumario de anomalías y resolución:		
d) Evaluación de calidad del software:		
e) Recomendaciones:		
Equipo de Trabajo:		
Nombre:	Firma	

4.1.4.2. REPORTE DE ANOMALÍAS

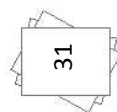
REPORTE DE ANOMALIAS		Pág.
FASE:		*****
# De Reporte:	Fecha: / /	
Lugar:	Hora:	



<p>a) Descripción y ubicación:</p> <p>b) Impacto:</p> <p>c) Causa:</p> <p>d) Critibilidad:</p> <p>e) Recomendaciones:</p>	
Equipo de Trabajo:	Firma
Nombre:	

4.1.4.3. REPORTE FINAL V&V

REPORTES DE V&V		Pág.
FASE:		*****
# De Reporte:	Fecha: / /	

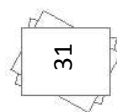


Lugar:	Hora:
<p>a) Resumen de todas las tareas V&V, durante el ciclo de vida del software:</p> <p>b) Resumen de resultados de tareas:</p> <p>c) Resumen de anomalías, resolución:</p> <p>d) Evaluación total de la calidad del software:</p> <p>e) Recomendaciones:</p>	
Equipo de Trabajo:	Firma
Nombre:	

4.1.5. DOCUMENTACIÓN DE USUARIO

Esta descripción de documentación de usuario (UD) se basa en el estándar ANSI / IEEE – Std 1036 “STANDARD FOR SOFTWARE USER DOCUMENTATION”.

La información especificada debe ser incluida en la documentación del usuario, esta documentación de usuario comprenderá de un conjunto. En cada documento se debe tomar en cuenta y describir los siguientes puntos.



Los documentos de usuario serán presentados en dos modos: instruccional y de referencia. Los usuarios del software utilizarán los documentos ya sea para aprender acerca del software (modo instruccional) o para refrescar su memoria acerca del software (modo de referencia).

Modo Instruccional

Un modo instruccional de documento debe:

- Prover el ambiente y la información necesaria para entender el sistema.
- Prover la información necesaria para aprender lo que puede hacer con el software y como lo puede usar.
- Prover ejemplos para reforzar el proceso de aprendizaje.

Modo de Referencia

Un documento de modo de referencia debe:

- Organizar y proveer información necesaria.
- Facilitar accesos aleatorios a la información.

Los documentos de modo de referencia que debe ser incluido son:

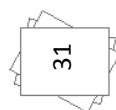
- a. Manual de comandos.
- b. Manual de mensajes de error.
- c. Manual de llamadas de programas.
- d. Guía de referencia rápida.
- e. Manual de Herramientas del software.
- f. Manual de utilitarios.

5. ESTANDARES, PRACTICAS Y CONVENCIONES

5.1. ESTÁNDAR DE CODIFICACIÓN

La codificación del software se realizará en la plataforma Java. 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.
 - Para asignar nombres a las variables debe de realizarse de la siguiente forma:
 - xNombreCliente
 - xSaldo
 - xFechaInicio

Donde x = indica el tipo de dato, puede representar: enteros, reales, cadenas, etc.
 - 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

- Se utilizarán las Convenciones de Codificación en Java de Sun Microsystems, Inc.

5.2. ESTÁNDAR DE COMENTARIO

- 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.

Estándares para documentación

Entregable	Estándar

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.

6. REVISIONES Y AUDITORIAS

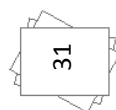
Los responsables de estas revisiones es la organización del SQA, con la participación de todo elemento de la organización que tengan que ver con los requerimientos, tales como: los diseñadores del software, agentes de pruebas.

6.1. Propósito

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.

6.2. Requisitos Mínimos



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

Revisión de los Requisitos de Software (RRS):

La RRS se genera para:

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

Los requerimientos de revisiones de ERS en la RRS son los siguientes:

- a. Fiable
- b. Completo
- c. Depurable
- d. Modificable
- e. Consistente
- f. Libre de ambigüedades
- g. Utilizable durante la fase de operación y mantenimiento.
- h. 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 DDS preliminar, antes de comenzar con el diseño detallado, define los siguientes puntos:

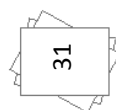
- Evaluar el progreso, consistencia y suficiencia técnica del alcance de diseño con los requerimientos funcionales de la ERS.
- Verificar la existencia y compatibilidad de las interfaces entre el software, el hardware y los usuarios finales.
- Determinar un diseño de software que cumpla con los requerimientos.

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 DDS cumple con la ERS. 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 ERS.
- 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 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.

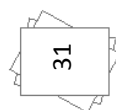
Se debe convocar a la reunión formalmente a los involucrados, informar del material que ellos deben preparar por adelantado, llevar una lista de preguntas y dudas que surgen del estudio del producto a ser revisado.

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

6.3. 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
.....			

Iteración N

Fase II – Elaboración

Iteración I

.....

Iteración N

Fase III – Construcción

Iteración I

.....

Iteración N

Fase IV – Transición

Iteración I

.....

Iteración N

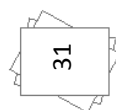
Y así sucesivamente para cada una de las fases del ciclo de vida de desarrollo de software.

7. GESTION DE CONFIGURACION

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.

8. INFORMACION 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.

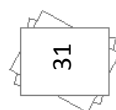
El propósito de un sistema de Gestión de Problemas y Acciones Correlativas es:

- 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.

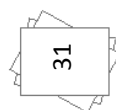
REPORTE FINAL DE PROBLEMAS	Pág.....
# De Reporte: _____ Lugar: _____	Fecha: / /... Hora:
a) Identificación del problema: _____	



b) Descripción: 	
c) El evento ejecutado cuando se presentó el problema es: 	
d) 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.
 - Registrar las causas sospechosas del origen del problema, esto asegura que no se descarta ninguna situación posible.
 - Registrar causas adicionales
 - Registrar causas plenamente identificadas, esto es, causas verificadas por detectores del problema
 - 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.

9. TESTEO

Revisión del Plan de Verificación y Validación (RP V & V):

La RP V & V es generado para la evaluación de:

- definidos en el V & V. Los métodos de Verificación y Validación
- software con el V & V. El cumplimiento durante el desarrollo del

10. HERRAMIENTAS, TECNICAS Y METODOLOGIAS

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.

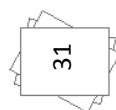
Algunas de las herramientas son:

- Utilidades del sistema operativo WINDOWS XP.
- Utilidades del sistema operativo LINUX.
- Documentación de ayuda.
- Instaladores.

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

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

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



11. CONTROL DEL 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:

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

11.1. 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.

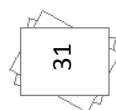
Se debería asegurar que:

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

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

11.2. CONTROL DE SUMINISTRADORES Y SUBCONTRATOS

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

12. RECOLECCION, 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.

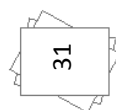
Se identifica aquella documentación que se debe retener, y se especifican los métodos y facilidades que se utilizarán para recolectar, proteger y mantener esta documentación.

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 repetibilidad 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.

13. REFERENCIAS

[1].-

[RJB-1999] James Rumbaugh, Ivar Jacobson, Grady Booch, "El proceso Unificado de Desarrollo de Software", Editorial ADISSON WESLEY, 1999, Ciudad de Madrid España.

[2]

[PRESSMAN-2002] Roger S. Pressman, "Ingeniería de Software. Un Enfoque Práctico Quinta Edición", Editorial McGRAW-HIL, 2002, España

[3]

[MOORE-1998] James W. Moore, "Software Engineering Standards, Cap 6, IEEE Computer Society", 1998.

[4]

[FFMD-1998] Finkelstein A., Fuggetta A., Montangero C., Derniame J.C., "Software Process: Principals, Methodology and Technology, Cap 2, Springer-Verlag", 1998.

[5]

[WWW-01] "Laboratorio de Sistemas de Información Facultad de Informática Universidad Politécnica de Valencia"

<http://campus.fortunecity.com/defiant/114/index.html>

[6]

[Dirección de tecnología y sistemas](#)

[<HTTP://WWW.SISTEMAS.EDU.BO/JBERMUDEZ/SIS3502A/ESTANDARESIEEE.PPT>](http://WWW.SISTEMAS.EDU.BO/JBERMUDEZ/SIS3502A/ESTANDARESIEEE.PPT)

