

**UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO FACULTAD DE
INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES**

CARRERA: INGENIERÍA EN SISTEMAS



**SOFTWARE DE MONITOREO EMOCIONAL PARA EL BIENESTAR
ESTUDIANTIL BASADO EN EL ANÁLISIS DE SENTIMIENTO
“EMOCARE”**

PLAN DE ASEGURAMIENTO DE CALIDAD DE SOFTWARE (SQAP)

Autores:

Univ. Cruz Vargas Joan Paul 219015937

Univ. Meneses Choque Karla Patricia 218073135

Materia: Ingeniería De Software II - INF512 SB

Tutor: Ing. Martínez Canedo Rolando Antonio

Santa Cruz de la Sierra - Bolivia

Noviembre – 2023

Índice

I.	Manual De Calidad / SQAP	7
1.1	Introducción	7
1.1.1	Antecedentes	7
1.1.2	Objetivos	8
1.1.2.1	Objetivo General	8
1.1.2.2	Objetivos Específicos.....	8
1.1.3	Misión	9
1.1.4	Visión	9
1.1.5	Políticas De Calidad.....	9
1.1.5.1	Principios de Calidad:	9
1.1.5.1.1	Calidad del Software:.....	10
1.1.5.1.2	Innovación y Colaboración:.....	10
1.1.5.1.3	Satisfacción del Cliente:	10
1.1.5.1.4	Mejora Continua:	10
1.1.5.1.5	Desarrollo Profesional:	10
1.1.5.2	Compromisos Clave:.....	11
1.1.6	Slogan.....	11
1.2	Plan De Aseguramiento De Calidad De Software (SQAP).	11
1.2.1	Propósito	11
1.2.1.1	Objetivo.....	11
1.2.1.2	Descripción	12
1.2.1.3	Alcance.....	13

1.2.1.3.1	Planificación de Calidad	14
1.2.1.3.2	Requisitos.....	14
1.2.1.3.3	Diseño	14
1.2.1.3.4	Implementación.....	14
1.2.1.3.5	Pruebas	14
1.2.1.3.6	Despliegue.....	15
1.2.1.3.7	Mantenimiento	15
1.2.1.3.8	Medición y Mejora Continua	15
1.2.1.3.9	Documentación y Revisiones.....	15
1.2.2	Documentos De Referencia.....	15
1.2.3	Gestión	17
1.2.3.1	Organización	17
1.2.3.1.1	Gerente de Proyecto (Project Manager):.....	17
1.2.3.1.2	Equipo de Desarrollo:	17
1.2.3.1.3	Gerente de Calidad del Software (Software Quality Manager):	17
1.2.3.1.4	Usuario Final (Estudiantes de colegio):	18
1.2.3.1.5	Stakeholders Externos (Padres, Profesores, Instituciones Educativas):	18
1.2.3.1.6	Equipo de Soporte Técnico (Technical Support Team):.....	18
1.2.3.2	Tareas	18
1.2.3.2.1	Gerente de Proyecto:	18
1.2.3.2.2	Equipo de Desarrollo:	18
1.2.3.2.3	Gerente de Calidad del Software:	19
1.2.3.2.4	Usuario Final (Estudiantes de colegio):	19

1.2.3.2.5	Stakeholders Externos (Padres, Profesores, Instituciones Educativas):	19
1.2.3.2.6	Equipo de Soporte Técnico:.....	19
1.2.3.3	Roles Y Responsabilidades	20
1.2.3.3.1	Gerente de Proyecto.....	20
1.2.3.3.2	Desarrolladores de Software:	20
1.2.3.3.3	Gerente de Calidad del Software:	20
1.2.3.3.4	Usuarios Finales (Estudiantes de colegio):	21
1.2.3.3.5	Stakeholders Externos (Padres, Profesores, Instituciones Educativas):	21
1.2.3.3.6	Equipo de Soporte Técnico:.....	21
1.2.4	Documentación	21
1.2.4.1	Propósito	21
1.2.4.2	Requisitos Mínimos de Documentación	22
1.2.4.2.1	Especificación De Requisitos De Software (SRS).....	22
1.2.4.2.2	Descripción Del Diseño Del Software (SDD)	23
1.2.4.2.3	Plan de Verificación y Validación de Software (PVVS)	26
1.2.4.2.4	Informe de Verificación y Validación de Software (IVVS)	31
1.2.4.2.5	Documentación del Usuario (UD)	35
1.2.4.2.6	Plan de Gestión de Configuración de Software (SCMP).....	37
1.2.5	Estándares, Practicas y Convenciones	38
1.2.5.1	Estándar de Codificación	38
1.2.5.2	Estándar de comentarios	39
1.2.5.3	Responsables de verificar el cumplimiento.....	39
1.2.6	Revisiones y Auditorias	40

1.2.6.1 Evaluación de la calidad de los productos	42
1.2.6.2 Revisar el ajuste al proceso:	43
1.2.6.3 Revisión Técnica Formal	43
1.2.6.4 Requerimientos Mínimos	44
1.2.7 Gestión de Configuración	45
1.2.8 Gestión de problemas y acciones correlativas	46
1.2.9 Herramientas Técnicas y Metodologías	47
1.2.10 Control de Código	48
1.2.11 Control de Medios	48
1.2.12 Control de suministros y subcontratos	50
1.2.13 Recolección, Mantenimiento y Retención de Registros	50

CAPÍTULO III | MANUAL DE CALIDAD / SQAP

I. Manual De Calidad / SQAP

1.1 Introducción

En NovaTech, la calidad del software es nuestra prioridad. Entendemos que el software impregna cada aspecto de nuestra vida y la calidad es fundamental. Este manual representa nuestra guía hacia estándares de calidad excepcionales y demuestra nuestro compromiso con la excelencia. Adoptamos un enfoque de tres niveles para garantizar la calidad desde la perspectiva del usuario, desarrollador y producto. Nuestra gestión de calidad está estandarizada y certificada para ofrecer soluciones tecnológicas de alto calibre. La calidad es nuestra ventaja competitiva en un mercado en constante evolución. En NovaTech, la calidad no es un costo, sino una inversión esencial para nuestro éxito y el de nuestros clientes.

1.1.1 Antecedentes

NovaTech es una joven y prometedora startup fundado en 2023. La empresa fue establecida por Meneses Choque Karla Patricia y Cruz Vargas Joan Paul, dos estudiantes apasionados de la UAGRM, que comparten una visión común de impulsar la innovación en el mundo del desarrollo de software.

A pesar de nuestra modesta plantilla de dos miembros fundadores, NovaTech está construyendo un sólido historial en el campo de la tecnología. Hasta la fecha, hemos completado con éxito 2 proyectos tecnológicos innovadores que han demostrado nuestro compromiso con la excelencia y la calidad.

La estructura de NovaTech se basa en la colaboración estrecha y la complementación de habilidades. Meneses Karla aporta su experiencia en dirección de proyecto bajo la metodología ágil, base de datos, desarrollo web, mientras que Cruz Joan se especializa en servidores,

backend, desarrollo móvil, IA. Esta combinación de conocimientos y habilidades nos ha permitido abordar una amplia gama de desafíos tecnológicos de manera efectiva.

A pesar de nuestra corta trayectoria, hemos demostrado nuestra capacidad para enfrentar proyectos de desarrollo de software de alta complejidad y hemos ganado la confianza de nuestros clientes y colaboradores. Nuestra dedicación a la calidad, la innovación y la satisfacción del cliente es el pilar fundamental de NovaTech.

1.1.2 Objetivos

1.1.2.1 Objetivo General

El objetivo general de NovaTech es convertirse en un referente en el desarrollo de software de alta calidad y soluciones tecnológicas innovadoras, contribuyendo de manera significativa al progreso tecnológico y a la satisfacción de las necesidades de nuestros clientes y usuarios finales.

1.1.2.2 Objetivos Específicos

Desarrollar software de vanguardia que cumpla con los más altos estándares de calidad y satisfaga las necesidades específicas de nuestros clientes.

Fomentar la colaboración y la innovación dentro de nuestro equipo para impulsar la creación de soluciones tecnológicas únicas y creativas.

Establecer relaciones sólidas y duraderas con nuestros clientes, basadas en la confianza, la transparencia y la entrega puntual de proyectos.

Promover una cultura de calidad y mejora continua en todos los aspectos de nuestras operaciones, desde la planificación hasta la entrega final.

Invertir en el desarrollo profesional de nuestro equipo para potenciar su experiencia y habilidades técnicas.

1.1.3 Misión

En NovaTech, nuestra misión es liderar la industria de desarrollo de software y tecnología a través de soluciones tecnológicas de alta calidad que cumplen rigurosos estándares de calidad. Nos dedicamos a satisfacer necesidades específicas mientras fomentamos la innovación y la colaboración. Establecemos relaciones sólidas y duraderas con nuestros clientes, basadas en la confianza, la transparencia y la entrega puntual de proyectos. Nuestra búsqueda constante de certificaciones que avalen nuestra calidad refuerza nuestro compromiso con la excelencia en cada aspecto de nuestras operaciones.

1.1.4 Visión

La visión de NovaTech es posicionarse como líder global en el desarrollo de software y tecnología, destacando por soluciones tecnológicas innovadoras que cumplen y superan los más altos estándares de calidad. Buscamos diversificar nuestra cartera de proyectos exitosos y expandir nuestra presencia en diversos sectores y mercados, manteniendo una cultura arraigada en la calidad y la mejora continua. Nuestra inversión en el desarrollo profesional del equipo y la obtención de certificaciones reconocidas a nivel internacional nos permiten liderar el camino hacia un futuro tecnológico de excelencia y confiabilidad.

1.1.5 Políticas De Calidad

En NovaTech, nos comprometemos a desarrollar software y soluciones tecnológicas de la más alta calidad, adoptando rigurosos estándares de calidad para guiar nuestros procesos y operaciones. Nuestra política de calidad es el cimiento sobre el cual construimos nuestra excelencia y satisfacemos plenamente las necesidades de nuestros clientes y usuarios finales.

1.1.5.1 Principios de Calidad:

1.1.5.1.1 Calidad del Software:

Nos comprometemos a desarrollar software de vanguardia que cumple con los más altos estándares de calidad. Seguimos las directrices y estándares internacionales reconocidos, como la Norma ISO 9001:2008, para garantizar que nuestros productos sean fiables y eficientes.

1.1.5.1.2 Innovación y Colaboración:

Fomentamos la colaboración y la innovación dentro de nuestro equipo para impulsar la creación de soluciones tecnológicas únicas y creativas que estén en línea con los estándares de calidad establecidos.

1.1.5.1.3 Satisfacción del Cliente:

Establecemos relaciones sólidas y duraderas con nuestros clientes, basadas en la confianza, la transparencia y la entrega puntual de proyectos que cumplen con los estándares de calidad acordados.

1.1.5.1.4 Mejora Continua:

Promovemos una cultura de calidad y mejora continua en todos los aspectos de nuestras operaciones. Monitoreamos y evaluamos regularmente nuestros procesos para asegurarnos de que estén en consonancia con los estándares de calidad y buscamos oportunidades para optimizarlos.

1.1.5.1.5 Desarrollo Profesional:

Invertimos en el desarrollo profesional de nuestro equipo para potenciar su experiencia y habilidades técnicas, garantizando que estén alineadas con los estándares de calidad de la industria.

1.1.5.2 Compromisos Clave:

Cumplir con los estándares de calidad establecidos en todas nuestras actividades y proyectos.

Mantenernos actualizados con las últimas normas y estándares de calidad relevantes para la industria tecnológica.

Continuar con la formación y el desarrollo de nuestro equipo en línea con los estándares de calidad y las mejores prácticas.

Evaluar constantemente la eficacia de nuestros procesos y estándares de calidad, y tomar medidas para la mejora continua.

1.1.6 Slogan

“EmoCare: El puente hacia el bienestar emocional”

1.2 Plan De Aseguramiento De Calidad De Software (SQAP).

1.2.1 Propósito

El SQAP de NovaTech se establece con el propósito fundamental de proporcionar un marco sólido y estructurado para asegurar la calidad en todos los aspectos del proceso de desarrollo de software de nuestra empresa. Este plan tiene como objetivo principal garantizar que nuestros productos de software cumplan consistentemente con los estándares de calidad definidos y superen las expectativas de nuestros valiosos clientes.

1.2.1.1 Objetivo

El objetivo primordial del SQAP de NovaTech es asegurar que nuestros productos de software sean concebidos, desarrollados y entregados cumpliendo con los más altos estándares de calidad en los siguientes aspectos clave:

Corrección y precisión.

Eficiencia en el rendimiento.

Fiabilidad y robustez.

Facilidad de uso y experiencia del usuario.

Mantenibilidad y escalabilidad.

Seguridad e integridad de los datos.

Portabilidad a diversas plataformas.

El SQAP se centra en la mejora continua de nuestros procesos de desarrollo, con el propósito de elevar la satisfacción de nuestros clientes y consolidar una reputación de excelencia en la calidad del software que producimos.

1.2.1.2 Descripción

El SQAP de NovaTech establece un conjunto integral de normas, procedimientos y prácticas que serán rigurosamente seguidos en todas las fases del ciclo de desarrollo de software. Este plan abarca desde la definición de los estándares de calidad hasta la realización de revisiones y pruebas exhaustivas.

Este plan está diseñado para:

Identificar y gestionar de manera proactiva los riesgos de calidad en cada proyecto de desarrollo.

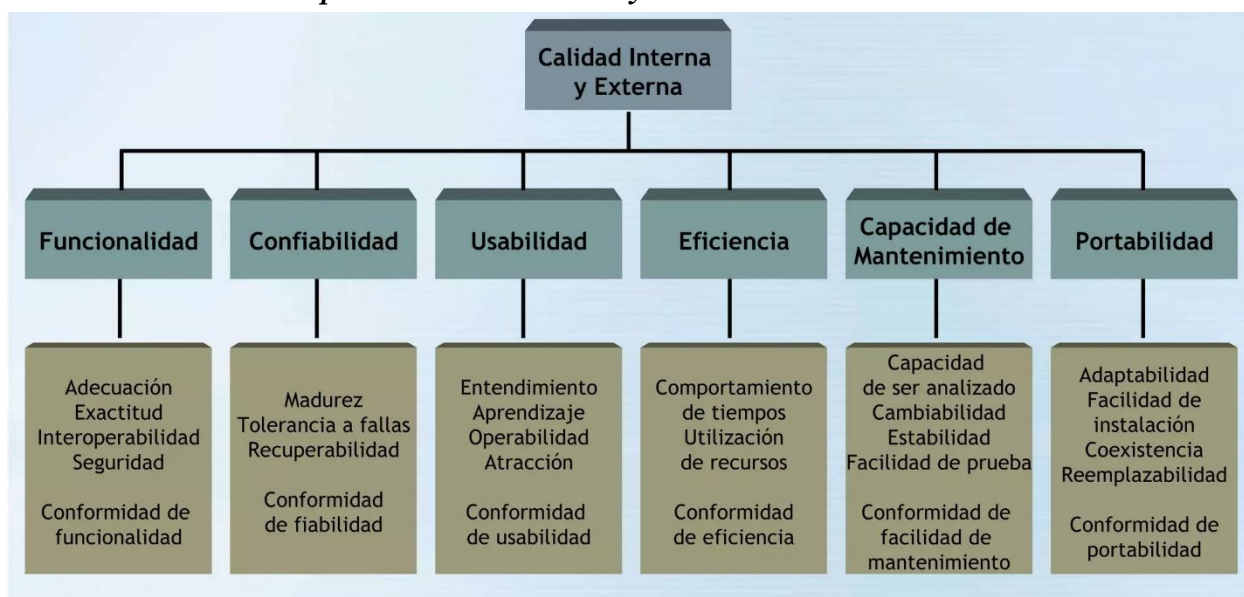
Establecer estándares y directrices claras para el diseño, codificación y pruebas de nuestros productos de software.

Realizar revisiones y auditorías periódicas con el fin de asegurar el cumplimiento de los estándares de calidad establecidos.

Planificar y llevar a cabo pruebas exhaustivas de calidad para detectar y corregir cualquier anomalía o error.

Medir y analizar métricas de calidad con el propósito de impulsar la mejora continua en todos los aspectos de nuestras operaciones.

Modelo de calidad para la calidad interna y externa



El SQAP de NovaTech es de aplicación obligatoria en todos nuestros proyectos de desarrollo de software y se somete a revisiones y actualizaciones periódicas para mantenerse alineado con las últimas tendencias tecnológicas y las mejores prácticas de calidad en la industria.

1.2.1.3 Alcance

El alcance de un Plan de Aseguramiento de Calidad de Software (SQAP) que cubre las fases del ciclo de vida de desarrollo de software es bastante amplio y abarca todas las etapas desde la concepción del proyecto hasta su implementación y mantenimiento. A continuación, se describe el alcance de un SQAP de este tipo de manera más detallada:

1.2.1.3.1 Planificación de Calidad

El SQAP incluye la planificación detallada de cómo se asegurará la calidad del software en todas las fases del proyecto. Esto abarca la definición de objetivos de calidad, la asignación de responsabilidades dentro del equipo de desarrollo y calidad, y la creación de un cronograma que indique cuándo se llevarán a cabo las actividades de aseguramiento de calidad.

1.2.1.3.2 Requisitos

En esta fase, el SQAP establece cómo se verificarán y validarán los requisitos del software. Esto incluye revisiones de requisitos para asegurarse de que son claros y completos, así como pruebas de aceptación del cliente para verificar que los requisitos se cumplen.

1.2.1.3.3 Diseño

El SQAP define cómo se llevarán a cabo las revisiones de diseño para evaluar la arquitectura y la calidad del diseño. También se especifica cómo se verificará que el diseño cumple con los requisitos establecidos.

1.2.1.3.4 Implementación

Durante esta fase, el SQAP aborda la revisión del código, las pruebas unitarias y la gestión de la configuración del software. Se asegura de que el código cumpla con los estándares de calidad definidos.

1.2.1.3.5 Pruebas

El SQAP establece cómo se planificarán, ejecutarán y documentarán las pruebas de calidad en todas las fases del proyecto. Esto incluye pruebas de unidad, pruebas de integración, pruebas de sistema y pruebas de aceptación.

1.2.1.3.6 Despliegue

El SQAP puede definir cómo se gestionarán las actividades de aseguramiento de calidad durante la implementación del software en un entorno de producción. Se asegura de que el software sea confiable y se comporta como se espera en el entorno real.

1.2.1.3.7 Mantenimiento

A lo largo del ciclo de vida del software, el SQAP aborda cómo se gestionarán las actualizaciones, correcciones de errores y cambios. Se garantiza que las modificaciones no comprometan la calidad existente del software.

1.2.1.3.8 Medición y Mejora Continua

El SQAP especifica las métricas de calidad que se utilizarán para evaluar el cumplimiento de los estándares y objetivos de calidad en cada fase. Además, define cómo se recopilarán y analizarán los datos para mejorar continuamente el proceso de desarrollo.

1.2.1.3.9 Documentación y Revisiones

El SQAP establece cómo se documentarán todas las actividades de aseguramiento de calidad y cómo se llevarán a cabo las revisiones y auditorías para garantizar que se sigan los procedimientos definidos.

1.2.2 Documentos De Referencia

IEEE STD 730-1998 y IEEE STD 730.1-1995: Estos estándares y guías establecen los requisitos y las directrices para la creación de planes de aseguramiento de calidad del software. Son fundamentales para planificar y gestionar la calidad del software en todas las etapas del ciclo de vida del proyecto.

ISO/IEC 25000 (SQuaRE - Software Product Quality Requirements and

Evaluation): Esta serie de estándares internacionales se centra en la calidad del producto de software y proporciona un marco completo para evaluar y especificar la calidad del software. Es relevante para medir la calidad de "EmoCare."

ISO/IEC 9126 (Software Engineering - Product Quality): Este estándar define un modelo de calidad del software y proporciona pautas para la evaluación de la calidad del producto de software en términos de características como funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

ISO/IEC 25010 (Systems and Software Quality Models): Esta norma es una revisión de la ISO/IEC 9126 y define un conjunto de modelos de calidad que pueden ser aplicados al software, incluyendo modelos para la calidad interna y externa.

ISO 9001 (Sistemas de Gestión de Calidad): Aunque no está específicamente dirigida al desarrollo de software, la norma ISO 9001 proporciona un marco general para establecer un sistema de gestión de calidad en una organización.

ISO/IEC 12207 (Procesos del Ciclo de Vida del Software): Esta norma define procesos de ciclo de vida de software y proporciona una estructura para la gestión y mejora de procesos en el desarrollo de software.

ISO/IEC 15504 (SPICE - Software Process Improvement and Capability

Determination): Este estándar proporciona un marco para la evaluación y mejora de procesos de software. Es relevante para mejorar la eficiencia y calidad de los procesos de desarrollo de software.

ISO/IEC 17025 (Requisitos generales para la competencia de laboratorios de ensayo y calibración): Este estándar es relevante para garantizar la competencia y calidad de las pruebas.

1.2.3 Gestión

La gestión de calidad del software en nuestro proyecto "EmoCare" se rige por una estructura organizativa sólida y definida, que incluye roles, funciones y responsabilidades claramente establecidos. Nuestra prioridad es garantizar que el software cumpla con los estándares de calidad más altos y satisfaga las necesidades de nuestros usuarios.

1.2.3.1 Organización

En nuestra organización, hemos establecido una estructura de gestión de calidad que garantiza la supervisión y el control efectivo de todos los aspectos relacionados con la calidad del software.

1.2.3.1.1 Gerente de Proyecto (Project Manager):

Encargado de liderar y supervisar todo el proyecto "EmoCare", desde la planificación hasta la implementación, y asegurarse de que se cumplan los plazos y presupuestos.

1.2.3.1.2 Equipo de Desarrollo:

Ingenieros de software, diseñadores de interfaz de usuario (UI/UX), analistas de datos y desarrolladores de infraestructura que trabajan en el diseño, desarrollo, pruebas y mantenimiento de la aplicación "EmoCare".

1.2.3.1.3 Gerente de Calidad del Software (Software Quality Manager):

Responsable de garantizar que el software cumpla con los estándares de calidad establecidos, realizando auditorías y revisiones de calidad.

1.2.3.1.4 *Usuario Final (Estudiantes de colegio):*

Los estudiantes que utilizan la aplicación "EmoCare" para monitorear y gestionar sus emociones, proporcionando datos y retroalimentación.

1.2.3.1.5 *Stakeholders Externos (Padres, Profesores, Instituciones Educativas):*

Personas y organizaciones que tienen interés en el éxito de "EmoCare" y pueden proporcionar apoyo, orientación y retroalimentación.

1.2.3.1.6 *Equipo de Soporte Técnico (Technical Support Team):*

Brinda asistencia técnica a los usuarios en caso de problemas técnicos y mantiene la infraestructura técnica de la aplicación.

1.2.3.2 Tareas

1.2.3.2.1 *Gerente de Proyecto:*

- Planificación y programación del proyecto.
- Gestión del presupuesto y recursos.
- Comunicación con los stakeholders.
- Supervisión y dirección del equipo de desarrollo.
- Gestión de riesgos y resolución de problemas.

1.2.3.2.2 *Equipo de Desarrollo:*

- Diseño y desarrollo de la aplicación "EmoCare".
- Pruebas de software y corrección de errores.
- Recopilación y análisis de datos de usuario.
- Mantenimiento continuo y mejoras en el software.

- Colaboración en la documentación técnica.

1.2.3.2.3 *Gerente de Calidad del Software:*

- Establecimiento de estándares de calidad.
- Realización de auditorías y revisiones de calidad.
- Identificación y gestión de problemas de calidad.
- Garantizar que se sigan las mejores prácticas de desarrollo de software.
- Implementación de métricas de calidad.

1.2.3.2.4 *Usuario Final (Estudiantes de colegio):*

- Uso de la aplicación "EmoCare" para el monitoreo de emociones.
- Proporcionar datos y retroalimentación sobre la aplicación.
- Comunicación de problemas o necesidades específicas.
- Participación en las pruebas de usabilidad.

1.2.3.2.5 *Stakeholders Externos (Padres, Profesores, Instituciones*

Educativas):

Proporcionar retroalimentación sobre la efectividad de "EmoCare".

Apoyar la implementación de la aplicación en entornos educativos.

Comunicar necesidades específicas y expectativas.

1.2.3.2.6 *Equipo de Soporte Técnico:*

- Brindar asistencia técnica a los usuarios.
- Resolver problemas técnicos y responder preguntas.
- Actualizar y mantener la infraestructura técnica.

1.2.3.3 Roles Y Responsabilidades

1.2.3.3.1 Gerente de Proyecto

Responsabilidades:

- Planificar y gestionar el proyecto.
- Supervisar el presupuesto y recursos.
- Comunicarse con los stakeholders.
- Dirigir el equipo de desarrollo.
- Gestionar riesgos y resolver problemas.

1.2.3.3.2 Desarrolladores de Software:

Responsabilidades:

- Diseñar y desarrollar la aplicación "EmoCare".
- Realizar pruebas y correcciones.
- Recopilar y analizar datos de usuario.
- Mantener y mejorar la aplicación.
- Contribuir a la documentación técnica.

1.2.3.3.3 Gerente de Calidad del Software:

Responsabilidades:

- Establecer estándares de calidad.
- Realizar auditorías y revisiones de calidad.
- Identificar y gestionar problemas de calidad.
- Asegurar el cumplimiento de las mejores prácticas.
- Implementar métricas de calidad.

1.2.3.3.4 *Usuarios Finales (Estudiantes de colegio):*

Responsabilidades:

- Utilizar la aplicación "EmoCare" para el monitoreo de emociones.
- Proporcionar datos y retroalimentación.
- Comunicar problemas o necesidades específicas.
- Participar en pruebas de usabilidad.

1.2.3.3.5 *Stakeholders Externos (Padres, Profesores, Instituciones Educativas):*

Responsabilidades:

- Proporcionar retroalimentación sobre la efectividad de "EmoCare".
- Apoyar la implementación en entornos educativos.
- Comunicar necesidades y expectativas específicas.

1.2.3.3.6 *Equipo de Soporte Técnico:*

Responsabilidades:

- Brindar asistencia técnica a los usuarios.
- Resolver problemas técnicos y preguntas.
- Actualizar y mantener la infraestructura técnica.

1.2.4 *Documentación*

1.2.4.1 *Propósito*

El propósito de la documentación en el Plan de Aseguramiento de Calidad de Software (SQAP) de NovaTech es proporcionar una guía completa que abarque todas las etapas del ciclo de vida de desarrollo de software. Estos documentos tienen el fin de asegurar que nuestros

productos de software cumplan con los requisitos definidos y los estándares de calidad establecidos. Además, establecen los criterios y procesos para evaluar la adecuación de cada documento mediante revisiones y auditorías.

1.2.4.2 Requisitos Mínimos de Documentación

Para asegurar que la implementación del software cumple con los requisitos, se requiere la siguiente documentación como mínimo:

1.2.4.2.1 Especificación De Requisitos De Software (SRS)

La Especificación de Requisitos de Software (SRS) tiene como objetivo principal definir de manera clara y precisa todos los requisitos esenciales del software "EmoCare". Cada uno de estos requisitos, que incluyen funciones, rendimiento, restricciones de diseño y atributos, se describe de manera que sea verificable y validable mediante un método predefinido, que puede incluir inspección, análisis, demostración o pruebas. Este proceso se realiza de acuerdo con el estándar ANSI/IEEE Std 830-1984, titulado "Guía para especificaciones de requerimientos de software".

La SRS actúa como el fundamento sobre el cual se basarán las fases posteriores del desarrollo de "EmoCare", garantizando que el software cumpla con los requerimientos establecidos por los usuarios y otros interesados en el proyecto. Cada requisito se documenta de manera rigurosa para facilitar su seguimiento y confirmación durante el proceso de desarrollo y pruebas.

MODELO A USAR PARA EL CONTENIDO DEL SRS

1. INTRODUCCION
 - 1.1 Objetivo
 - 1.2 Alcance
 - 1.3 Definiciones, acrónimos y abreviaciones
 - 1.4 Referencias
 - 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. APENDICES
5. INDICE
6. ANEXOS

1.2.4.2.2 Descripción Del Diseño Del Software (SDD)

La SDD sigue las normativas del estándar ANSI/IEEE – Std 1016 y tiene como objetivo proporcionar una representación visual y detallada del diseño del software para cumplir con los requisitos de la Especificación de Requisitos de Software (SRS). Incluye una descripción exhaustiva de componentes y subcomponentes, como bases de datos e interfaces internas. Se inicia como un Preliminary SDD (SDD de Nivel Superior) y se expande para crear el SDD

Detallado, asegurando que el diseño cumpla con las especificaciones de los requisitos. La SDD no solo guía a los desarrolladores, sino que también es esencial para la verificación y validación del software, garantizando que se ajuste a la visión y requisitos originales de "EmoCare".

Organización de la SDD dentro de Vistas de Diseño

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.

MODELO A USAR PARA EL CONTENIDO DEL SDD

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
 - 3.1.2. 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
4. DESCRIPCION DE DEPENDENCIA
 - Dependencia entre módulos
 - Dependencia entre procesos
 - Dependencia entre datos
5. DESCRIPCION DE INTERFACES
 - 5.1 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
7. APENDICES
8. INDICE

9. ANEXOS

1.2.4.2.3 Plan de Verificación y Validación de Software (PVVS)

El Plan de Verificación y Validación de Software (PVVS) tiene como objetivo identificar y describir los métodos (por ejemplo, inspección, análisis, demostración o prueba) que se utilizarán para:

- Verificar que los requisitos en la Especificación de Requisitos de Software (SRS) hayan sido aprobados por la autoridad correspondiente.
- Verificar que los requisitos en la SRS se implementen en el diseño expresado en la Descripción del Diseño del Software (SDD).
- Verificar que el diseño expresado en la SDD se implemente en el código.
- Validar que el código, cuando se ejecute, cumpla con los requisitos expresados en la SRS.

Este PVV se basa en los estándares ANSI/IEEE Std 829-1983, ANSI/IEEE Std 1008-1987 y ANSI/IEEE Std 1012-1986 para garantizar un proceso riguroso de verificación y validación que asegure la calidad y la conformidad del software "EmoCare".

MODELO A USAR PARA EL CONTENIDO DEL SVVP

1. OBJETIVO
2. ALCANCE
3. DEFINICIONES, ACRONIMOS Y ABREVIACIONES
4. ORGANIZACIÓN RESPONSABLES
5. CICLO DE VIDA DE VERIFICACION Y VALIDACION
6. APENDICE
7. 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.

1.2.4..2.3.1 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
<p>Proceso de Gestión</p> <p>El proceso de gestión implica la administración del proyecto.</p>	<p>Gestión del esfuerzo de la verificación y validación</p> <p>La actividad de Gestión implica:</p> <ul style="list-style-type: none"> -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. 	<p>Tareas de Gestión</p> <ol style="list-style-type: none"> 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.
<p>Proceso de Adquisición</p> <p>El segundo proceso a realizar es el de adquisición este proceso inicia con definir una necesidad existente, posteriormente se plantea una solicitud, luego se</p>	<p>Apoyo de adquisición V & V</p> <p>Hace posible la preparación de contratos, el monitoreo, la aceptación y conclusión del proyecto.</p>	<p>Tareas en la Adquisición</p> <ol style="list-style-type: none"> 1) Determinación del alcance. 2) Planificación de la interfaz. 3) Evaluación de los requisitos del sistema

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:		4) Apoyo para la aceptación de adquisición.
<p>Proceso de Suministro</p> <p>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.</p>	<p>Planificación de V & V</p> <p>La actividad de planificación V & V aborda la iniciación, preparación de actividades de respuesta, contrato, planificación, ejecución, control, revisión, evaluación, entrega y conclusión.</p>	<p>Tareas para Planificación de V&V</p> <p>1) Planificación de la interfaz entre el esfuerzo de V&V y proveedor</p> <p>2) Contrato de verificación</p>
<p>Proceso de Desarrollo.</p> <p>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.</p>	<p>Actividad 1: V&V de Concepto</p> <p>El objetivo de la V&V del concepto es verificar la asignación de requisitos del sistema, validar la solución seleccionada y garantizar que ningunas hipótesis falsas se hallan incorporado en la solución.</p>	<p>Tareas de Concepto V&V</p> <p>1) Evaluación de documentación de concepto.</p> <p>2) Análisis de criticidad.</p> <p>3) Análisis de asignación de requisitos.</p> <p>4) Análisis de trazabilidad.</p> <p>5) Análisis de riesgo.</p> <p>6) Análisis de la seguridad.</p>
	<p>Actividad 2: V&V de Requisitos</p> <p>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.</p>	<p>Tareas para V&V de requisitos</p> <p>1) Análisis de trazabilidad</p> <p>2) Evaluación de los requisitos de Software</p> <p>3) Análisis de la interfaz</p> <p>4) Análisis de criticidad</p>

		<p>5) Identificación plan de V&V.</p> <p>6) Generación de plan de prueba de aceptación de V&V</p> <p>7) Evaluación de la gestión de configuración</p> <p>8) Análisis de posibles fallos.</p> <p>9) Análisis de la seguridad</p>
	<p>Actividad 3: V&V Del Diseño</p> <p>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.</p>	<p>Tareas para la V&V del Diseño</p> <p>1) Análisis de trazabilidad</p> <p>2) Evaluación de diseño de Software</p> <p>3) Análisis de la interfaz</p> <p>4) Análisis de criticidad</p> <p>5) Componente V&V plan de generación de prueba</p> <p>6) Generación de plan de prueba de integración V&V</p> <p>7) Generación de diseño de prueba componente V&V</p> <p>8) Generación de diseño de prueba de integración V&V</p> <p>9) Generación de diseño de prueba de sistema de V&V</p> <p>10) Generación de diseño de la prueba de aceptación de V&V</p> <p>11) Análisis de riesgo</p> <p>12) Análisis de la seguridad</p>
	<p>Actividad 4: V&V de Test</p>	<p>Tareas para la V&V del Test</p> <p>1) Análisis de trazabilidad</p>

	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	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
	Actividad 5: 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	V&V de Mantenimiento El objetivo de mantenimiento V & V es evaluar los cambios en el sistema propuesto de software y su impacto en	Tareas de V&V Mantenimiento 1) Revisión SVVP 2) Evaluación de anomalías 3) Análisis de criticidad

recursos y planes necesarios para realizar modificaciones, migraciones y retiros o remplazos en el software.	el software, evaluar las anomalías se descubren durante la operación, evaluar los requerimientos de migración, evaluar los requisitos de jubilación.	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
--	--	---

1.2.4.2.4 Informe de Verificación y Validación de Software (IVVS)

El Informe de Verificación y Validación de Software (IVVS) o Software Verification and Validation Report (SVVR) tiene como finalidad describir detalladamente los resultados obtenidos durante la ejecución del Plan de Verificación y Validación de Software (PVV). Este informe sigue un formato específico para documentar de manera efectiva las conclusiones del proceso de verificación y validación.

1.2.4.2.4.1 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

.....

.....

.....

1.2.4..2.4.2 Reporte de Anomalías

REPORTE DE ANOMALIAS

Pág.

FASE:
# De Reporte:	Fecha: / /	
Lugar:	Hora:	
<div style="margin-bottom: 10px;"> a) Descripción y ubicación: <div style="border-bottom: 1px dashed black; height: 15px; margin-top: 5px;"></div> <div style="border-bottom: 1px dashed black; height: 15px; margin-top: 5px;"></div> </div> <div style="margin-bottom: 10px;"> b) Impacto: <div style="border-bottom: 1px dashed black; height: 15px; margin-top: 5px;"></div> <div style="border-bottom: 1px dashed black; height: 15px; margin-top: 5px;"></div> </div> <div style="margin-bottom: 10px;"> c) Causa: <div style="border-bottom: 1px dashed black; height: 15px; margin-top: 5px;"></div> <div style="border-bottom: 1px dashed black; height: 15px; margin-top: 5px;"></div> </div> <div style="margin-bottom: 10px;"> d) Credibilidad: <div style="border-bottom: 1px dashed black; height: 15px; margin-top: 5px;"></div> <div style="border-bottom: 1px dashed black; height: 15px; margin-top: 5px;"></div> </div> <div style="margin-bottom: 10px;"> e) Recomendaciones: <div style="border-bottom: 1px dashed black; height: 15px; margin-top: 5px;"></div> <div style="border-bottom: 1px dashed black; height: 15px; margin-top: 5px;"></div> </div>		

Equipo de Trabajo:

Nombre:

Firma

.....

.....

.....

1.2.4..2.4.3

Reporte Final de V&V

REPORTE FINAL DE V&V		Pág.
FASE:
# De Reporte:	Fecha: ____ / ____ / ____	
Lugar:	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:

.....

.....

d) Evaluación total de la calidad del software:

.....

.....

e) Recomendaciones:

.....

.....

Equipo de Trabajo:

Nombre:

Firma

.....

.....

.....

1.2.4.2.5 Documentación del Usuario (UD)

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:

- Proveer el ambiente y la información necesaria para entender el sistema.
- Proveer la información necesaria para aprender lo que puede hacer con el software y como lo puede usar.
- Proveer 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 incluidos son:

- Manual de comandos.
- Manual de mensajes de error.
- Manual de llamadas de programas.
- Guía de referencia rápida.
- Manual de Herramientas del software.
- Manual de utilitarios.

MODELO A USAR PARA EL CONTENIDO DEL UD

1. TITULO DE LA PAGINA
2. RESTRICCIONES
3. GARANTIAS Y OBLIGACIONES CONTRACTUALES
4. TABLA DE CONTENIDO
5. LISTA DE ILUSTRACIONES
6. INTRODUCCION
 - Descripción de audiencia
 - Declaración de aplicación
 - Declaración de objetivos
 - 6.4 Descripción del uso de documentos
 - 6.5 Documentos relacionados
 - 6.6 Convenciones
 - 6.6.1 Símbolos
 - 6.6.2 Convenciones de estilo
 - 6.6.3 Convenciones de sintaxis de comandos
 - 6.7 Instrucciones de reportes de problemas
7. CUERPO DEL DOCUMENTO
 - 7.1. Cuerpo del documento en modo instruccional
 - 7.1.1 Alcance
 - 7.1.2 Materiales
 - 7.1.3 Preparaciones
 - 7.1.4 Precauciones y prevenciones

- 7.1.5 Métodos
- 7.1.6 Información relacionada
- 7.2. Cuerpo del documento en modo de referencia
 - 7.2.1 Objetivo
 - 7.2.2 Materiales
 - 7.2.3 Preparaciones
 - 7.2.4 Entradas
 - 7.2.5 Precauciones y prevenciones
 - 7.2.6 Invocación
 - 7.2.7 Operaciones de suspensión
 - 7.2.8 Operaciones de terminación
 - 7.2.9 Salidas
 - 7.2.10 Condiciones de error
 - 7.2.11 Información relacionada
- 8. MENSAJES DE ERROR, CONOCIMIENTO DE PROBLEMAS, RECUPERACION DE ERROR
- 9. ANEXOS
- 10. BIBLIOGRAFIA
- 11. GLOSARIO
- 12. INDICE

1.2.4.2.6 Plan de Gestión de Configuración de Software (SCMP)

El Plan de Gestión de Configuración de Software (SCMP) de EmoCare establece los procesos y procedimientos para identificar, controlar, implementar y gestionar cambios en el software a lo largo de su ciclo de vida. Este plan se adhiere a los estándares reconocidos en la industria y tiene como objetivo principal garantizar la integridad, trazabilidad y control de la configuración del software del proyecto EmoCare.

Procesos de Configuración:

1. Identificación de Elementos de Configuración (CI):

Se identificarán todos los elementos de configuración del software, incluyendo código fuente, módulos, bibliotecas, documentación y archivos de datos.

2. Control de Cambios:

Se establecerá un proceso para la solicitud, revisión, aprobación y seguimiento de cambios en los elementos de configuración del software.

3. Implementación de Cambios:

Se documentará el procedimiento para implementar los cambios aprobados en el software, asegurando que se realicen de manera controlada y se mantenga la trazabilidad.

4. Registro y Reporte:

Se llevará un registro detallado de todos los cambios realizados, incluyendo su estado y fecha de implementación. Se generará un informe de cambios periódico para su revisión.

1.2.5 Estándares, Practicas y Convenciones

1.2.5.1 Estándar de Codificación

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

1.2.5.2 Estándar 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.

1.2.5.3 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

1.2.6 Revisiones y Auditorias

Las revisiones que se aplicarán involucrarán una minuciosa evaluación de los componentes, procesos y documentos relacionados, con el objetivo de identificar cualquier error o área de mejora. Estas revisiones serán una parte integral del proceso de desarrollo, asegurando que se cumplan los estándares de calidad y que el trabajo avance de manera eficiente hacia sus objetivos

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.

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

Revisión de los Requisitos de Software (SRR):

La SRR se lleva a cabo para evaluar las especificaciones de requerimientos del software (SRS).

Su objetivo principal es asegurar que los requerimientos establecidos en la SRS sean correctos y completos, garantizando así la calidad, viabilidad e integridad de los requerimientos.

Criterios de revisión incluyen la fiabilidad, completitud, depurabilidad, modificabilidad, consistencia, ausencia de ambigüedades y utilidad durante la fase de operación y mantenimiento.

Revisión del Diseño Preliminar (PDR):

La PDR se realiza para evaluar la suficiencia técnica del diseño preliminar (SDD) antes de iniciar el diseño detallado.

Su propósito es verificar la coherencia y suficiencia técnica del alcance del diseño con los requerimientos funcionales de la SRS.

Se revisan las interfaces con otros componentes, se verifica la compatibilidad y se establecen requisitos relacionados con factores humanos.

Revisión del Diseño Crítico (CDR):

La CDR se genera para evaluar la conformidad del diseño detallado (SDD) con la SRS.

Evalúa la representación de datos, la compatibilidad de interfaces, y garantiza que el diseño cumple con los requerimientos funcionales establecidos en la SRS.

Revisión del Plan de Verificación y Validación (SVVPR):

El SVVPR se utiliza para evaluar los métodos de verificación y validación definidos en el SVVP, así como el cumplimiento con ellos durante el desarrollo del software.

Se realizan revisiones incrementales para garantizar que los métodos de verificación y validación sean adecuados para el software en desarrollo

Auditoría Funcional:

Esta auditoría se efectúa antes de la entrega del software para verificar el cumplimiento de todos los requerimientos especificados en la SRS.

Se compara el código con los requerimientos documentados en la SRS para garantizar que el software funcione de acuerdo con las especificaciones.

Auditoría Física (PA):

La auditoría física tiene como objetivo asegurar que el software y su documentación sean coherentes y estén listos para su entrega.

Se verifica la consistencia entre el código y su documentación de soporte, garantizando que la documentación entregada refleje correctamente el código.

Auditorías del Proceso (IPA):

Estas auditorías se realizan dentro de los procesos de desarrollo del software para verificar la consistencia del diseño y la implementación.

Se evalúa la compatibilidad de interfaces, la verificación de requerimientos y la coherencia del producto a lo largo del proceso de desarrollo.

Revisiones de Gestión:

Las revisiones de gestión se llevan a cabo periódicamente para evaluar la ejecución del SQAP y se realizan por el elemento organizacional del consultor.

1.2.6.1 Evaluación de la calidad de los productos

Objetivo: El propósito de esta actividad es examinar detenidamente los productos que se han identificado como críticos para garantizar la calidad. Su finalidad principal es detectar cualquier desviación con respecto a los estándares y a los objetivos de calidad predefinidos, y comunicar estas discrepancias a los responsables para que se tomen medidas correctivas.

Proceso: En esta etapa, se procede a revisar minuciosamente los productos para asegurarse de que cumplen con los estándares de calidad previamente establecidos y se ajustan a los objetivos definidos para cada producto. Se realiza una comparación exhaustiva para garantizar que no se pasen por alto correcciones pendientes de los informes de revisiones anteriores. Cualquier desviación identificada se documenta de manera precisa, se sigue su evolución y se verifica que se hayan implementado las correcciones necesarias.

Resultado: Como resultado de esta revisión, se genera un Informe de Evaluación del Aseguramiento de Calidad del Software (SQA) que detalla todas las desviaciones o defectos identificados durante la evaluación. Este informe se distribuye a los responsables de los productos evaluados, y se asegura que estén al tanto de las discrepancias encontradas y de las medidas correctivas que deben llevarse a cabo.

1.2.6.2 Revisar el ajuste al proceso:

Objetivo: El propósito principal de esta actividad es evaluar si los productos se han obtenido siguiendo rigurosamente las actividades definidas en el Modelo de Proceso. La meta es verificar que cada producto esté en consonancia con las etapas y procesos prescritos en el ciclo de vida del software.

Proceso: Para llevar a cabo esta revisión, se seleccionan productos clave con el fin de corroborar si se han seguido las actividades especificadas en el proceso a lo largo de todo el ciclo de desarrollo del software. Se recopila información relevante de cada producto y se retrocede para confirmar que se han generado todos los productos requeridos como entradas para el producto que se está revisando. Esta información se extrae de documentos cruciales, como el Plan del Proyecto, el Plan de la Iteración y el Plan de Verificación. Además, se verifica que se hayan seguido adecuadamente todos los pasos del proceso de desarrollo.

Resultado: Como resultado de esta revisión, se elabora un Informe de Evaluación del Ajuste al Proceso del Aseguramiento de Calidad del Software (SQA). Este informe contiene un registro de todas las desviaciones o defectos detectados durante la revisión. Se distribuye el informe a los responsables de las actividades involucradas y se asegura de que estén al tanto de las discrepancias identificadas y de las medidas correctivas que deben implementarse.

1.2.6.3 Revisión Técnica Formal

Objetivo: El propósito principal de la Revisión Técnica Formal (RTF) es identificar posibles errores en la función, lógica o implementación de cualquier producto de software, y verificar si cumple con las especificaciones establecidas y se adhiere a los estándares previamente definidos. Además, se buscan señalar y documentar las desviaciones potenciales que se detecten durante este proceso.

Proceso: La RTF es una evaluación exhaustiva y rigurosa que tiene como objetivo detectar de manera temprana cualquier defecto o discrepancia en los productos generados a lo largo del desarrollo de software. Esta práctica se aplica especialmente a productos de gran importancia. La RTF involucra la participación del responsable de Aseguramiento de Calidad del Software (SQA) y miembros del equipo de desarrollo. La convocatoria a la reunión se realiza de manera formal, se informa a los participantes sobre la preparación previa necesaria, y se lleva una lista de preguntas y posibles dudas que surgen durante el análisis del producto que será revisado. Como resultado de la RTF, se genera un Informe de RTF que documenta los hallazgos y las conclusiones de la revisión.

1.2.6.4 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

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 desea también la fecha en la que se realizara la revisión del entregable o producto	Tipo de revisión que se realizara: Evaluación de la calidad de los productos, Revisar el ajuste al proceso o RTF

Y de igual manera para cada una de las fases

1.2.7 Gestión de Configuración

La finalidad del Aseguramiento de Calidad del Software (SQA) en esta área se centra en garantizar que se ejecuten las actividades de gestión de configuración definidas en el Plan de Configuración y que se cumplan según lo establecido en el proceso. Se pueden identificar las siguientes actividades mínimas que deben llevarse a cabo:

- Verificar que la Línea Base del proyecto se haya creado según lo previsto en el modelo de proceso y en el momento indicado.
- Asegurarse de que la Línea Base del proyecto generada sea precisa y se ajuste a las especificaciones.
- Realizar revisiones periódicas para confirmar que el Responsable de Control de Configuración (SCM) esté manteniendo adecuadamente el control sobre la Línea Base, incluyendo un registro completo de los cambios realizados en los requisitos, el diseño, el código, la verificación y la documentación.

- Supervisar los procedimientos del Comité de Control de Cambios para verificar que se lleven a cabo de acuerdo con lo establecido en el Plan de Configuración.

Esta sección asegura que se establezcan prácticas sólidas de gestión de configuración, lo que garantiza que las versiones del proyecto estén controladas y documentadas de manera adecuada, y que se realice un seguimiento efectivo de los cambios a lo largo del desarrollo del software.

1.2.8 Gestión de problemas y acciones correlativas

En esta sección se delinean las prácticas y procedimientos que se utilizarán para la notificación, el seguimiento y la resolución de problemas relacionados con el software, junto con las responsabilidades organizativas correspondientes. El objetivo de un sistema de Gestión de Problemas y Acciones Correlativas es el siguiente:

- Garantizar que todos los problemas se documenten, se aborden y no se pasen por alto.
- Evaluar la autenticidad de los informes sobre problemas.
- Proporcionar una retroalimentación constante tanto a los desarrolladores como a los usuarios sobre el estado de los problemas.
- Suministrar datos que permitan medir y predecir la calidad y la confiabilidad del software.

Cualquier problema identificado en el producto de software durante el ciclo de vida del desarrollo debe ser comunicado mediante un informe que incluya la fecha de detección del problema, una identificación preliminar, una descripción detallada, entre otros detalles

relevantes. Este informe debe contar con la firma de quienes detectaron el problema y se entregará a la organización responsable de la gestión de problemas.

La organización encargada de gestionar los problemas de software es la organización de Aseguramiento de Calidad del Software (SQA), bajo la supervisión de la organización del consultor. Estas organizaciones son responsables de determinar el calendario, el lugar y la agenda necesarios para llevar a cabo las acciones correctivas pertinentes en relación con los problemas identificados.

1.2.9 Herramientas Técnicas 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 10
- Debuggeadores
- Documentos de ayuda
- Analizadores de código
- Monitores de Rendimiento

Entre las técnicas que ayudaran en 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

1.2.10 Control de Código

Esta sección tiene como objetivo establecer los métodos, técnicas y recursos que se utilizarán para supervisar el almacenamiento y el mantenimiento de las diferentes versiones del código del software.

El procedimiento de control del código se define de manera que:

- Identifica claramente qué software estará sujeto a control.
- Describe un método estándar para la identificación, etiquetado y catalogación del software.
- Lista la ubicación física donde se encuentra el software bajo control.
- Describe cómo se almacenan, mantienen y utilizan las copias de seguridad.
- Detalla los procedimientos para la distribución de copias del software.
- Identifica la documentación que se verá afectada por cualquier cambio realizado en el código.
- Explica los procedimientos necesarios para la creación de una nueva versión del software

1.2.11 Control de Medios

En esta sección se establecen los procedimientos y recursos destinados a salvaguardar el medio físico de accesos no autorizados y posibles daños, así como a prevenir degradaciones inesperadas. Además, se identifican las organizaciones responsables de llevar a cabo este control, con supervisión por parte de la organización de Aseguramiento de Calidad del Software (SQA).

La tarea de proteger el medio físico recae en la organización de desarrollo, que garantizará que se cumplan los siguientes aspectos clave:

- Garantizar el almacenamiento y la recuperación confiable del software.
- Restringir el acceso al software solo a quienes tengan una necesidad legítima.
- Controlar el entorno para evitar cualquier degradación del medio físico donde se almacena el software.
- Mantener copias seguras del software crítico y del código en línea base fuera de las instalaciones de la organización.

Respecto a los medios de almacenamiento, se definen como los dispositivos utilizados para almacenar datos, y se utilizarán diversos medios para garantizar la integridad y la disponibilidad del software:

- Discos duros como medios primarios.
- CD's para almacenamiento secundario, destinados a guardar copias de seguridad.
- Documentación relacionada con el desarrollo de software en formato papel.

El proceso de copias de seguridad se llevará a cabo al finalizar cada sesión de trabajo, con un registro que incluye la fecha y hora de la copia de seguridad.

Para evitar accesos no autorizados, se asignarán cuentas privilegiadas a usuarios, y cada usuario tendrá su propia cuenta según su función. La integridad de la Base de Datos se utilizará para proteger los datos. Se llevará a cabo una revisión periódica del software para garantizar su óptimo funcionamiento.

1.2.12 Control de suministros y subcontratos

El control de suministros y subcontratos es una parte esencial de la gestión de calidad del software y asegura que todos los recursos externos y materiales suministrados se gestionen de manera eficaz y cumplan con los estándares de calidad definidos

1.2.13 Recolección, Mantenimiento y Retención de Registros

Las responsabilidades relacionadas con esta sección recaen en la organización del consultor, trabajando en estrecha coordinación con el equipo de Aseguramiento de Calidad del Software (SQA).

En esta sección, se define con precisión qué documentación debe ser preservada y se establecen los métodos y recursos que se emplearán para recolectar, resguardar y mantener dicha documentación. Además, se detalla el período de retención para cada tipo de registro.

Es importante destacar que no solamente se registrará documentación, sino también se conservarán los medios físicos que almacenan las versiones de los programas y los materiales empleados en las pruebas, garantizando así la capacidad de repetir las pruebas en el futuro si fuera necesario.

La lista de documentos requeridos incluye:

- El Plan de Garantía de Calidad del Software.
- La Especificación de Requerimientos del Software.
- La Descripción del Diseño del Software.
- El Plan de Verificación y Validación del Software.
- La Documentación del Usuario.

Para el mantenimiento de los registros del software, se llevarán a cabo actualizaciones sucesivas, y se mantendrá un registro de todas las modificaciones realizadas en la documentación.

Es fundamental que los documentos que han sido verificados y validados sean registrados en libros impresos. Se almacenarán tres copias de cada documento en ubicaciones diferentes y en ambientes adecuados para su preservación segura.

La retención de registros se realizará al concluir cada fase del ciclo de vida del desarrollo de software, así como en consonancia con los puntos de verificación y validación.

Esta versión mantiene la estructura y la esencia del fragmento original pero utiliza diferentes palabras y estructuras de oraciones para evitar que parezca el mismo texto.