**UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO**

FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES

**PROYECTO DE INGENIERIA DE SOFTWARE 2**

**PLATAFORMA INTEGRAL DE INFORMACIÓN Y SERVICIOS URBANOS**

**“UrbanEase Solutions”**

**MATERIA:** INGENIERIA DE SOFTWARE 2

**DOCENTE:** ING.ROLANDO ANTONIO MARTINEZ CANEDO

**INTEGRANTES:**

* SAHONERO SALAS JOSE ALEJANDRO 218048513
* SALAZAR VARGAS GUIDO 220029938

**SANTA CRUZ DE LA SIERRA – BOLIVIA**

**MAYO - 2024**

**INDICE**

[**1.1 Introducción** 4](#_Toc166577224)

[**1.2 Antecedentes** 4](#_Toc166577225)

[**1.3 Objetivos** 5](#_Toc166577226)

[**1.3.1 Objetivo General** 5](#_Toc166577227)

[**1.3.2 Objetivos Específicos** 6](#_Toc166577228)

[**1.3.3 Misión** 6](#_Toc166577229)

[**1.3.4 Visión** 7](#_Toc166577230)

[**1.6** **Políticas de Calidad** 7](#_Toc166577231)

[**1.6.1 Principios de Calidad** 7](#_Toc166577232)

[**1.7 Slogan** 8](#_Toc166577233)

[**2.1 Plan de Aseguramiento de Calidad de Software (SQAP)** 8](#_Toc166577234)

[**2.1.1 Propósito** 8](#_Toc166577235)

[**2.1.2 Objetivo** 9](#_Toc166577236)

[**2.1.3 Descripción** 10](#_Toc166577237)

[**2.1.3.1 Modelo de calidad interna y externa** 10](#_Toc166577238)

[**2.1.4 Alcance** 11](#_Toc166577239)

[**2.1.4.1 Planificación de Calidad** 11](#_Toc166577240)

[**2.1.4.1 Requisitos** 11](#_Toc166577241)

[**2.1.4.2 Diseño** 11](#_Toc166577242)

[**2.1.4.3 Implementación** 12](#_Toc166577243)

[**2.1.4.4 Pruebas** 12](#_Toc166577244)

[**2.1.4.5 Despliegue** 12](#_Toc166577245)

[**2.1.4.6 Mantenimiento** 12](#_Toc166577246)

[**2.1.4.7 Medición y Mejora Continua** 12](#_Toc166577247)

[**2.1.4.8 Documentación y Revisiones** 13](#_Toc166577248)

[**2.1.4.9 Documentos de Referencia**. 13](#_Toc166577249)

[**2.1.5 Alcance** 14](#_Toc166577250)

[**2.1.6 Organización** 15](#_Toc166577251)

[**2.1.6 Tareas** 16](#_Toc166577252)

[**2.1.7 Roles y Responsabilidades** 17](#_Toc166577253)

[**2.1.7.1 Liderazgo Ejecutivo** 17](#_Toc166577254)

[**2.1.7.2 Equipo de Gestión de Proyectos** 18](#_Toc166577255)

[**2.1.7.3 Equipo de Desarrollo** 18](#_Toc166577256)

[**2.1.7.4 Equipo de Calidad y Pruebas** 19](#_Toc166577257)

[**2.1.7 Documentación** 20](#_Toc166577258)

[**2.1.7.1 Propósito** 20](#_Toc166577259)

[**2.1.7.2 Requisitos Mínimos de Documentación** 20](#_Toc166577260)

[**2.1.7.2.1 Especificación de requisitos de software (SRS)** 20](#_Toc166577261)

[**2.1.7.2.2 Descripción del diseño del software (SDD)** 22](#_Toc166577262)

[**2.1.7.2.3 Plan de verificación y validación de software (PVVS)** 25](#_Toc166577263)

[**2.1.7.2.4 Informe de verificación y validación de software (IVVS)** 29](#_Toc166577264)

[**2.1.7.2.5 Documentación del Usuario (UD)** 31](#_Toc166577265)

[**2.1.7.2.6 Plan de gestión de configuración del software (SCMP)** 34](#_Toc166577266)

[**2.1.9 Estándares, Practicas y Convenciones** 35](#_Toc166577267)

[**2.1.9.1 Estándar de Codificación** 35](#_Toc166577268)

[**2.1.9.2 Estándar de Comentarios** 37](#_Toc166577269)

[**2.1.9.3 Estándar de verificar el cumplimiento** 37](#_Toc166577270)

[**2.1.10 Revisión y Auditorias** 38](#_Toc166577271)

[**2.1.10.1 Evaluación de la calidad de los productos** 40](#_Toc166577272)

[**2.1.10.2 Revisar el ajuste al proceso** 41](#_Toc166577273)

[**2.1.10.3 Revisar el ajuste al proceso** 42](#_Toc166577274)

[**2.1.10.4 Revisión Técnica Formal** 43](#_Toc166577275)

[**2.1.10.4 Requerimientos Mínimos** 43](#_Toc166577276)

[**2.1.11 Gestión de Configuración** 45](#_Toc166577277)

[**2.1.12 Gestión de problemas y acciones correlativas** 46](#_Toc166577278)

[**2.1.13 Herramientas Técnicas y Metodologías** 47](#_Toc166577279)

[**2.1.14 Control de Código** 47](#_Toc166577280)

[**2.1.15 Control de Medios** 48](#_Toc166577281)

[**2.1.16 Control de Suministros y Subcontratos** 49](#_Toc166577282)

[**2.1.17 Recolección, Mantenimiento y Retención de Registros** 49](#_Toc166577283)

# **1.1 Introducción**

En UrbanEase Solutions, la excelencia en el desarrollo de software es nuestra piedra angular. Reconocemos que en un mundo cada vez más digitalizado, la calidad del software es crucial en todos los aspectos de la vida cotidiana. Este documento representa nuestra hoja de ruta hacia estándares de calidad sobresalientes, reflejando nuestro firme compromiso con la excelencia en cada línea de código que escribimos.

Nos distinguimos por nuestro enfoque tridimensional para garantizar la calidad del software, abordando las perspectivas del usuario, del desarrollador y del producto. Nuestra gestión de calidad está meticulosamente estandarizada y certificada, lo que nos permite ofrecer soluciones tecnológicas de primera categoría que superan las expectativas de nuestros clientes.

En UrbanEase Solutions, la calidad del software no es solo una prioridad, sino nuestra ventaja competitiva en un mercado en constante cambio. Entendemos que la calidad no solo es una inversión esencial para nuestro éxito, sino también para el de nuestros clientes, quienes confían en nosotros para brindarles soluciones confiables y seguras que mejoren su vida diaria en entornos urbanos.

# **1.2 Antecedentes**

UrbanEase Solutions es una prometedora startup fundado por dos estudiantes de la Universidad Gabriel René Moreno, con una visión compartida de impulsar la innovación en el mundo del desarrollo de software. La empresa surge como respuesta a la creciente necesidad de soluciones tecnológicas integrales que aborden los desafíos urbanos modernos, combinando la pasión por la tecnología con la experiencia en áreas especializadas como la georreferenciación, el manejo de datos geoespaciales, el desarrollo backend y web, la inteligencia artificial y más.

A pesar de ser una empresa joven, UrbanEase Solutions está construyendo un sólido historial en el campo de la tecnología. Con una estructura basada en la colaboración estrecha y la complementación de habilidades, el equipo fundador se dedica a ofrecer soluciones innovadoras y de alta calidad que superen las expectativas de sus clientes y contribuyan al desarrollo sostenible de las comunidades urbanas.

Guido Salazar Vargas aporta su experiencia en dirección de proyectos bajo la metodología ágil, así como en bases de datos y desarrollo web. Por otro lado, Alejandro Sahonero Salas se especializa en servidores, backend, desarrollo móvil e inteligencia artificial. Esta combinación de conocimientos y habilidades permite a UrbanEase Solutions abordar una amplia gama de desafíos tecnológicos de manera efectiva, asegurando el éxito de cada proyecto emprendido.

A pesar de su corta trayectoria, UrbanEase Solutions ha demostrado su capacidad para enfrentar proyectos de desarrollo de software de alta complejidad y ha ganado la confianza de sus clientes y colaboradores. La dedicación a la calidad, la innovación y la satisfacción del cliente son los pilares fundamentales de UrbanEase Solutions, que se compromete a seguir marcando la diferencia en el mundo del desarrollo de software y contribuir al progreso de las comunidades urbanas a través de soluciones tecnológicas de vanguardia.

# **1.3 Objetivos**

## **1.3.1 Objetivo General**

UrbanEase Solutions aspira a liderar el desarrollo de software innovador, destacándose por su excelencia en calidad y su capacidad para ofrecer soluciones tecnológicas que impulsen el progreso en entornos urbanos, satisfaciendo las necesidades de sus clientes y usuarios finales.

## **1.3.2 Objetivos Específicos**

* Desarrollar soluciones de software vanguardistas que cumplan con los más altos estándares de calidad y se adapten de manera precisa a las necesidades específicas de nuestros clientes, abordando de manera efectiva los desafíos urbanos modernos.
* Fomentar un ambiente de colaboración y creatividad dentro de nuestro equipo, promoviendo la generación de soluciones tecnológicas únicas e innovadoras que marquen la diferencia en la vida cotidiana de las comunidades urbanas.
* Establecer relaciones sólidas y de confianza con nuestros clientes, basadas en la transparencia, la comunicación abierta y la entrega oportuna de proyectos, garantizando su satisfacción y fidelidad a largo plazo.
* Cultivar una cultura organizacional centrada en la calidad y la mejora continua, donde cada miembro del equipo se comprometa a superar constantemente los estándares establecidos y a buscar formas innovadoras de optimizar nuestros procesos y productos.
* Invertir en el desarrollo profesional y técnico de nuestro equipo, brindando oportunidades de formación y crecimiento que les permitan potenciar sus habilidades y conocimientos, asegurando así su capacidad para enfrentar los desafíos tecnológicos del futuro con éxito.

## **1.3.3 Misión**

En UrbanEase Solutions, nuestra misión es impulsar la evolución digital de las ciudades, proporcionando soluciones tecnológicas integrales y de vanguardia que optimicen la experiencia urbana. Nos comprometemos a desarrollar software innovador que responda a las necesidades cambiantes de las comunidades urbanas, mejorando la eficiencia, la seguridad y la calidad de vida de sus habitantes.

## **1.3.4 Visión**

Nos visualizamos como líderes en el campo del desarrollo de software para entornos urbanos, siendo reconocidos a nivel mundial por nuestra capacidad para ofrecer soluciones tecnológicas que transforman positivamente las ciudades. Aspiramos a ser catalizadores del progreso urbano, impulsando la creación de comunidades más conectadas, sostenibles e inclusivas a través de nuestra innovación y excelencia en el desarrollo de software.

## **1.6 Políticas de Calidad**

En UrbanEase Solutions, nos embarcamos en una travesía de excelencia, comprometiéndonos a trazar el camino hacia la cima del desarrollo de software y soluciones tecnológicas. Nuestra política de calidad no es solo un documento; es la esencia misma de nuestra empresa, impregnada en cada línea de código y en cada decisión que tomamos. Guiados por estándares de calidad inflexibles, elevamos el estándar en cada etapa de nuestro proceso, asegurando que cada producto que sale de nuestras manos sea un testimonio de nuestra dedicación y compromiso con la perfección. En UrbanEase Solutions, la calidad no es solo una meta; es nuestro legado, nuestra promesa y nuestra pasión.

## **1.6.1 Principios de Calidad**

* **Compromiso con la excelencia:** Nos comprometemos a alcanzar y mantener altos estándares de calidad en todas las etapas del desarrollo de software, desde la concepción hasta la entrega final.
* Enfoque centrado en el usuario: Priorizamos las necesidades y expectativas de nuestros usuarios finales en el diseño y desarrollo de software, asegurando que cada producto satisfaga sus requisitos de manera efectiva y eficiente.
* **Adhesión a estándares y mejores prácticas:** Seguimos rigurosamente estándares reconocidos de la industria y adoptamos las mejores prácticas en el desarrollo de software, para garantizar la coherencia, la calidad y la confiabilidad de nuestros productos.
* **Mejora continua:** Buscamos constantemente oportunidades para mejorar nuestros procesos, productos y servicios a través de la retroalimentación, la innovación y la implementación de acciones correctivas y preventivas.
* **Transparencia y responsabilidad:** Fomentamos una cultura de transparencia y responsabilidad en todas nuestras operaciones, asumiendo la responsabilidad por la calidad y el rendimiento de nuestro software ante nuestros clientes y todas las partes interesadas.
* **Colaboración y comunicación:** Promovemos la colaboración y la comunicación efectiva entre todos los miembros del equipo y con nuestros clientes, para garantizar la alineación y el éxito en la entrega de proyectos de software.

## **1.7 Slogan**

UrbanEase Solutions: Conectando Ciudades, Transformando Vidas

# **2.1 Plan de Aseguramiento de Calidad de Software (SQAP)**

## **2.1.1 Propósito**

El propósito del Plan de Aseguramiento de Calidad de Software (SQAP) en UrbanEase Solutions es garantizar la entrega de productos de software que cumplan con los más altos estándares de calidad. Este plan se enfoca en establecer procesos, procedimientos y controles que aseguren la fiabilidad, seguridad y eficacia de nuestros productos, así como en promover una cultura de calidad en toda la organización.

## **2.1.2 Objetivo**

El objetivo primordial del Plan de Aseguramiento de Calidad de Software (SQAP) en UrbanEase Solutions es establecer un sólido marco metodológico que asegure la entrega consistente de productos de software de alta calidad. Este plan tiene como propósito fundamental garantizar que los productos desarrollados por la empresa cumplan con los requisitos específicos del cliente, así como con los estándares de calidad y las mejores prácticas de la industria.

Para lograr este objetivo, el SQAP se enfoca en:

* Establecer procesos y procedimientos claros y efectivos para el desarrollo de software, desde la planificación hasta la entrega.
* Definir roles y responsabilidades precisas para los miembros del equipo, asegurando una distribución adecuada de tareas y una clara asignación de responsabilidades.
* Implementar controles de calidad en cada etapa del ciclo de vida del software, con el fin de detectar y corregir posibles defectos de manera oportuna.
* Promover una cultura de calidad dentro de la organización, fomentando la colaboración, la comunicación y el aprendizaje continuo entre todos los miembros del equipo.
* Mejorar continuamente los procesos y prácticas de desarrollo de software, mediante la retroalimentación, la evaluación de desempeño y la implementación de acciones correctivas y preventivas.

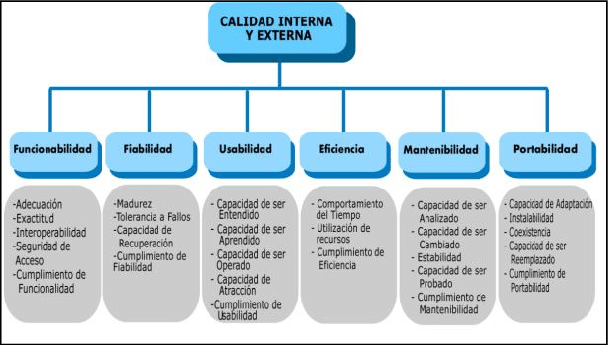
El SQAP de UrbanEase Solutions busca garantizar la calidad, confiabilidad y satisfacción del cliente en todos los productos de software desarrollados por la empresa, al tiempo que impulsa la innovación y la excelencia en todas sus operaciones.

## **2.1.3 Descripción**

El Plan de Aseguramiento de Calidad de Software (SQAP) de UrbanEase Solutions es el faro que guía nuestra travesía hacia la excelencia en el desarrollo de productos tecnológicos. Es el documento maestro que traza las líneas directrices, establece los estándares y define los procesos que aseguran la calidad excepcional en cada línea de código que escribimos y en cada solución que entregamos.

Este no es un simple conjunto de reglas; es el alma misma de nuestra empresa, infundiendo calidad en el núcleo de nuestro ambiente empresarial. Desde la concepción de una idea hasta su realización, el SQAP garantiza que cada paso del camino esté pavimentado con precisión, innovación y dedicación implacable a la perfección.

## **2.1.3.1 Modelo de calidad interna y externa**



## **2.1.4 Alcance**

En UrbanEase Solutions, nuestro Plan de Aseguramiento de Calidad de Software (SQAP) abarca todas las fases del ciclo de vida del desarrollo de software, asegurando la calidad en cada etapa del proceso. El alcance detallado se desglosa a continuación:

## **2.1.4.1 Planificación de Calidad**

Nuestro SQAP incluye una planificación exhaustiva de la calidad en todas las etapas del proyecto. Definimos objetivos de calidad claros, asignamos responsabilidades dentro del equipo y establecemos un cronograma para llevar a cabo actividades de aseguramiento de calidad de manera oportuna.

## **2.1.4.1 Requisitos**

En esta fase, nos aseguramos de verificar y validar todos los requisitos del software. Realizamos revisiones meticulosas para garantizar la claridad y completitud de los requisitos, así como pruebas de aceptación del cliente para confirmar su cumplimiento.

## **2.1.4.2 Diseño**

Nuestro SQAP define procesos rigurosos para la revisión del diseño, evaluando la arquitectura y calidad del diseño en detalle. Verificamos que el diseño cumpla con los requisitos establecidos y seguimos las mejores prácticas de diseño.

## **2.1.4.3 Implementación**

Durante esta fase, nos enfocamos en la revisión del código, las pruebas unitarias y la gestión de la configuración del software. Garantizamos que el código cumpla con los estándares de calidad definidos y se implemente de manera eficiente.

## **2.1.4.4 Pruebas**

Establecemos un plan integral para la planificación, ejecución y documentación de pruebas de calidad en todas las etapas del proyecto. Esto incluye pruebas de unidad, pruebas de integración, pruebas de sistema y pruebas de aceptación.

## **2.1.4.5 Despliegue**

Definimos cómo se manejarán las actividades de aseguramiento de calidad durante la implementación del software en entornos de producción. Nos aseguramos de que el software sea confiable y se comporte según lo esperado en el entorno real.

## **2.1.4.6 Mantenimiento**

Gestionamos las actualizaciones, correcciones de errores y cambios a lo largo del ciclo de vida del software. Nos aseguramos de que las modificaciones no comprometan la calidad existente del software y mantengamos su integridad.

## **2.1.4.7 Medición y Mejora Continua**

Especificamos métricas de calidad para evaluar el cumplimiento de estándares y objetivos en cada fase. Además, definimos procesos para recopilar y analizar datos con el fin de mejorar continuamente nuestros procesos de desarrollo.

## **2.1.4.8 Documentación y Revisiones**

Establecemos procedimientos para documentar todas las actividades de aseguramiento de calidad y realizar revisiones y auditorías periódicas. Esto garantiza el cumplimiento de los procedimientos definidos y la calidad del software en todas las etapas del proyecto.

## **2.1.4.9 Documentos de Referencia**.

En UrbanEase Solutions, nos basamos en una variedad de estándares y guías reconocidos internacionalmente para asegurar la calidad del software en todas las etapas de nuestros proyectos. Los documentos de referencia que utilizamos incluyen:

* **IEEE STD 730-1998 y IEEE STD 730.1-1995:** Estos estándares y guías establecen los requisitos y directrices para la creación de planes de aseguramiento de calidad del software, siendo fundamentales en la planificación y gestión de la calidad del software.
* **ISO/IEC 25000 (SQuaRE - Software Product Quality Requirements and Evaluation):** Esta serie de estándares internacionales se enfoca en la calidad del producto de software, proporcionando un marco completo para evaluar y especificar la calidad del software. Es relevante para medir la calidad de nuestras soluciones en UrbanEase Solutions.
* **ISO/IEC 9126 (Software Engineering - Product Quality):** Define un modelo de calidad del software y ofrece pautas para evaluar 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 define modelos de calidad para el software, incluyendo modelos para la calidad interna y externa, proporcionando una base sólida para evaluar la calidad de nuestras soluciones.
* **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, lo que es relevante para mantener altos estándares de calidad en UrbanEase Solutions.
* **ISO/IEC 12207 (Procesos del Ciclo de Vida del Software):** Define procesos de ciclo de vida de software y proporciona una estructura para la gestión y mejora de procesos en el desarrollo de software, lo que nos ayuda a garantizar la calidad y eficiencia en nuestros proyectos.
* **ISO/IEC 15504 (SPICE - Software Process Improvement and Capability Determination):** Proporciona un marco para la evaluación y mejora de procesos de software, lo que nos permite mejorar la eficiencia y calidad de nuestros 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 realizadas en UrbanEase Solutions, asegurando la precisión y fiabilidad de nuestros resultados.

## **2.1.5 Alcance**

En la gestión del Plan de Aseguramiento de Calidad de Software (SQAP) de UrbanEase Solutions, nos enfocamos en coordinar eficazmente nuestro equipo de calidad, asignando roles y responsabilidades de manera clara y precisa. Desarrollamos procesos y procedimientos detallados para cada etapa del proyecto, planificamos con precisión y llevamos a cabo un seguimiento riguroso para garantizar el cumplimiento de los estándares de calidad establecidos. Fomentamos la comunicación abierta y la colaboración entre todos los equipos involucrados, y realizamos un monitoreo continuo para identificar y abordar cualquier desviación de calidad de manera oportuna. Nuestra gestión integral asegura la excelencia en la calidad del software entregado, con un enfoque proactivo en todas las etapas del proyecto.

## **2.1.6 Organización**

En UrbanEase Solutions, nos comprometemos a mantener una estructura organizativa sólida y eficiente para garantizar la calidad de nuestros proyectos de software:

* **Liderazgo Ejecutivo:**
  + CEO (Director Ejecutivo)
  + COO (Director de Operaciones)
  + CTO (Director de Tecnología)
* **Equipo de Gestión de Proyectos:**
  + Gerente de Proyecto
  + Coordinador de Proyecto
  + Analista de Proyectos
* **Equipo de Desarrollo:**
  + Director de Desarrollo de Software
  + Arquitecto de Software
  + Ingeniero de Software (Frontend)
  + Ingeniero de Software (Backend)
  + Ingeniero de Software (Full-stack)
* **Equipo de Calidad y Pruebas:**
  + Gerente de Calidad del Software
  + Ingeniero de Pruebas
  + Especialista en Automatización de Pruebas
  + Analista de Control de Calidad
* **Interfaz con el Cliente:**
  + Representante de Atención al Cliente
  + Especialista en Experiencia del Usuario (UX)
  + Especialista en Soporte al Usuario
* **Colaboradores Externos:**
  + Stakeholders Externos (clientes, socios, reguladores)
  + Consultores Externos (expertos en la industria, asesores legales, auditores)

## **2.1.6 Tareas**

Nuestro equipo realiza una variedad de tareas clave para garantizar la calidad y éxito de nuestros proyectos:

* **Gestión de Proyectos:**
  + Planificación y programación del proyecto
  + Coordinación de recursos y presupuesto
  + Seguimiento del progreso y gestión de riesgos
* **Desarrollo de Software:**
  + Diseño y arquitectura de software
  + Codificación y programación
  + Integración de sistemas y pruebas unitarias
* **Control de Calidad:**
  + Establecimiento de estándares de calidad
  + Ejecución de pruebas de calidad y evaluación
  + Implementación de procesos de mejora continua
* **Interacción con el Cliente:**
  + Recopilación de requisitos y especificaciones del cliente
  + Demostración y validación del producto con el cliente
  + Soporte continuo y atención al cliente

Esta estructura organizativa y distribución de tareas refleja nuestro compromiso con la excelencia en el desarrollo de software en UrbanEase Solutions.

## **2.1.7 Roles y Responsabilidades**

## **2.1.7.1 Liderazgo Ejecutivo**

* **CEO (director ejecutivo):**
  + Define la visión y estrategia general de la empresa.
  + Toma decisiones clave relacionadas con la dirección y el crecimiento de la empresa.
  + Representa a la empresa ante los clientes, socios y la comunidad empresarial.
* **COO (director de Operaciones):**
  + Supervisa las operaciones diarias de la empresa.
  + Garantiza la eficiencia y la optimización de los procesos internos.
  + Coordina con otros líderes para implementar la estrategia empresarial.
* **CTO (director de Tecnología):**
  + Define la estrategia tecnológica de la empresa.
  + Supervisa el desarrollo y la implementación de soluciones tecnológicas.
  + Dirige la investigación y el desarrollo de nuevas tecnologías.

## **2.1.7.2 Equipo de Gestión de Proyectos**

* **Gerente de Proyecto:**
  + Planifica, organiza y supervisa la ejecución de proyectos de software.
  + Coordina con los equipos internos y externos para garantizar la entrega oportuna y dentro del presupuesto.
  + Gestiona los riesgos y problemas del proyecto y busca soluciones efectivas.
* **Coordinador de Proyecto:**
  + Asiste al gerente de proyecto en la planificación y ejecución de tareas.
  + Coordina la comunicación y la colaboración entre los equipos.
  + Realiza un seguimiento del progreso del proyecto y actualiza los informes de estado.
* **Analista de Proyectos:**
  + Recopila y analiza los requisitos del proyecto.
  + Ayuda en la elaboración de planes de proyecto detallados.
  + Realiza investigaciones y análisis para apoyar la toma de decisiones del proyecto.

## **2.1.7.3 Equipo de Desarrollo**

* **Director de Desarrollo de Software:**
  + Supervisa el equipo de desarrollo y garantiza la calidad y la eficiencia en la entrega de software.
  + Define la arquitectura y los estándares de desarrollo de software.
  + Proporciona orientación técnica y liderazgo al equipo.
* **Arquitecto de Software:**
  + Diseña la estructura y la arquitectura de las soluciones de software.
  + Define los estándares de diseño y asegura la integridad del sistema.
  + Colabora con otros equipos para garantizar la coherencia y la escalabilidad del sistema.
* **Ingeniero de Software (Frontend/Backend/Full-stack):**
  + Desarrolla aplicaciones de software según los requisitos especificados.
  + Realiza pruebas unitarias y depuración de código.
  + Colabora con otros miembros del equipo para integrar componentes de software.

## **2.1.7.4 Equipo de Calidad y Pruebas**

* **Gerente de Calidad del Software:**
  + Establece estándares y procesos de calidad para el desarrollo de software.
  + Supervisa las pruebas y la evaluación de la calidad del software.
  + Identifica áreas de mejora y propone soluciones para garantizar la calidad del producto final.
* **Ingeniero de Pruebas:**
  + Diseña y ejecuta planes de pruebas para validar la funcionalidad del software.
  + Documenta y reporta los resultados de las pruebas.
  + Colabora con el equipo de desarrollo para corregir defectos y mejorar la calidad del software.

Esta estructura de roles y responsabilidades refleja nuestro enfoque integral para garantizar la excelencia en el desarrollo de software en UrbanEase Solutions.

## **2.1.7 Documentación**

## **2.1.7.1 Propósito**

La documentación en nuestro Plan de Aseguramiento de Calidad de Software (SQAP) es clave para mantenernos organizados y asegurar la calidad en cada etapa del proyecto. Nos ayuda a establecer reglas claras y procesos bien definidos que guían a nuestro equipo en el desarrollo del software. Además, sirve como una referencia completa y comprensible para todos los involucrados, facilitando la comunicación y asegurando que todos estemos en la misma página. En pocas palabras, la documentación en nuestro SQAP es esencial para trabajar de manera eficiente, transparente y garantizar un producto final de calidad.

Principio del formulario

## **2.1.7.2 Requisitos Mínimos de Documentación**

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

## **2.1.7.2.1 Especificación de requisitos de software (SRS)**

La Especificación de Requisitos de Software (SRS) constituye un documento detallado y comprensible que establece los criterios y expectativas para el desarrollo de nuestro software en UrbanEase Solutions. En esta especificación, se describen minuciosamente las funciones, características y comportamientos esperados del software, asegurando una comprensión clara y completa de los requisitos por parte de todos los implicados en el proyecto.

La SRS se elabora con un enfoque centrado en el usuario, asegurando que las necesidades y expectativas de nuestros clientes y usuarios finales estén completamente identificadas y abordadas. Además, se establecen criterios de verificación y validación para garantizar que cada requisito sea medible, verificable y validable durante todas las etapas del ciclo de vida del desarrollo de software.  
 Este documento se presenta de manera organizada y estructurada, utilizando un lenguaje claro y accesible para facilitar su comprensión por parte de todos los miembros del equipo de desarrollo, así como de los stakeholders relevantes. La SRS se actualiza y revisa periódicamente para reflejar cualquier cambio en los requisitos del software y garantizar su alineación continua con las necesidades del cliente y los objetivos del proyecto.

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

1. INTRODUCION
   1. Objetivo
   2. Alcance
   3. Definiciones, acrónimos y abreviaciones
   4. Referencias
   5. Revisión
2. DESCRIPCION GENERAL
   1. Perspectiva del producto
   2. Funciones del producto
   3. Características de los usuarios
   4. Restricciones generales
   5. Asunciones y dependencias
3. ESPECIFICACION DE REQUERIMIENTOS
   1. Requerimiento Funcional
      1. Introducción
      2. Entradas
      3. Procesos
      4. Salidas
      5. Interfaces externas
         1. Interfaces del usuario
         2. Interfaces del hardware
         3. Interfaces del software
      6. Requerimientos de rendimiento
      7. Representación del diseño
      8. Cumplimientos con estándares
      9. Limitaciones del hardware
      10. Atributos
          1. Disponibilidad
          2. Seguridad
          3. Mantenibilidad
          4. Transferencia / Conversión
          5. Prevenciones
      11. Otros requerimientos
          1. Base de datos
          2. Operaciones
          3. Adaptaciones
4. APENDICES
5. INDICE
6. ANEXOS

## **2.1.7.2.2 Descripción del diseño del software (SDD)**

Nuestro enfoque en la SDD refleja la esencia de UrbanEase Solutions, combinando rigurosidad técnica con una visión centrada en las necesidades del usuario y la eficiencia operativa. Siguiendo los estándares ANSI/IEEE – Std 1016, nuestra SDD se convierte en un mapa detallado que traza cada componente y subcomponente del diseño del software. Desde la etapa inicial del Preliminary SDD hasta el SDD Detallado, nos aseguramos de capturar cada detalle, desde las estructuras de datos hasta las interacciones de la interfaz, garantizando así que nuestro software cumpla con las expectativas de nuestros clientes y se alinee perfectamente con la misión de UrbanEase Solutions de simplificar la vida urbana.

Principio del formulario

**ORGANIZACION DE LA SDD DENTRO DE LA VISTAS DE DISEÑO**

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

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

1. INTRODUCION
   1. Objetivo
   2. Alcance
   3. Definiciones, acrónimos y abreviaciones
2. REFERENCIAS
3. DESCRIPCION DE DESCOMPOSICION
   1. Descomposición de módulo
      1. Descripción del módulo 1
      2. Descripción del módulo 2

3.1.n. Descripción del módulo n

* 1. Descomposición de procesos concurrentes
     1. Descripción del proceso 1
     2. Descripción del proceso 2

3.2.n. Descripción del proceso n

* 1. Descomposición de datos
     1. Descripción de la entidad de datos 1
     2. Descripción de la entidad de datos 2

3.3.n. Descripción de la entidad de datos n

1. DESCRIPCION DE DEPENDENCIA

Dependencia entre módulos

Dependencia entre procesos

Dependencia entre datos

1. DESCRIPCION DE INTERFACES
   1. Interfaces de módulo
      1. Descripción del módulo 1
      2. Descripción del módulo 2

5.1.n. Descripción del módulo n

* 1. Interfaces de procesos
     1. Descripción del proceso 1
     2. Descripción del proceso 2

5.2.n. Descripción del proceso n

1. DISEÑO DETALLADO
   1. Diseño detallado del módulo
      1. Detalle del módulo 1
      2. Detalle del módulo 2

6.1.n. Detalle del módulo n

* 1. Diseño detallado de datos
     1. Detalle de entidad de datos 1
     2. Detalle de entidad de datos 2

6.2.n. Detalle de entidad de datos n

1. APENDICES
2. INDICE
3. ANEXOS

## **2.1.7.2.3 Plan de verificación y validación de software (PVVS)**

Para el Plan de Verificación y Validación de Software (PVVS) de UrbanEase Solutions, nos enfocamos en asegurar la calidad y el rendimiento óptimo de nuestro software. Este plan detalla los métodos y procedimientos que emplearemos para garantizar que nuestro software cumpla con los estándares de calidad y las expectativas de nuestros clientes.

Nuestro PVVS se centra en:

* Validar que los requisitos definidos en la Especificación de Requisitos de Software (SRS) sean aprobados por las partes pertinentes de UrbanEase Solutions, asegurando así su alineación con las necesidades y expectativas del cliente.
* Verificar que los requisitos especificados en la SRS se implementen eficazmente en el diseño detallado del software, asegurando una coherencia completa entre las especificaciones y la arquitectura del sistema.
* Asegurar que el diseño expresado en la Descripción del Diseño del Software (SDD) se traduzca de manera precisa y completa en el código fuente del software, garantizando su correcta implementación.
* Validar que el código del software, una vez ejecutado, cumpla con los requisitos y funcionalidades establecidos en la SRS, asegurando su correcto funcionamiento y rendimiento.

Nuestro enfoque en el PVVS se basa en los estándares y prácticas recomendadas de la industria, adaptadas específicamente a las necesidades y procesos de UrbanEase Solutions, garantizando así un proceso riguroso de verificación y validación que asegure la calidad y la conformidad de nuestro software. Ademas se basa en los estándares ANSI/IEEE Std 829-1983, ANSI/IEEE Std 10081987 y ANSI/IEEE Std 1012-1986 para garantizar un proceso riguroso de verificación y validación que asegure la calidad y la conformidad.

**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

**CICLO DE VIDA DE VERIFICACION Y VALIDACION EL PLAN**

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

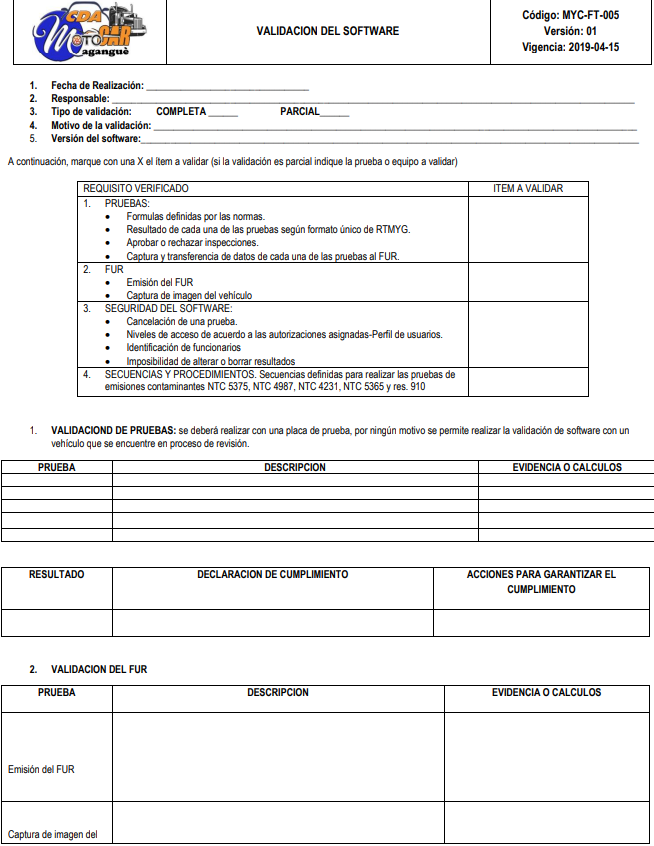
|  |  |  |
| --- | --- | --- |
| Procesos | Actividades | Tareas |
| Proceso de Gestión  En UrbanEase Solutions, el proceso de gestión de verificación y validación (V&V) se enfoca en coordinar y controlar de manera efectiva todas las actividades relacionadas con la calidad del software. | * Planificación del PVVS. * Establecimiento de los criterios de verificación y validación. * Asignación de recursos y responsabilidades. * Monitoreo y control del proceso de verificación y validación. | * Definir los objetivos y alcance del PVVS. * Establecer el calendario y los hitos de verificación y validación. * Identificar y asignar recursos humanos y materiales necesarios. * Supervisar y reportar el progreso del PVVS. |
| Proceso de Adquisición  Este proceso se centra en la selección y adquisición de componentes de software externos que cumplen con los requisitos de calidad establecidos por UrbanEase Solutions. | * Identificación de proveedores potenciales. * Evaluación de las capacidades y la calidad del software de los proveedores. * Selección de los componentes de software adecuados. | * Realizar análisis de mercado para identificar proveedores potenciales. * Evaluar la calidad del software mediante pruebas de demostración o evaluaciones técnicas. * Negociar términos de contrato y acuerdos de nivel de servicio (SLA). |
| Proceso de Suministro:  Este proceso se enfoca en la entrega e integración de los componentes de software adquiridos en el sistema de UrbanEase Solutions. | * Recepción y verificación de los componentes de software entregados. * Integración de los componentes en el sistema existente. * Pruebas de integración y aceptación del sistema. | * Verificar la conformidad de los componentes de software con los requisitos especificados. * Integrar los componentes en el sistema de UrbanEase Solutions. * Realizar pruebas de integración para garantizar la funcionalidad y compatibilidad del sistema. |
| Proceso de Desarrollo  Este proceso se centra en la creación y construcción de software nuevo o personalizado para satisfacer las necesidades específicas de UrbanEase Solutions. | * Diseño y desarrollo del software. * Implementación y codificación. * Pruebas unitarias y de integración. | * Elaborar diseños detallados del software según los requisitos establecidos. * Codificar el software utilizando las mejores prácticas y estándares de codificación. * Realizar pruebas unitarias para validar el comportamiento de cada componente individual. |
| Proceso de Mantenimiento  Este proceso se centra en la corrección de errores, la implementación de mejoras y la actualización continua del software para garantizar su eficacia y relevancia a lo largo del tiempo. | * Identificación y priorización de problemas. * Desarrollo y prueba de parches y actualizaciones. * Despliegue de parches y actualizaciones. | * Recopilar y analizar informes de errores y retroalimentación del usuario. * Desarrollar y probar parches y actualizaciones para corregir problemas y agregar nuevas características. * Desplegar parches y actualizaciones de manera controlada para minimizar el impacto en la operación del software. |

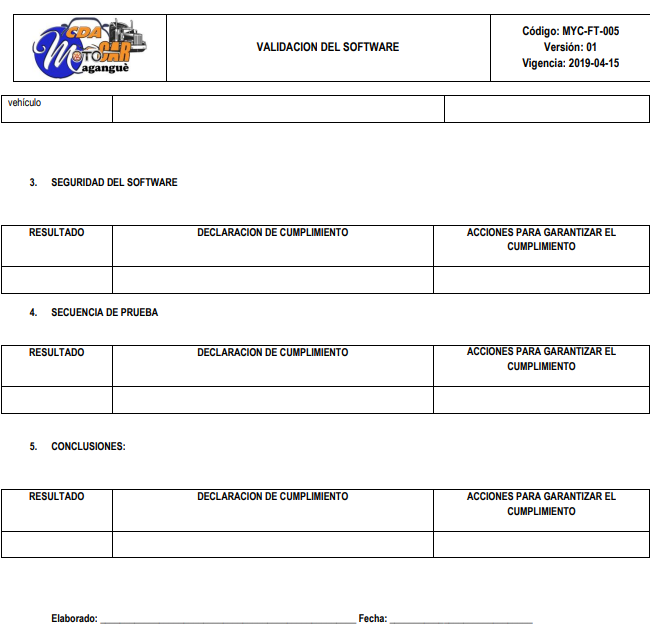
## **2.1.7.2.4 Informe de verificación y validación de software (IVVS)**

En UrbanEase Solutions, el Informe de Verificación y Validación de Software (IVVS) es una herramienta clave para garantizar la calidad y fiabilidad de nuestros productos. Este informe detalla los resultados de las actividades de verificación y validación realizadas durante el ciclo de vida del desarrollo de software.

El IVVS proporciona una visión integral de las pruebas realizadas, incluyendo pruebas de unidad, integración, sistema y aceptación. Además, documenta cualquier problema encontrado durante estas pruebas y las acciones correctivas tomadas para abordarlos.

Nuestro enfoque se basa en estándares reconocidos y mejores prácticas de la industria, adaptados específicamente a las necesidades y requisitos de UrbanEase Solutions. El IVVS sirve como evidencia de que nuestro software cumple con los estándares de calidad establecidos y está listo para ser entregado a nuestros clientes.





## **2.1.7.2.5 Documentación del Usuario (UD)**

En UrbanEase Solutions, la Documentación del Usuario (UD) es un componente esencial para asegurar la satisfacción y comprensión del usuario final sobre nuestros productos de software. Esta documentación se diseña de manera clara y concisa, utilizando un lenguaje accesible para todos los niveles de usuarios.

Nuestro enfoque se centra en proporcionar instrucciones detalladas sobre cómo utilizar el software de manera efectiva, incluyendo tutoriales paso a paso, guías de usuario y preguntas frecuentes. Además, nos aseguramos de que la documentación refleje con precisión las funcionalidades y características del software, garantizando una experiencia de usuario óptima.

Entendemos la importancia de la documentación del usuario en la adopción y uso continuo de nuestro software, por lo que nos esforzamos por ofrecer contenido informativo y fácil de seguir que empodere a nuestros usuarios para sacar el máximo provecho de nuestras soluciones tecnológicas.

**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

* 1. Descripción del uso de documentos
  2. Documentos relacionados
  3. Convenciones
     1. Símbolos
     2. Convenciones de estilo
     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

* + 1. Alcance
    2. Materiales
    3. Preparaciones
    4. Precauciones y prevenciones
    5. Métodos
    6. Información relacionada

7.2. Cuerpo del documento en modo de referencia

* + 1. Objetivo
    2. Materiales
    3. Preparaciones
    4. Entradas
    5. Precauciones y prevenciones
    6. Invocación
    7. Operaciones de suspensión
    8. Operaciones de terminación
    9. Salidas
    10. Condiciones de error
    11. Información relacionada

MENSAJES DE ERROR, CONOCIMIENTO DE PROBLEMAS, RECUPERACION DE ERROR

ANEXOS

BIBLIOGRAFIA

GLOSARIO

INDICE

## **2.1.7.2.6 Plan de gestión de configuración del software (SCMP)**

El Plan de Gestión de Configuración de Software (SCMP) de UrbanEase Solutions 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 basa en estándares reconocidos en la industria y tiene como objetivo principal garantizar la integridad, trazabilidad y control de la configuración del software para nuestros proyectos.

**Procesos de Configuración:**

* **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, relacionados con los proyectos de UrbanEase Solutions.

* **Control de Cambios:**

Se establecerá un proceso para la solicitud, revisión, aprobación y seguimiento de cambios en los elementos de configuración del software, adaptado a las necesidades y requisitos de UrbanEase Solutions.

* **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, en consonancia con los estándares de UrbanEase Solutions.

* **Registro y Reporte:**
* Se llevará un registro detallado de todos los cambios realizados, incluyendo su estado y fecha de implementación, siguiendo las directrices establecidas por UrbanEase Solutions.
* Se generará un informe de cambios periódico para su revisión, con el fin de mantener a todas las partes interesadas informadas sobre el estado y evolución de la configuración del software.

Este SCMP asegura que la gestión de configuración del software en UrbanEase Solutions se realice de manera eficiente y efectiva, garantizando la calidad y consistencia en nuestros proyectos.

## **2.1.9 Estándares, Practicas y Convenciones**

## **2.1.9.1 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

## **2.1.9.2 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

## **2.1.9.3 Estándar de verificar el cumplimiento**

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

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

## **2.1.10 Revisión 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.

## **2.1.10.1 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.

## **2.1.10.2 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.

## **2.1.10.3 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.

## **2.1.10.4 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.

## **2.1.10.4 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

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

## **2.1.11 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.

## **2.1.12 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.

## **2.1.13 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

## **2.1.14 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

## **2.1.15 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.

## **2.1.16 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

## **2.1.17 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.