**PAPUSCORP**

**Papu Software Corporation**

Estándares de desarrollo y mantenimiento.

Integrantes:

Paola Alejandra Novelo Chi

Diego Alfonso Burgos Tzuc

Juan Osvaldo Salazar Puerto

Francisco Joaquin Ruiz Ayala

Jose Armando Aviles Lopez

Contenido

[1. Propósito 2](#_Toc135871685)

[2. Documentos de referencia 2](#_Toc135871686)

[3. Gestión 3](#_Toc135871687)

[3.1 Organización 3](#_Toc135871688)

[3.2 Tareas 4](#_Toc135871689)

[3.3 Roles y responsabilidades 5](#_Toc135871690)

[4. Documentación 5](#_Toc135871691)

[El registro de rastreo describe la forma en la cual todos los tipos de entregable se manejan, asegurando una experiencia eficiente al momento de buscar cierto documento que se necesite en cualquier momento, se tendrá como base el **standard CMMI-DEV 1.3** y se realizaran auditorias periódicamente para tener en correcto orden todos los documentos que sean parte de este registro. 7](#_Toc135871692)

[5. Estándares, prácticas, convenciones y métricas 8](#_Toc135871693)

[6. Revisiones de Software 10](#_Toc135871694)

[7. Pruebas 13](#_Toc135871695)

[8. Reportes de problemas y acciones correctivas 13](#_Toc135871696)

[9. Herramientas, técnicas y metodologías. 13](#_Toc135871697)

[9.1 Herramientas 13](#_Toc135871698)

[9.2 Técnicas 16](#_Toc135871699)

[9.3 Metodologías. 16](#_Toc135871700)

[10. Control de medios. 17](#_Toc135871701)

[10.1 Acceso no autorizado 17](#_Toc135871702)

[10.2 Daños o degradación involuntarios 17](#_Toc135871703)

[11. Control de código. 18](#_Toc135871704)

[12. Recopilación, mantenimiento y retención de registros. 18](#_Toc135871705)

[12.1 Recopilación de registros 18](#_Toc135871706)

[12.2 Mantenimiento de registros 19](#_Toc135871707)

[12.3 Conservación de registros 20](#_Toc135871708)

[13. Entrenamiento 21](#_Toc135871709)

[14. Gestión de riesgos 22](#_Toc135871710)

[15. Glosario 23](#_Toc135871711)

[15.1 Acronimos 24](#_Toc135871712)

[16. Procedimiento e historial de cambio de SQAP 25](#_Toc135871713)

control de la configuración.

|  |  |  |
| --- | --- | --- |
| Nombre | Versión | Descripción |
| Plan de aseguramiento de la calidad | 1.0 | Versión inicial del plan de aseguramiento |

Link del GitHub:

<https://github.com/ThuPapuha/PlanSQA_PapusCorp>

Plan de aseguramiento de la calidad del software

# 1. Propósito

El presente documento describe un plan de aseguramiento de la calidad para el software “Calculadora de matrices”, el cual como su nombre indica, es una calculadora diseñada para realizar operaciones con matrices. Este software es parte de un proyecto escolar y se espera que sea utilizado por los estudiantes para facilitar el cálculo y manipulación de matrices.

El objetivo de este documento es desarrollar un plan de aseguramiento de calidad (PSQA) para el proyecto “Calculadora de matrices”. Este plan se centrará específicamente en la fase de diseño del software. Para lograr este objetivo, se seguirá el estándar IEEE 730, que establece las directrices para la planificación del PSQA en todas las fases del ciclo de vida del software.

Se identificarán los objetivos específicos del proyecto y se definirán los criterios de calidad para la fase de diseño del software. A continuación, se identificarán los estándares y prácticas recomendadas para el diseño de software. Se definirá un proceso de revisión para asegurarse de que los documentos de diseño cumplan con los requisitos de calidad identificados.

Además, se establecerán métricas de calidad para evaluar el desempeño del proceso de diseño y se asignan responsabilidades claras para el aseguramiento de calidad. Estas responsabilidades incluirán la designación de un arquitecto de software o desarrollador experimentado como responsable de la revisión de diseño.

Este documento es consistente con el estándar de desarrollo y mantenimiento de software anexado, y es consistente con los estándares que se encuentran en ese documento.

# 2. Documentos de referencia

* [730-2002 Software Quality Assurance Plans](https://extranet.matematicas.uady.mx/enlinea/pluginfile.php/13283/mod_folder/content/0/730-2002%20Software%20Quality%20Assurance%20Plans.pdf?forcedownload=1).
* [730-2014 Software Quality Assurance Processes](https://extranet.matematicas.uady.mx/enlinea/pluginfile.php/13283/mod_folder/content/0/730-2014%20Software%20Quality%20Assurance%20Processes.pdf?forcedownload=1).
* [7301-1995 Guide for Software Quality Assurance Planning](https://extranet.matematicas.uady.mx/enlinea/pluginfile.php/13283/mod_folder/content/0/7301-1995%20%20Guide%20for%20Software%20Quality%20Assurance%20Planning.pdf?forcedownload=1).
* ISO\_IEC\_15939-2007 Systems and software engineering — Measurement process.
* ISO\_9001.
* ISO/IEC 12207:2017.
* IEEE Std 828-2012.
* PMBOKGuideFourthEdition.

# 3. Gestión

El plan se ha desarrollado teniendo en cuenta las características y necesidades específicas del proyecto y se ha diseñado para garantizar la integridad y la trazabilidad de los elementos de configuración a lo largo de todo el ciclo de vida del software. Además, se han establecido procedimientos y herramientas para la gestión de versiones, la gestión de cambios, la auditoría y la retroalimentación.

## 3.1 Organización

Organigrama jerárquico

Diagrama

Descripción generada automáticamente

Para el proyecto “calculadora de matrices”, se planea aplicar una metodología de análisis y diseño que asegure la calidad del software desde las etapas iniciales. Se definirán los criterios de aceptación y validación y se diseñará la arquitectura y la estructura del software, utilizando diagramas y modelos que faciliten su comprensión y verificación. Se realizarán revisiones periódicas para detectar y corregir posibles errores o inconsistencias. Se establecerán mecanismos de comunicación y coordinación entre los miembros del equipo para garantizar la coherencia y la integración del software.

En cuanto a la organización encargada de preparar y mantener el plan SQA, se ha identificado al equipo de diseño y al gerente de proyecto, quienes serán responsables de asegurar que el Plan SQA se mantenga actualizado y de que se realicen las revisiones necesarias para garantizar la calidad del producto de software.

## 3.2 Tareas

1. Distribuir tareas a los miembros del equipo según su rol, siguiendo el Plan de Desarrollo actual.
2. Documentar o modificar el Análisis y Diseño:

* Analizar la Especificación de Requerimientos y describir la estructura interna del sistema y su descomposición en subsistemas y componentes.
* Detallar la apariencia y comportamiento de la interfaz.
* Describir los componentes de manera que sean fácilmente construibles.
* Generar o actualizar el Análisis y Diseño.
* Generar o modificar el Registro de Rastreo.

1. Verificar el Análisis y Diseño y el Registro de Rastreo utilizando plantillas y checklists.
2. Corregir los defectos encontrados en el Análisis y Diseño y el Registro de Rastreo basándose en el Reporte de Verificación y obtener la aprobación.
3. Validar el Análisis y Diseño utilizando una plantilla basada en el estándar ISO 9001.
4. Corregir los defectos encontrados en el Análisis y Diseño basándose en el Reporte de Validación y obtener la aprobación. Aplicar acciones correctivas y preventivas según el estándar ISO/IEC 12207:2017.
5. Elaborar o modificar el Plan de Pruebas de Integración siguiendo el estándar ISO/IEC 12207 y utilizando un formato similar a los otros archivos.
6. Verificar el Plan de Pruebas de Integración y aplicar acciones correctivas y preventivas según el estándar ISO/IEC 12207:2017.
7. Corregir los defectos encontrados en el Plan de Pruebas de Integración basándose en el Reporte de Verificación y obtener la aprobación. Aplicar acciones correctivas y preventivas según el estándar ISO/IEC 12207:2017.
8. Incorporar Análisis y Diseño, Registro de Rastreo y Plan de Pruebas de Integración como líneas base en la Configuración de Software siguiendo el estándar IEEE Std 828-2012. Realizar la identificación de elementos, definir la estructura, establecer el proceso de aprobación, identificar herramientas y realizar la integración de los elementos.
9. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsables por actividad y mediciones requeridas.
10. Elaborar el Reporte del presupuesto de Inputs, outputs y técnicas para delimitar el presupuesto total del proyecto, siguiendo la guía del PMI.
11. Gestionar las actividades del proyecto de análisis y diseño con un cronograma basado en el estándar PMBOK. Gestionar los riesgos utilizando una plantilla basada en el mismo estándar.

## 3.3 Roles y responsabilidades

**Líder de diseño:** responsable de liderar y supervisar el equipo de diseño, asegurando que se cumplan los plazos y los estándares de calidad del diseño.

**Diseñadores de software:** responsables de crear y revisar los documentos de diseño, asegurando que cumplan con las especificaciones del cliente y los estándares de calidad establecidos.

**Revisor de calidad del diseño:** responsable de verificar que los diseños cumplan con los criterios de calidad establecidos, identificar y rastrear los problemas de diseño y trabajar con el equipo de diseño para resolverlos de manera oportuna y efectiva.

**Comité de SQA:** responsable de revisar y aprobar los planes de diseño y de realizar revisiones técnicas de los documentos de diseño para asegurar que cumplan con los estándares de calidad y las especificaciones del cliente.

**Gerente de proyecto:** responsable de coordinar y supervisar el proyecto de software en su conjunto, asegurando que se cumplan los plazos, presupuestos y objetivos de calidad establecidos.

**Equipo de pruebas:** responsable de realizar pruebas de diseño para garantizar la calidad del software y asegurar que cumpla con las especificaciones del cliente.

# 4. Documentación

La documentación es una herramienta la cual beneficia de forma directa si usada correctamente, y todo plan de software necesita un mínimo de documentos para su correcto desarrollo, esto no solo para tener una relación de los documentos desarrollados durante el proceso de desarrollo, sino también para poder tener una retrospectiva de los cambios, decisiones, y crítica que se pueda desarrollar en base a el seguimiento o falta del mismo, creando una forma concreta para poder verificar los fallos o triunfos del equipo de desarrollo y el software desarrollado.

La documentación se puede encontrar en los documentos referenciados, en el capítulo 5:

#### Software Requirements Specifications (SRS)

La especificación de los requerimientos debe tener los requerimientos tratados con el desarrollador y el cliente que lo requiere, este normalmente está desarrollado y deriva de varios documentos y contiene una lista de referencia que identifica a los modelos en los que se estén trabajando tanto prototipos como productos, este debe identificar a los documentos que van a ser usados y sean de mayor importancia para el desarrollo y debe ayudar a identificar cuando estos documentos contengan requerimientos contradictorios o no plausibles para el desarrollo en el que se encuentra.

#### Software Design Description (SDD)

Esta es una descripcion tecnica de como el software cumplira los requerimientos a los que se comprometen en el SRS, su función más fundamental es la descomposición de todo el sistema en componentes completos, documentan también las razones por las cuales, estas decisiones son las más sensatas para facilitar el entendimiento de la estructura del sistema.

Este documento debe describir las funciones mayores dentro del sistema, como lo pueden ser las bases de datos, interfaces tanto externas como internas y la estructura general del diseño, implica describir el ambiente de operación, procesamiento, modelamiento, simulación, etc.

#### User documentation

La sección de documentación de usuarios se concentra en el archivo de todos los medios por los cuales se le describe al usuario el uso operacional de la pieza de software, tanto los métodos como instrucciones como muestras de todo la metodología de uso para el usuario, esto con el conocimiento de que existen varios tipos de usuarios o anuales que necesitan un método diferente para que se les sea comunicado esta información de manera oportuna.

#### Software Configuration Management Plan (SCMP)

El plan del manejo de la configuración describe las tareas, metodologías y herramientas que aseguran el control adecuado de procedimientos y los controles que son documentados e implementados de manera correcta, si este no es un documento por sí mismo, deberá formar parte del SQAP, y si no,no es necesario que el elemento de la organización de SQA prepare el SCMP, Este documento será hecho conforme al **standard 828-1983 para planes de configuración de software**

1. Documento de Análisis y diseño

El documento de análisis y diseño define todas las decisiones en cuanto al diseño del producto de software, teniendo como objetivo la uniformidad en cuanto al todas las formas en las cuales se desarrolla el software y trata de evitar decisiones con poca razón de ser, **basado en anexo G de J-STD-016-1995**, será evaluado con el anexo K y L del mismo standard y se co-evaluara en Mettings definidas en el anexo N.

1. Registro de Rastreo

### El registro de rastreo describe la forma en la cual todos los tipos de entregable se manejan, asegurando una experiencia eficiente al momento de buscar cierto documento que se necesite en cualquier momento, se tendrá como base el **standard CMMI-DEV 1.3** y se realizaran auditorias periódicamente para tener en correcto orden todos los documentos que sean parte de este registro.

### 

# 5. Estándares, prácticas, convenciones y métricas

Los estándares que se serán utilizados son los siguientes:

|  |  |
| --- | --- |
| **Gestión de la configuración de software** | |
| IEEE Std 828-2012 (Identificación de la configuración) | Establece las prácticas recomendadas para la identificación y gestión de la configuración en proyectos de desarrollo de sistemas y software. |
| ISO/IEC 12207 (Control de versiones) | Este estándar proporciona un marco de trabajo para el desarrollo, mantenimiento y suministro de software. |
| CMMI-DEV, V1.3 (Modelo de Madurez de Capacidad Integrado para el Desarrollo) | Proporciona un marco de referencia para evaluar y mejorar la capacidad de una organización en términos de sus procesos de desarrollo de software. |
| **Plantillas** | |
| J-STD-016-1995 (Software Development and Documentation) | Establece pautas para el desarrollo de software y la documentación relacionada, incluyendo los requisitos, el diseño, las pruebas y la gestión del ciclo de vida del software. |
| **Corrección de los defectos encontrados** | |
| ISO/IEC 12207:2017 (Identificación y corrección de defectos encontrados) | Es una norma internacional que establece los procesos del ciclo de vida del software.  Aborda varios aspectos del ciclo de vida del software, incluyendo la identificación y corrección de defectos |
| **Documentos de control** | |
| IEEE Std 1063 (Documentación del usuario) | Aborda aspectos como la estructura y organización de la documentación, el estilo de escritura, la presentación de la información, los requisitos de usabilidad y accesibilidad, la integración de ejemplos y casos de uso, y la documentación para diferentes audiencias y niveles de habilidad. |
| ISO 9001 (Documentos de Control) | Establece que una organización debe establecer, implementar y mantener la documentación necesaria para asegurar la planificación, operación y control efectivos de sus procesos.  Esto implica la necesidad de documentar información relevante para el sistema de gestión de calidad, como políticas, objetivos, procedimientos, instrucciones de trabajo, registros y otros documentos necesarios para el funcionamiento eficaz de la organización. |
| **Revisiones** | |
| IEEE 1028 (IEEE Standard for Software Reviews and Audits) | Este estándar establece las pautas para llevar a cabo revisiones y auditorías de software en el proceso de desarrollo.  Proporciona directrices detalladas sobre la planificación, preparación, realización y documentación de revisiones y auditorías de software, con el objetivo de mejorar la calidad y la eficacia del software desarrollado. |
| **Métricas** | |
| IEEE 1061 (IEEE Standard for Software Quality Metrics Methodology) | Este estándar establece una metodología para la medición y evaluación de la calidad del software utilizando métricas específicas. |
| ISO/IEC 9126 (Software Engineering - Product Quality) | Define un conjunto de características y subcaracterísticas que se utilizan para evaluar la calidad de un producto de software. Estas características incluyen la funcionalidad, la confiabilidad, la usabilidad, la eficiencia, la mantenibilidad y la portabilidad del software.    Además de las características, el estándar también proporciona una guía sobre cómo medir y evaluar la calidad del software, incluyendo la definición de métricas y la realización de evaluaciones. |
| ISO IEC 15939 (Systems and Software Engineering - Measurement Process) | Es proporcionar orientación sobre cómo medir y evaluar diversos aspectos del desarrollo de sistemas y software, como el desempeño, la calidad, la productividad y el cumplimiento de los requisitos. |
| **Control del progreso del proyecto y costos de calidad** | |
| IPMBOK (Project Management Body of Knowledge) | Cubre áreas clave de conocimiento en la gestión de proyectos, como la gestión del alcance, el cronograma, los costos, la calidad, los recursos humanos, la comunicación, los riesgos y las adquisiciones, entre otros.  Proporciona una descripción detallada de los procesos, las entradas, las salidas, las herramientas y las técnicas utilizadas en cada una de estas áreas de conocimiento. |
| **Aseguramiento de la calidad de la contribución de participantes externos** | |
| IEEE 12207 (IEEE Standard for Systems and Software Engineering - Software Life Cycle Processes) | Establece un marco común y estructurado de procesos, lo que facilita la comunicación y la colaboración entre los equipos de desarrollo.  Además, ayuda a garantizar la calidad y la consistencia en los productos de software. |
| **Pruebas de software** | |
| IEEE 829 (IEEE Standard for Software and System Test Documentation) | Establece los requisitos para la documentación de pruebas de software y sistemas.  Proporciona pautas detalladas sobre los diferentes tipos de documentos que se deben crear durante el proceso de pruebas. |
| **Planes referenciados** | |
| Plan de gestión de medios | El objetivo principal del plan de gestión de medios es asegurar que todos los recursos y activos digitales utilizados durante el proceso de diseño sean gestionados de manera adecuada, garantizando su disponibilidad, integridad y confidencialidad. Esto incluye tanto los medios informáticos empleados, como los resultados generados, tales como documentos, imágenes, archivos de diseño y otros elementos relacionados. |
| Plan de capacitación | Establece las pautas para llevar a gestión de capacitación del personal que utilizara este plan de SQA. |
| Plan de gestión de riesgos | Establece las pautas para llevar a gestión de acciones preventivas y correctivas. |

# 6. Revisiones de Software

Las revisiones que se llevarán a cabo en este proyecto son las siguientes:

**Verificaciones y validaciones**

1. **A2.3** Verificar la Especificación de Requerimientos

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Producto** | **Descripción** |
| Verificar la Especificación de Requerimientos | Especificación de Requerimientos | Verificar la claridad de redacción de la especificación de requerimientos y su consistencia con la descripción del producto y con el estándar de documentación requerido en el proceso específico. Adicionalmente revisar que los requerimientos sean completos y no ambiguos o contradictorios. Los defectos encontrados se documentan en un reporte de verificación. |
| Validar la Especificación de Requerimientos | Especificación de Requerimientos | Validar que la especificación de requerimientos cumple con las necesidades y expectativas acordadas, incluyendo la realización de la prueba de usabilidad de la interfaz del usuario. Los defectos encontrados se documentan en un reporte de validación. |
| Verificar el Plan de Pruebas de Sistema | Plan de Pruebas de Sistema | Verificar consistencia del plan de pruebas de sistema con la especificación de requerimientos y con el estándar de documentación requerido en el proceso específico. Los defectos encontrados se documentan en un reporte de verificación. |
| Verificar el Manual de Usuario | Manual de Usuario | Verificar consistencia del manual de usuario con la especificación de requerimientos y con el estándar de documentación requerido en el proceso específico. Los defectos encontrados se documentan en un reporte de verificación. |
| Verificar el Análisis y Diseño y el Registro de Rastreo. | Análisis y Diseño Registro de Rastreo | Verificar claridad de la documentación del análisis y diseño, su factibilidad y la consistencia con la especificación de requerimientos y con el estándar de documentación requerido en el proceso específico. verificar que el registro de rastreo contenga las relaciones adecuadas entre los requerimientos y los elementos de análisis y diseño. los defectos encontrados se documentan en un reporte de verificación.  esto se hará mediante el uso de la plantilla y los checklists del anexo L del J-STD-016-1995, en específico los productos de software 7 a 12. |
| Validar el Análisis y Diseño | Análisis y Diseño | Validar que el análisis y diseño cumple con las necesidades y expectativas acordadas con el cliente. Los defectos encontrados se documentan en un reporte de validación.  Para los documentos de Validación se empleará una plantilla basada en el formado dicho en el estándar ISO 9001. |
| Verificar el Plan de Pruebas de Integración | Plan de Pruebas de Integración | Verificar consistencia del plan de pruebas de integración con el análisis y diseño y con el estándar de documentación requerido en el proceso específico. Los defectos encontrados se documentan en un reporte de verificación.  El estandar ISO/IEC 12207:2017 nos proporcionará referencias para la aplicación de las acciones correctivas y preventivas. |
| Verificar el Registro de Rastreo | Registro de Rastreo | Verificar que el registro de rastreo contenga las relaciones adecuadas entre los elementos de análisis y diseño y los componentes. los defectos encontrados se documentan en un reporte de verificación. |
| Verificar el Manual de Operación | Manual de Operación | Verificar consistencia del manual de operación con el software y con el estándar de documentación requerido en el proceso específico. Los defectos encontrados se documentan en un reporte de verificación. |
| Verificar el Manual de Usuario | Manual de Usuario | Verificar consistencia del manual de usuario con el sistema de software y con el estándar de documentación requerido en el proceso específico. Los defectos encontrados se documentan en un reporte de verificación. |
| Verificar el Manual de Mantenimiento | Manual de Mantenimiento | Verificar consistencia del manual de mantenimiento con la configuración de software y con el estándar de documentación requerido en el proceso específico. Los defectos encontrados se documentan en un reporte de verificación. |

# 7. Pruebas

**Esta sección no es relevante en el contexto de este plan**, ya que se centra en un proceso de diseño de software en el cual no se generará código en esta etapa.

# 8. Reportes de problemas y acciones correctivas

En caso de requerir la implementación de acciones preventivas y correctivas, se hará uso de nuestro plan de acciones preventivas y correctivas, el cual se basa en el estándar ISO 9001:2015. Este enfoque nos permite abordar de manera sistemática y efectiva las necesidades de mejora continua y la gestión de la calidad en nuestra organización. Al seguir las directrices establecidas por el estándar ISO 9001:2015, nos aseguramos de contar con un marco de trabajo reconocido internacionalmente para identificar, analizar y abordar los riesgos, así como implementar acciones correctivas y preventivas pertinentes. Este plan se adapta a nuestras necesidades y contexto específico, permitiéndonos gestionar eficientemente los desafíos y mejorar nuestros procesos y resultados en busca de la excelencia organizacional.

# 9. Herramientas, técnicas y metodologías.

En este apartado se identifican las herramientas, técnicas y metodologías que se utilizarán para respaldar el aseguramiento de la calidad del software. Debe enumerar o hacer referencia a las herramientas, técnicas y metodologías que están disponibles y aquellas que necesitan ser adquiridas o desarrolladas.

## 9.1 Herramientas

Durante la fase de diseño del proyecto, además de las utilidades del sistema operativo, los documentos de ayuda y las plantillas adoptadas del modelo de procesos Papuscorp, será necesario emplear las siguientes herramientas:

|  |  |
| --- | --- |
|  | Tabla de Herramientas |
| Herramienta | Descripción |
| Procesador de texto: Word | Microsoft Word es una aplicación de procesamiento de texto desarrollada por Microsoft y viene integrada por defecto en el paquete de software Microsoft Office. En el contexto de este proyecto, se utilizará Microsoft Word para la creación y edición de los documentos resultantes, así como para la elaboración de plantillas que serán empleadas durante el proceso de diseño. |
| Herramienta de diagramas:  Lucidchart  UMLDesigner | Lucidchart es una herramienta en línea que facilita la creación de diagramas de diversa índole, como diagramas de flujo, diagramas de clases, diagramas de secuencia y muchos otros. Su interfaz intuitiva y su amplia gama de formas y plantillas permiten representar visualmente los conceptos y relaciones involucrados en el diseño de software. Lucidchart también facilita la colaboración en tiempo real, lo que favorece la participación de equipos distribuidos geográficamente.  UMLDesigner: UMLDesigner es una herramienta específicamente diseñada para el modelado y diseño de software utilizando el lenguaje de modelado unificado (UML). Proporciona una amplia variedad de elementos y herramientas para crear diagramas UML, como diagramas de casos de uso, diagramas de clases, diagramas de actividad y más. UMLDesigner permite representar de manera precisa y detallada la estructura y comportamiento del software en desarrollo.  El uso de ambas herramientas brinda a los equipos de diseño de software la capacidad de representar visualmente los diferentes aspectos del diseño, lo que facilita la comunicación, la comprensión y la documentación del software en proceso de desarrollo. |
| Herramientas de Apoyo:  -Internet  -Photoshop | El acceso a Internet desempeñará un papel crucial como nuestra principal fuente de consulta y referencia. Nos permitirá acceder a recursos en línea, documentación técnica, tutoriales, foros y otros recursos relevantes. La capacidad de buscar información y estar al tanto de las últimas tendencias, tecnologías y mejores prácticas en el diseño de software contribuirá a enriquecer y mejorar nuestros diseños.  Por otro lado, Photoshop, una aplicación de edición y manipulación de imágenes ampliamente utilizada, nos permitirá dar una apariencia visualmente atractiva a nuestros diseños. Con sus diversas herramientas y capacidades, podremos retocar, crear gráficos, ajustar colores y efectos, y optimizar las imágenes utilizadas en nuestros diseños de software. Esto asegurará que los elementos visuales sean atractivos, coherentes y en línea con las expectativas de los usuarios.  En resumen, el acceso a Internet nos brindará recursos y conocimientos actualizados, mientras que Photoshop nos permitirá crear una presentación visualmente atractiva para nuestros diseños de software. Ambas herramientas serán indispensables para garantizar la calidad y la estética de nuestros productos de software. |

## 9.2 Técnicas

Dentro de nuestro plan de aseguramiento de la calidad del software, hacemos uso de diversos estándares que se encuentran detallados en la sección 5 de dicho documento. Estos estándares se aplican con diferentes propósitos a lo largo de nuestro proceso de diseño, abarcando áreas como la gestión de versiones, el uso de plantillas, las revisiones de documentos, el control de cambios, la medición de métricas, entre otros.

Estos estándares nos proporcionan directrices y mejores prácticas establecidas por la industria para asegurar la calidad y la eficiencia en nuestro trabajo de diseño de software. A través de su implementación, buscamos garantizar la consistencia, la estandarización y el cumplimiento de requisitos en todas las etapas del proceso.

Al seguir estos estándares, nos beneficiamos de procesos más efectivos y una mayor calidad en nuestros productos de software. Nos ayudan a establecer una base sólida para la gestión del diseño, facilitan la colaboración entre los equipos involucrados y nos permiten tomar decisiones informadas basadas en prácticas reconocidas y probadas en la industria.

En resumen, el uso de los estándares identificados en la sección 5 de nuestro plan de aseguramiento de la calidad del software nos proporciona una guía valiosa para el diseño de software de calidad, abarcando aspectos como la gestión de versiones, el uso de plantillas, las revisiones de documentos, el control de cambios y la medición de métricas, entre otros. Estos estándares nos orientan y nos ayudan a mantener altos estándares de calidad en todo nuestro proceso de diseño.

## 9.3 Metodologías.

La descripción detallada sobre cómo se emplearán las diferentes técnicas y herramientas en cada fase del proceso de diseño se encuentra disponible en el documento de modelos de procesos de PapusCorp. Este documento se puede acceder a través del siguiente enlace: https://docs.google.com/document/d/1pBAa4nM1Mc7EBJ7Nu7N8k1QHUM4QWqom/edit.

Dicho documento proporciona una guía exhaustiva que detalla la aplicación de diversas técnicas y herramientas en cada etapa del proceso de diseño. Estas técnicas y herramientas se han seleccionado cuidadosamente para maximizar la eficiencia y calidad del diseño de software. De igual manera podrá encontrar información detallada sobre la implementación de las técnicas recomendadas en cada fase específica del proceso de diseño. Esto incluye la descripción de las técnicas utilizadas, los pasos a seguir, las mejores prácticas y los recursos disponibles para su aplicación.

# 10. Control de medios.

Los medios de programas informáticos se pueden definir como aquellos medios en los que se almacenan datos informáticos. En nuestro caso si bien utilizaremos dispositivos convencionales como memoria RAM, USB o almacenamiento en la nube los métodos e instalaciones de control de medios deben garantizar que:

a) El software está almacenado y su recuperación está asegurada.

b) Se proporciona almacenamiento y recuperación fuera del sitio para software crítico y copias del código de referencia.

c) El software es accesible solo para aquellos con la necesidad de acceso.A

d) El entorno está controlado para que los medios físicos en los que se almacena el software no se degraden.

e) Se describa cómo se asegura el cumplimiento de lo anterior.

Considerando los aspectos previamente mencionados, se llevará a cabo la implementación del documento Plan gestión de medios, el cual ha sido desarrollado siguiendo el **estándar IEEE-730 Standard for Software Quality Assurance Plans**.

## 10.1 Acceso no autorizado

Hay varios métodos disponibles que proporcionarán una protección adecuada contra el acceso no autorizado a los medios de los programas informáticos. El método principal es proporcionar un esquema de identificación o etiquetado permanente dentro de los medios de almacenamiento, esta técnica puede proporcionar un control de contraseña adecuado o protección de acceso. Otros métodos incluyen una biblioteca de programas de acceso limitado, cifrado, marcas externas y declaraciones de propiedad que identifican un programa controlado. También se debe considerar la seguridad física de todos los medios.

Las actividades de SQA para verificar la idoneidad y la implementación de los procedimientos de acceso deben documentarse en el SQAP. Las áreas de preocupación incluyen la identificación de los programas que requieren acceso limitado, el cumplimiento de las restricciones de etiquetas y archivos, garantizar el uso de restricciones de etiquetado externas adecuadas y proporcionar un entorno controlado, como una biblioteca de programas.

## 10.2 Daños o degradación involuntarios

El daño o la degradación de los medios se pueden minimizar al proporcionar técnicas de gestión de configuración adecuadas, ubicaciones de almacenamiento seguras y respaldos constantes, métodos para documentos y medios físicos como bóvedas a prueba de incendios, y prácticas de empaque que tengan un diseño antiestático. De igual manera se realizarán revisiónes periódica para garantizar el uso de prácticas ambientales y de catalogación controladas minimizará la degradación de los medios.

# 11. Control de código.

**Esta sección no es relevante en el contexto de este plan**, dado que se trata de un proceso enfocado en el diseño de software y aún no se ha generado ningún código en esta etapa.

# 12. Recopilación, mantenimiento y retención de registros.

Los procedimientos relacionados con la recopilación, mantenimiento y retención de registros del Aseguramiento de la Calidad del Software (SQA, por sus siglas en inglés) establecidos en el Plan de Aseguramiento de la Calidad del Software (SQAP) deben estar en conformidad con los métodos definidos en el Plan de Gestión de la Configuración del Software (SCMP, por sus siglas en inglés) aprobado. Estos métodos se encuentran detallados en el SCMP, el cual se define en el capítulo 4 del documento y se basa en el estándar IEEE 1033-1985.

## 12.1 Recopilación de registros

El tipo de registros que se recopilarán ya sea en formato digital o impreso, está determinado por los objetivos generales para el mantenimiento de registros. Estos objetivos deben documentarse en el SQAP. Los posibles objetivos son:

a) Proporcionar evidencia contractual de que el proceso de desarrollo de software se realizó de conformidad con la práctica establecida y los requisitos del cliente:

1) Se sigue el SQAP y cumple con los requisitos de las normas aplicables.

2) El software cumple con la intención del diseño y satisface los requisitos contractuales.

3) La acción correctiva es efectiva, oportuna y completa (es decir, los elementos de acción se rastrean hasta la resolución).

4) Las pruebas se han realizado de acuerdo con los planes de prueba.

b) Proporcionar datos históricos o de referencia que puedan utilizarse para descubrir tendencias a largo plazo en las técnicas de desarrollo de la organización. Los documentos recopilados con fines históricos o de referencia deben ser capaces de proporcionar datos para estudios de productividad, calidad y metodología. Las métricas recopiladas deben revisarse para detectar tendencias y mejorar los procesos. Los documentos deben proporcionar suficientes datos de diseño, implementación y prueba para que sean útiles para el desarrollo futuro.

Además de los documentos SQA, los registros deben incluir medios de programas que contengan la versión exacta de los programas y los materiales utilizados en la realización de las pruebas para asegurar la repetibilidad de las pruebas en cualquier momento en el futuro.

El tipo de registros que se recopilarán ya sea en formato digital o impreso, está determinado por los objetivos generales para el mantenimiento de registros, los cuales deben ser documentados en el Plan de Aseguramiento de la Calidad del Software (SQAP). A continuación, se presentan los posibles objetivos:

a) Proporcionar evidencia contractual de que el proceso de desarrollo de software se llevó a cabo de acuerdo con las prácticas establecidas y los requisitos del cliente. Para cumplir con este objetivo, se deben considerar los siguientes puntos:

1. El SQAP es seguido y cumple con los requisitos de las normas aplicables.
2. El diseño cumple con su intención y satisface los requisitos contractuales.
3. Las acciones correctivas son efectivas, oportunas y completas, y se realiza un seguimiento de los elementos de acción hasta su resolución.

b) Proporcionar datos históricos o de referencia que puedan utilizarse para identificar tendencias a largo plazo en las técnicas de desarrollo de la organización. Los documentos recopilados con fines históricos o de referencia deben ser capaces de proporcionar datos para realizar estudios de productividad, calidad y metodología. Las métricas recopiladas deben ser revisadas para identificar tendencias y mejorar los procesos. Los documentos deben contener suficiente información sobre el diseño e implementación para que sean útiles en el desarrollo futuro.

Es esencial que estos objetivos y requisitos de registro sean debidamente documentados y seguidos para garantizar la trazabilidad, la transparencia y la evidencia adecuada del proceso de desarrollo de software, así como para facilitar la mejora continua y el aprendizaje organizacional.

## 12.2 Mantenimiento de registros

El SCMP, el cual se define en el capítulo 4 del documento y se basa en el estándar IEEE 1033-1985 especifica la forma en que se mantendrán los registros. Ademas indica la manera en que estos se almacenaran para protegerlos de incendios, robos o deterioro ambiental.

## 12.3 Conservación de registros

* **Documentos de diseño y especificaciones técnicas:** Estos documentos contienen información detallada sobre el diseño del software y las especificaciones técnicas relevantes. Un periodo de retención adecuado para estos documentos podría ser de al menos 5 años después de la finalización del proyecto, o incluso más tiempo si existen requisitos regulatorios o contratos que lo exijan.
* **Registros de acciones correctivas y preventivas:** Los registros que documentan las acciones correctivas y preventivas tomadas durante el desarrollo del software son esenciales para garantizar la mejora continua del proceso. Estos registros deben conservarse durante un periodo de retención de al menos 3 años después de su implementación y cierre.
* **Documentación de cambios y versiones:** Los registros que documentan los cambios realizados en el software y las diferentes versiones lanzadas deben conservarse durante un periodo de retención adecuado. Esto puede variar según las necesidades de la organización y los requisitos contractuales, pero un periodo de retención comúnmente utilizado es de al menos 5 años después de la fecha de lanzamiento de la última versión del software.
* **Documentos de requisitos del software:** Estos documentos describen los requisitos funcionales y no funcionales del software. Un periodo de retención recomendado para los documentos de requisitos es de al menos 5 años después de la finalización del proyecto, o incluso más tiempo si existen requisitos regulatorios o contratos que lo exijan.
* **Documentos de capacitación y manuales de usuario**: Estos documentos proporcionan información sobre la capacitación del personal y el uso del software por parte de los usuarios. Un periodo de retención apropiado para estos documentos puede ser de al menos 5 años después de la fecha de lanzamiento del software.
* **Documentos de análisis de riesgos**: Los documentos de análisis de riesgos identifican y evalúan los riesgos asociados al proyecto y al software desarrollado. Se recomienda conservar estos documentos durante un periodo de retención de al menos 3 años después de la finalización del proyecto para tener un registro histórico de los riesgos identificados y las medidas tomadas para mitigarlos.

# 13. Entrenamiento

El plan de entrenamiento se llevará a cabo siguiendo la siguiente tabla, la cual establece los requisitos principales para los distintos perfiles que utilizarán este plan de SQA.

|  |  |  |  |
| --- | --- | --- | --- |
| Perfil | Requisito de habilidades/conocimientos | Habilidades del personal | Brechas de capacitación |
| Especialista en Estándares de Calidad de Software | Conocimiento de los estándares establecidos | Pensamiento analítico | Certificación los estándares establecidos |
| Analista de Procesos de Software | Comunicación efectiva | Comunicación efectiva | Entrenamiento en técnicas de comunicación y presentación |
| Desarrollador de Proyectos de Software | Trabajo en equipo | Trabajo en equipo | Capacitación en gestión de proyectos y colaboración en equipo |
| Analista de Calidad de Software | Resolución de problemas | Resolución de problemas | Entrenamiento en técnicas de análisis y documentación |
| Modelador de Procesos de Software | Conocimiento de herramientas de modelado de procesos de software (por ejemplo, UML) | Orientación a la calidad | Capacitación en herramientas de modelado de procesos |
| Gerente de Calidad de Software | Capacidad para gestionar cambios y adaptarse a nuevas tecnologías | Pensamiento crítico | Capacitación en gestión del cambio y adopción de nuevas tecnologías |

#### **Plan de capacitación.**

La capacitación se llevará a cabo de acuerdo con nuestro plan de capacitación, el cual ha sido diseñado en base al estándar ISO 9001 y se ha fundamentado en los modelos ADDIE y SCORM. Estos modelos han sido utilizados como base para la creación y desarrollo del plan de capacitación.

# 14. Gestión de riesgos

La gestión de riesgos se llevará a cabo en conformidad con nuestro plan de gestión de riesgos, el cual se basa en el estándar ISO/IEC 31010:2009 "Gestión del riesgo - Técnicas para el proceso de evaluación de riesgos".

|  |  |
| --- | --- |
| Riesgo | Solución |
| Falta de recursos adecuados para el SQA. | Identificar los recursos necesarios (personal, herramientas, tiempo) y garantizar su disponibilidad. Asignar roles y responsabilidades claras para el equipo de SQA. |
| Cambios frecuentes en los requisitos del proyecto. | Establecer un proceso formal de gestión de cambios para evaluar, aprobar y controlar los cambios en los requisitos. Realizar un seguimiento y comunicar cualquier cambio a todas las partes interesadas relevantes. |
| Inadecuada definición y seguimiento de métricas de calidad. | Identificar las métricas de calidad relevantes para el proyecto y establecer métodos para medirlas y recopilar datos. Establecer objetivos claros y realizar un seguimiento regular para evaluar el cumplimiento de las métricas establecidas. |
| Comunicación deficiente entre los equipos de desarrollo y el equipo de SQA. | Establecer canales de comunicación claros y frecuentes entre los equipos. Realizar reuniones regulares para discutir el progreso del SQA, compartir información y abordar cualquier problema o preocupación. |
| Deficiencias en el proceso de pruebas. | Establecer un proceso de pruebas estructurado y bien definido. Identificar los tipos de pruebas adecuados para el proyecto, incluyendo pruebas de unidad, integración, funcionales, de rendimiento, etc. Realizar pruebas exhaustivas y documentar los resultados. |
| Falta de capacitación y habilidades adecuadas del equipo de SQA. | Evaluar las habilidades del equipo de SQA y proporcionar la capacitación necesaria para llenar cualquier brecha identificada. Fomentar el desarrollo profesional y el intercambio de conocimientos entre los miembros del equipo. |
| Falta de seguimiento y control del proceso de SQA. | Establecer un sistema de seguimiento y control para monitorear el progreso del SQA. Realizar revisiones y auditorías periódicas para identificar problemas y tomar acciones correctivas cuando sea necesario. |
| Falta de documentación y trazabilidad adecuadas. | Establecer un proceso de documentación claro y asegurar que la trazabilidad entre los requisitos, las pruebas y los resultados esté bien documentada. Utilizar herramientas de gestión de requisitos y de gestión de pruebas para facilitar este proceso. |
| Falta de compromiso y apoyo de la alta dirección. | Obtener el compromiso y el apoyo de la alta dirección mediante la comunicación efectiva de la importancia del SQA y sus beneficios. Proporcionar informes regulares sobre el estado del SQA y los logros alcanzados. |

# 15. Glosario

1. SQAP (Software Quality Assurance Plan): Plan de Aseguramiento de la Calidad del Software. Documento que establece las actividades y los procesos para garantizar la calidad del software desarrollado.
2. Ciclo de vida del software: Conjunto de fases o etapas por las que atraviesa el desarrollo de un software, desde su concepción hasta su retirada. Incluye etapas como análisis de requisitos, diseño, implementación, pruebas y mantenimiento.
3. Control de calidad: Conjunto de actividades y procesos que se llevan a cabo para asegurar que el software cumple con los estándares de calidad establecidos.
4. Métricas de calidad: Indicadores cuantitativos utilizados para medir y evaluar la calidad del software. Ejemplos de métricas de calidad incluyen la tasa de defectos, la cobertura de pruebas y el tiempo promedio de respuesta.
5. Revisión de código: Proceso de examinar y evaluar el código fuente del software para identificar posibles problemas, errores o violaciones de estándares de codificación.
6. Prueba de aceptación: Pruebas realizadas para verificar si el software cumple con los requisitos del cliente y si está listo para ser entregado y utilizado en el entorno de producción.
7. No conformidad: Situación en la que el software o los procesos no cumplen con los estándares o requisitos establecidos en el SQAP. Requiere acciones correctivas para solucionar la no conformidad.
8. Auditoría de calidad: Proceso de evaluación independiente para determinar si los procesos y productos del software cumplen con los estándares de calidad establecidos.
9. Acción preventiva: Medida tomada para evitar la aparición de problemas o desviaciones en el desarrollo del software, con el objetivo de mejorar la calidad y evitar riesgos potenciales.
10. Plan de pruebas: Documento que describe las estrategias, técnicas y recursos utilizados para realizar pruebas en el software y garantizar su funcionamiento correcto y la detección de posibles fallos.

## 15.1 Acronimos

Los acrónimos encontrados en este documento son los siguientes:

ADR architecture design review

DDR detailed design review

SCM software configuration management

SCMP software configuration management plan

SCMPR software configuration management plan review

SDD software design description

SQA software quality assurance

SQAP software quality assurance plan

SRD software requirements description

SSR software specifications review

UDR user documentation review

# 16. Procedimiento e historial de cambio de SQAP

Procedimiento para Modificar el SQAP:

1. Cualquier cambio propuesto al SQAP deberá ser presentado por escrito al Comité de Control de Cambios (CCB, por sus siglas en inglés) designado.
2. El CCB revisará los cambios propuestos y evaluará su impacto en el SQAP y los objetivos del proyecto.
3. Basándose en la evaluación, el CCB aprobará o rechazará los cambios propuestos.
4. Si se aprueba, el CCB documentará los cambios aprobados y actualizará el SQAP en consecuencia.
5. El SQAP actualizado deberá ser comunicado a todos los miembros del equipo y partes interesadas relevantes.

Historial de Modificaciones:

El SQAP mantendrá un historial de todas las modificaciones realizadas, que incluirá lo siguiente:

* Fecha de la modificación.
* Descripción detallada de los cambios realizados.
* Justificación o razón detrás de la modificación.
* Autor o responsable de la modificación.
* Estado actualizado del SQAP después de la modificación.