****

**PLAN DE ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE**

FemmeCode

**Aseguramiento de la calidad de software**

**Entrega final**

Facultad de matemáticas UADY

**Autores:**

Manuel Antonio Cituk Martínez

Jorge Teodoro Dawn Rodríguez

Geovanna Fernanda De La Cruz Medina

Alma Angélica Ordoñez Sánchez

Verónica Marilyn Rivera Manzanero

**Fecha de elaboración**: mayo 2023

**Control de configuración**

|  |  |
| --- | --- |
| **Plan de aseguramiento de la calidad de software** | |
| Autores: | Manuel Antonio Cituk Martínez |
| Jorge Teodoro Dawn Rodríguez |
| Geovanna Fernanda De La Cruz Medina |
| Alma Angélica Ordoñez Sánchez |
| Verónica Marilyn Rivera Manzanero |
| Fecha: | 26 de abril 2023 |

**Histórico de versiones**

Estado: (B)orrador, (R)evisión, (A)probado

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Fecha** | **Estado** | **Responsable** | **Nombre de archivo** |
| 1.0 | 13/05/2023 | A | Equipo | PMPF\_v1.0.docx |
| 2.0 | 25/05/2023 | R | Equipo | PMPF\_v2.0.docx |

**Histórico de cambios**

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Cambios** |
| 1.0 | 16/05/2023 | Se cambió el estándar a usar para realizar el plan de aseguramiento de calidad.  Se realizó un cambio a la organización de la unidad de SQA.  Se añadieron y modificaron documentos referenciados  Se modificó el plan para que únicamente hiciera referencia a la fase de construcción. |

**Índice**

[**1. Propósito**](#_4yv3gs1hiqqm) 5

[**2. Documentos referenciados**](#_dmu0a3xttkt9) 6

[**3. Gestión**](#_x5wkwu3ecjcy) 8

[**4. Documentación**](#_hqucvrcuro0a) 16

[4.1 Plan de verificación y validación del software](#_1siipjs8swjo) 16

[4.2 Reporte de verificación y validación del software](#_mlhpgwy41nxx) 16

[4.3 Registro de rastreo de código](#_qu82tpbpzi39) 17

[4.4 Plan de pruebas Unitarias](#_3u6y8pc4z6ux) 18

[4.5 Reporte de Actividades](#_43ma2kwpoge5) 18

[4.6 Plan de Gestión de la Configuración de software](#_zh1tigo94bfq) 19

[**5. Estándares, prácticas, convenciones y métricas**](#_ttk5mqai2t55) 20

[5.1 Propósito.](#_qlslmrqazjtk) 20

[5.2 Contenido](#_hhveu4knuq1a) 20

[**6. Revisiones y auditorías**](#_rkbu13qswpwc) 21

[6.1 Auditoría del código fuente](#_yx7sp5xyx7wg) 21

[6.2 Revisión del Registro de rastreo de código](#_d79kvldvwskt) 22

[6.3 Revisión del plan de pruebas unitarias](#_lhjz0gj5euxi) 22

[6.4 Revisión del plan de verificación y validación del software](#_fbgouw3pfwjm) 23

[6.5 Revisión del reporte de actividades](#_nonpjv4sqdsw) 24

[6.6 Revisión del Plan de Gestión de la Configuración de software](#_ro89v2uu8uoo) 24

[**7. Pruebas**](#_x82mzwi9p9w2) 25

[7.1 Pruebas Unitarias](#_7otswq76bmfm) 25

[**8. Reporte de problemas y acciones correctivas**](#_m3tuht739x29) 26

[**9. Herramientas, tecnicas y metodologias**](#_vdby96kkrnqs) 27

[9.1 Herramientas](#_7wmq1fklznyn) 27

[9.2 Técnicas](#_ouvjy7guffdu) 28

[9.3 Metodologías](#_vb7ervw2mxs3) 28

[**10. Control de código**](#_4myjhv3m1s0t) 28

[10.1 Identificación del código](#_rr7wmfen9hb3) 29

[10.2 Localización física del código](#_9zmotx44ruv2) 29

[10.3 Solicitud de cambios](#_i815bimh0glt) 29

[10.4 Aprobación o desaprobación de la solicitud](#_3dsgva8hncd0) 30

[10.5 Copias de seguridad](#_3n8n6zdb9kh) 30

[10.6 Distribución de copias](#_f93t963js7oz) 30

[10.7 Proceso de liberación de la nueva versión](#_cis74q985zgs) 30

[10.8 Aseguramiento de los cambios](#_i6fiekl3hni5) 31

[**11. Control de medios**](#_dolykdf3yheu) 32

[11.1 Acceso no autorizado](#_qr4cbfomfssy) 32

[11.2 Daño o degradación inadvertida](#_3l94ad85v1o0) 33

[**12. Recopilación, mantenimiento y retención de registros**](#_vppeq1s2uyrg) 34

[12.1 Recopilación de registros](#_qiw44su94bij) 34

[12.2 Mantenimiento de registros](#_uobjxx6gd4hc) 35

[12.3 Retención de registros](#_61xg40vmc7em) 35

[**13. Capacitación**](#_1gjgw0vwhtau) 35

[**14. Gestión de riesgos**](#_5gdxv0spys3i) 38

[**15. Anexos**](#_hix2iakbfamo) 40

# Propósito

El siguiente plan de aseguramiento de la calidad del software se encuentra basado en el estándar IEEE Std 730:1995. Tiene como propósito brindarle al equipo de aseguramiento de la calidad de la empresa FemmeCode actividades específicas a realizar que les permitan garantizar la calidad del proceso de desarrollo de la aplicación.

El plan será aplicado en el proyecto de desarrollo de la aplicación de la empresa, la cual tiene como objetivo brindar seguridad a las mujeres, teniendo la funcionalidad de contactar a las autoridades en caso de haber un posible escenario de violencia. De igual forma, la aplicación contará con un apartado de asesoría jurídica para orientar a las mujeres que deseen levantar denuncias.

El presente plan contiene los elementos involucrados en el soporte de la calidad del proceso de la fase de construcción, así como los productos de dicha fase. Mismos que serán sometidos a revisión para asegurar su calidad. Como modelo del ciclo de vida, se usará el de cascada.

Los elementos cubiertos en este plan son:

* Código fuente
* Registro de rastreo
* Plan de pruebas unitarias
* Plan de verificación y validación del software
* Plan de gestión de la configuración de software
* Reporte de actividades
* Reporte de verificación del plan de verificación y validación

Este documento está dirigido al equipo de aseguramiento de calidad de software, al responsable de la elaboración y actualización del presente plan, al líder del proyecto y equipo de desarrollo y mantenimiento, esto con el objetivo de garantizar que los productos de software y la documentación correspondiente cumpla con los requisitos que fueron especificados.

# Documentos referenciados

ISO/IEC 12207: Systems and Software Engineering - Software Life Cycle Processes

Este estándar establece los procesos del ciclo de vida del software, incluyendo la adquisición, el suministro, el desarrollo, la operación, el mantenimiento y la retirada del software. Proporciona una estructura general para la gestión y la ejecución de proyectos de software.

IEEE Std 730.1-1995: Guide for Software Quality Assurance Planning

Establece pautas para la planificación efectiva del aseguramiento de la calidad del software en proyectos. Proporciona directrices sobre la identificación de actividades y recursos necesarios, la definición de estándares y procedimientos de calidad, la gestión de riesgos y la evaluación de los resultados.

"Software Quality Assurance: From Theory to Implementation"

Proporciona una amplia visión general de los principios y prácticas de aseguramiento de calidad del software, desde los fundamentos teóricos hasta su implementación en proyectos reales. Se centra en aspectos clave como la planificación y gestión de la calidad, el control de calidad del proceso de desarrollo, las técnicas de prueba y verificación, y la gestión de la configuración.

IEEE Std 829-2008: IEEE Standard for Software and System Test Documentation:

Este es un estándar para la documentación de pruebas de software. Proporciona un marco para documentar diferentes tipos de pruebas de software, como pruebas de unidad, pruebas de integración y pruebas de sistema.

“Code Complete: A Practical Handbook of Software Construction” de Steve McConnell

Este libro aborda todos los aspectos del desarrollo de software, incluyendo la codificación, y ofrece prácticas recomendadas para crear un código de alta calidad.

“Java Code Conventions”

Estándares y recomendaciones establecidas por Oracle para escribir código Java de manera consistente y legible. Estas convenciones ayudan a mejorar la claridad y la comprensión del código, facilitando el mantenimiento y la colaboración en proyectos de desarrollo de software en Java.

"Clean Code: A Handbook of Agile Software Craftsmanship"

Se enfoca en las mejores prácticas de codificación para producir un código limpio y de calidad. El libro proporciona principios y técnicas específicas para mejorar la legibilidad, la mantenibilidad y la eficiencia del código.

IEEE Std 1008-1987: IEEE Standard for Software Unit Testing

Define un enfoque sistemático y documentado para las pruebas unitarias.

MPS 3.0 - Modelo de procesos de software

Este documento representa el modelo de procesos adoptado por nuestra organización, el cual se ha desarrollado utilizando el marco de referencia MoProSoft. Contiene una serie de componentes que sirven como guía para establecer una secuencia ordenada de actividades y fases necesarias para garantizar la calidad tanto de nuestros procesos como de los productos resultantes.

Administración de la Configuración de Software

En este documento se detalla la gestión de cada uno de los componentes relacionados con la configuración de software dentro del modelo de procesos de software adoptado por la organización.

ISO/IEC 27001 - Tecnología de la información - Técnicas de seguridad - Sistemas de gestión de la seguridad de la información - Requisitos: Este estándar establece los requisitos para el establecimiento, implementación, mantenimiento y mejora de un sistema de gestión de seguridad de la información. Incluye directrices y controles relacionados con el control de medios y la gestión de activos de información.

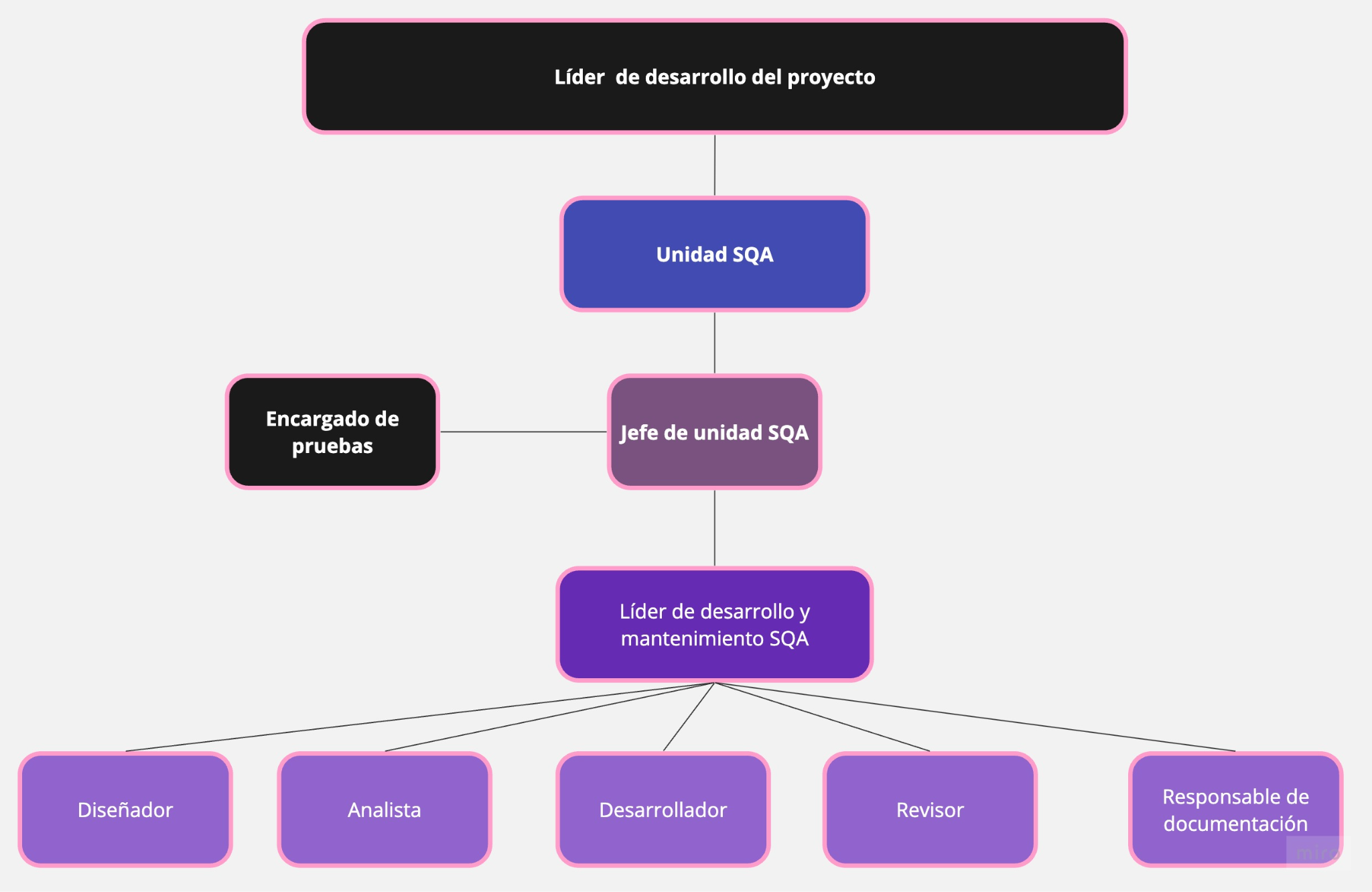
ISO 31000:2018 - Gestión del riesgo: Directrices: Esta es una norma internacional que establece los principios y directrices generales para la gestión de riesgos. Proporciona un marco de trabajo integral para identificar, analizar, evaluar y tratar los riesgos de manera sistemática y efectiva.

# Gestión

**3.1 Organización**

La unidad de aseguramiento de la calidad **(SQA-U)** que llevará a cabo la implementación del plan de SQA deberá estar conformada por 7 integrantes:

1. Jefe de unidad (SQA-JU)
2. Líder de equipo de desarrollo y mantenimiento (SQA-DM)
3. Diseñador (DI)
4. Analista (AN)
5. Desarrollador (DEV)
6. Revisor (REV)
7. Responsable de documentación. (R-DOC)



*Imagen 1. Organigrama representativo de la unidad de SQA*

El líder de desarrollo del proyecto (LP) y el encargado de pruebas (R-TEST) no pertenecen a la unidad de SQA, pero su relación con ella es crucial para la correcta aplicación del plan de aseguramiento ya que deben estar en constante comunicación para desarrollar sus actividades correspondientes (mismas que se especificarán más adelante).

Dentro de las responsabilidades del jefe de unidad (SQA-JU) se encuentra el mantenerse en contacto con el líder del proyecto (LP) para notificarle sobre el seguimiento de las actividades que se encuentra realizando la unidad. De igual forma, el líder de proyecto mantendrá actualizada a la unidad sobre los cambios o modificaciones que se tengan que realizar a los componentes de software.

El encargado de pruebas (R-TEST) deberá mantenerse en contacto constante con el jefe de unidad, ya que mantendrá actualizadas las plantillas y checklist a usar en caso de ser necesario o de que SQA-JU se lo indique.

El líder de desarrollo y mantenimiento (SQA-DM) deberá coordinar las actividades a elaborar por parte de los integrantes de la unidad, con el fin de que el plan se siga en tiempo y forma. También deberá notificar el reporte de las actividades realizadas al SQA-JU, con el fin de que éste le brinde retroalimentación y posibles mejoras.

El resto de los integrantes pertenecientes a la unidad de SQA (Diseñador, Analista, Desarrollador y Revisor) se encargará de revisar y validar los componentes de software. El responsable de documentación (R-DOC) se encargará de documentar todo el proceso para posteriormente hacer llegar el reporte de las actividades realizadas al SQA-DM.

**3.2 Tareas**

Tareas generales

|  |  |  |
| --- | --- | --- |
|  | **Rol** | **Actividad** |
| **1** | SQA-JU  SQA-DM  DI  AN  DEV  REV  R-DOC | Revisar el Plan de Desarrollo actual a través de reuniones acordadas, con el fin de lograr un entendimiento común y obtener su compromiso con el aseguramiento de la calidad. |
| **2** | REV  SQA-JU  SQA-DM | Evaluar las actividades realizadas de SQA para verificar que se estén cumpliendo de manera adecuada. |
| **3** | SQA-U  R-DOC  SQA-JU  SQA-DM | Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad, participantes y anotaciones. |
| **4** | LP  SQA-U | Comprobar que lo documentado en el reporte de actividades coincida con los objetivos iniciales |

*Tabla 1.0 Tareas generales*

Tareas de inicio

No aplica. Este plan está enfocado en la fase de construcción.

Tareas del análisis y diseño

No aplica. Este plan está enfocado en la fase de construcción.

Tareas de construcción

**Entradas:**

* Registro de rastreo de código
* Plan de Gestión de la Configuración de software
* Plan de verificación y validación del software

**Salidas:**

* Registro de rastreo de código
* Reporte de verificación y validación del software
* Reporte de actividades
* Plan de pruebas unitarias

**Actividades**

|  |  |  |
| --- | --- | --- |
|  | **Rol** | **Actividad** |
| **1** | SQA-DM | Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo con el Plan de Desarrollo actual. |
| **2** | SQA-JU  SQA-DM  AN | Recolectar los componentes de software con base a la parte detallada de la fase de construcción. |
| **3** | SQA-JU  SQA-DM | Realizar una sesión de revisión para verificar los componentes previamente recolectados. |
| **4** | R-DOC | Documentar la revisión de los componentes previamente revisados. |
| **5** | AN  DEV  REV  R-DOC | Realizar las modificaciones necesarias a los componentes revisados. |
| **6** | DEV  DI | Definir y aplicar pruebas unitarias para verificar que el funcionamiento de cada componente esté acorde con la parte detallada del proceso de construcción. |
| **7** | DEV | Corregir los defectos encontrados hasta lograr pruebas unitarias exitosas (sin defectos). |
| **8** | SQA-JU  SQA-DM  DI  AN  DEV  REV | Realizar la sesión de revisión para la verificación del Plan de pruebas unitarias. |
| **9** | REV  R-DOC | Actualizar el Registro de Rastreo, incorporando los componentes construidos o modificados. |
| **10** | SQA-DM  AN  DEV  REV | Corregir los defectos encontrados en el Registro de Rastreo con base en el Reporte de Verificación y obtener la aprobación de las correcciones. |
| **11** | REV  R-DOC | Incorporar Componentes y Registro de Rastreo como líneas base a la Configuración de Software. |
| **12** | SQA-JU  SQA-DM  REV  R-DOC | Elaborar el Reporte de Actividades. |

*Tabla 2.0 Tareas de la fase de construcción*

Tareas de implementación y pruebas

No aplica. Este plan está enfocado en la fase de construcción.

Tareas de cierre

No aplica. Este plan está enfocado en la fase de construcción.

**3.3 Responsabilidades**

|  |  |  |
| --- | --- | --- |
|  | **Rol** | **Responsabilidades** |
| **LP** | Líder del proyecto | * Mantener informada a la unidad de SQA sobre los cambios o modificaciones que se tengan que realizar a los componentes de software. |
| **SQA-U** | Unidad de SQA | * Garantizar la calidad de los ítems de la fase de construcción a través de actividades asignadas a cada miembro de la unidad. |
| **SQA-JU** | Jefe de unidad de SQA | * Informar al líder del proyecto sobre todos los cambios y avances relevantes que se encuentre realizando la unidad. * Mantener el seguimiento de las actividades a realizar por la unidad de SQA. * Mantener la comunicación con el responsable de pruebas para realizar modificaciones correspondientes de ser necesario. |
| **SQA-DM** | Líder de desarrollo y mantenimiento SQA | * Seguir en tiempo y forma las actividades establecidas para el aseguramiento de la calidad. * Mantenerse en constante comunicación con el jefe de la unidad para notificarle de todo lo que acontece en el proceso de aseguramiento de la calidad por parte de la unidad. * Guiar a su equipo y mantenerse al tanto de los avances de cada integrante. |
| **DI** | Diseñador | * Diseñar las pruebas unitarias con base en el plan de la empresa, para que el líder de desarrollo y mantenimiento posteriormente pueda pasarle la documentación al encargado de pruebas. * Revisar la construcción de los componentes de software para informar de cualquier cambio o solicitud de cambio al líder. |
| **AN** | Analista | * Revisar los componentes de software, estar al tanto de la funcionalidad de cada uno de ellos y comunicarle al líder de desarrollo y mantenimiento cualquier detección de defectos. |
| **DEV** | Desarrollador | * Revisar que el código tenga una correcta construcción y cumpla sus funcionalidades correspondientes. * Revisar los demás elementos a revisar para brindar su opinión técnica. * Mantenerse en comunicación con el líder de desarrollo y mantenimiento ante alguna detección de defectos. |
| **REV** | Revisor | * Participar en todas las revisiones correspondientes al aseguramiento de la calidad de cada elemento a revisar, de acuerdo con el plan. * Brindar retroalimentación en todas las sesiones de revisión. * Detectar desviaciones de estándares para informarle al líder. |
| **R-DOC** | Responsable de documentación | * Documentar de acuerdo con el plan de configuración de la empresa todos aquellos elementos sometidos a revisión que incluyan documentación. |
| **R-TEST** | Responsable de pruebas | * Mantenerse en contacto constante con el jefe de unidad, ya que mantendrá actualizadas las plantillas y checklist a usar en caso de ser necesario o de que SQA-JU se lo indique. |

*Tabla 3.0 Responsabilidades de la organización*

# Documentación

A continuación, se presentan aquellos documentos pertenecientes a la fase de construcción que deben ser sometidos a revisión:

## 4.1 Plan de verificación y validación del software

Este plan establece las actividades de verificación y validación de los requerimientos y el código de software desarrollados por la compañía "FemmeCode". Su propósito es ofrecer orientación sobre los procesos y herramientas de verificación y validación que se aplicarán en estos productos. El contenido de este plan está basado en el estándar de verificación y validación para sistemas, software y hardware de la IEEE 1012-2016.

*Para más información sobre el plan de la empresa, ir al anexo 9. “Plan de verificación y validación”.*

Para asegurarse de que este documento cumple con la calidad deseada, se realizará una revisión por pares, para así, obtener múltiples perspectivas y opiniones sobre la calidad y efectividad del plan de verificación y validación. El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”.*

## 4.2 Reporte de verificación y validación del software

El reporte de verificación y validación del software tiene como objetivo principal proporcionar un resumen detallado de los resultados obtenidos después de la implementación de las actividades del plan de verificación y validación del software. Este reporte se hace con base en la plantilla del reporte de verificación y validación del software que está realizada con la norma ISO/IEC 12207.

*Para más información sobre la plantilla, ir al anexo 10.” Plantilla verificación y validación”.*

Para asegurarse de que este documento cumple con la calidad deseada, se realizará una revisión por pares, para así, obtener múltiples perspectivas y opiniones sobre cómo se ha realizado el reporte de verificación y validación del software conforme a la plantilla antes mencionada. El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”.*

## 4.3 Registro de rastreo de código

El registro de rastreo en la fase de construcción de software sirve para mantener la coherencia entre los requisitos, el diseño, la implementación y las pruebas, asegurando que el software desarrollado cumpla con los requisitos establecidos y facilitando la identificación y solución de problemas durante el proceso de construcción. El registro de rastro deberá realizarse de acuerdo con la plantilla desarrollada con base en el estándar “IEEE Std 1012-2016: IEEE Standard for System, Software, and Hardware Verification and Validation”.

*Para más información sobre esta plantilla, consultar el anexo 1. “Plantilla Registro de rastreo”.*

Para asegurarse de que este documento cumple con la calidad deseada, se realizará una revisión por pares, para así, obtener múltiples perspectivas y opiniones sobre la calidad y efectividad del registro. El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”.*

## 4.4 Plan de pruebas Unitarias

El plan de pruebas unitarias en la fase de codificación es una herramienta esencial para garantizar la calidad del código y mitigar posibles problemas antes de que el software sea implementado. Ayuda a identificar errores, mejorar la calidad del código y aumentar la confiabilidad del software, lo que finalmente contribuye a entregar un producto de software más robusto y satisfactorio para los usuarios.

Para llevar a cabo el plan de pruebas unitarias, se debe consultar el plan de pruebas creado por la empresa, basado en el estándar “IEEE Std 1008-1987 el cual contiene 3 plantillas en donde se deberá registrar la descripción, el plan, y el reporte de las pruebas.

*Para más información sobre las plantillas previamente descritas, ir al anexo 3. “Plantillas para plan de pruebas”.*

*Para más información sobre el plan de pruebas de la empresa ir al anexo 4. “Plan de pruebas”.*

Para asegurarse de que este documento cumple con la calidad deseada, se realizará una revisión por pares, para así, obtener múltiples perspectivas y opiniones sobre la calidad y efectividad del plan de pruebas unitarias. El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”.*

## 4.5 Reporte de Actividades

Al final de la fase de construcción, se deben registrar y describir en un documento, las actividades realizadas durante esta misma. Este documento debe proporcionar una visión general de las tareas llevadas a cabo, el progreso alcanzado, los problemas encontrados y las soluciones implementadas a lo largo de esta parte del proyecto.

Se deberán registrar las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas. Lo anterior de acuerdo con la Norma ISO 9001: La norma exige que se documenten los resultados de las actividades realizadas, así como los procesos utilizados para llevarlas a cabo.

La plantilla para implementar para dicho reporte será: “Informe del estado del proyecto”, creada por Microsoft.

*Para más información sobre la plantilla, ir al anexo 5.0 “Plantilla de Informe del estado del proyecto”.*

Para asegurarse de que este documento cumple con la calidad deseada, se realizará una revisión por pares, para así, obtener múltiples perspectivas y opiniones sobre la calidad y efectividad del reporte de actividades. El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”*

## 4.6 Plan de Gestión de la Configuración de software

El plan de gestión de configuración de software en la fase de implementación tiene como objetivo principal establecer y mantener la integridad y la consistencia del software durante todo el proceso de implementación. Este plan proporciona una estructura y un conjunto de procedimientos para gestionar los cambios y las versiones del software, asegurando que se mantenga una trazabilidad adecuada y que se realicen las actualizaciones y despliegues de manera controlada y eficiente.

El plan de gestión de configuración de software debe describir: las actividades de la gestión de configuración, los responsables de la organización en realizar las actividades y la gestión de configuración, y su relación con otras organizaciones, como desarrollo de software o mantenimiento. Este plan se deberá basar en el estándar “ISO 10007-2017: Quality Management: Guidelines for configuration management”.

Para asegurarse de que este documento cumple con la calidad deseada, se realizará una revisión por pares, para así, obtener múltiples perspectivas y opiniones sobre la calidad y efectividad del plan de Gestión de la Configuración de software.

El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”*

# Estándares, prácticas, convenciones y métricas

## 5.1 Propósito.

Apartado dedicado a la identificación de los estándares, prácticas, convenciones y métricas que serán utilizados durante el proyecto.

## 5.2 Contenido

5.2.1 Fase de requerimientos

No aplica, ya que este plan, se cubre únicamente los aspectos relacionados a la fase de Construcción

5.2.2 Fase de análisis y diseño

No aplica, ya que este plan, se cubre únicamente los aspectos relacionados a la fase de Construcción

5.2.3 Fase de Construcción

* “Code Complete: A Practical Handbook of Software Construction” de Steve McConnell”
* “Java Code Conventions”
* "Clean Code: A Handbook of Agile Software Craftsmanship"
* IEEE Std 1008-1987: IEEE Standard for Software Unit Testing
* IEEE Std 1012-2016: IEEE Standard for System, Software, and Hardware Verification and Validation
* MIL-STD-498 Software Development and Documentation

5.2.4 Fase de Integración y Pruebas

No aplica, ya que este plan, se cubre únicamente los aspectos relacionados a la fase de Construcción

5.2.5. Cierre

No aplica, ya que este plan, se cubre únicamente los aspectos relacionados a la fase de Construcción

5.2.6. Documentación

* IEEE Std 829-2008: IEEE Standard for Software and System Test Documentation
* IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits
* ISO 10007-2017: Quality Management: Guidelines for configuration management

5.2.7. Métricas

* IEEE Std 1061-1998: IEEE Standard for a Software Quality Metrics Methodology

# Revisiones y auditorías

A continuación, se describirán las revisiones y/o auditorías que se deberán llevar a cabo para determinar el alcance del progreso del proyecto y asegurar el seguimiento de los elementos establecidos en el plan de aseguramiento de la calidad.

## 6.1 Auditoría del código fuente

Se aplicará una auditoría funcional al producto de código para verificar si cumple con los requisitos funcionales, es decir, se evalúa el comportamiento del código en diferentes escenarios para asegurar la calidad deseada.

El proceso para realizar dicha auditoría se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”.*

## 6.2 Revisión del Registro de rastreo de código

Se aplicará una revisión por pares al registro de rastreo, para así, obtener múltiples perspectivas y opiniones sobre la calidad y efectividad del registro. De igual forma, para verificar que se esté llevando a cabo de manera correcta, ya que es de vital importancia.

El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”.*

## 6.3 Revisión del plan de pruebas unitarias

Se aplicará una revisión por pares al plan de pruebas unitarias para asegurarse de que este documento cumple con la calidad deseada, de igual forma, para obtener múltiples perspectivas y opiniones sobre la calidad y efectividad del plan de pruebas unitarias.

El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”*.

## 6.4 Revisión del plan de verificación y validación del software

Se aplicará una revisión por pares al plan de verificación y validación del software para lograr tener una retroalimentación del plan, áreas de mejora y la calidad de este. De igual forma, la revisión aportará diversas perspectivas y opiniones a considerar para la implementación de mejoras.

El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”*.

**6.5 Revisión del reporte del plan de verificación y validación**

Se aplicará una revisión por pares al reporte del plan de verificación y validación para asegurarse de que este documento cumple con la calidad deseada, para así, obtener múltiples perspectivas y opiniones sobre la calidad y efectividad del reporte de actividades.

De igual forma la revisión detectará el correcto seguimiento de la documentación de acuerdo con la administración de la configuración y la gestión de control de la empresa. Cabe destacar que a través de esta revisión se detectarán posibles desviaciones de la plantilla a usar para el reporte.

El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”*

## 6.5 Revisión del reporte de actividades

Se aplicará una revisión por pares al reporte de actividades para asegurarse de que este documento cumple con la calidad deseada, para así, obtener múltiples perspectivas y opiniones sobre la calidad y efectividad del reporte de actividades. De igual forma la revisión detectará el correcto seguimiento de la documentación de acuerdo con la administración de la configuración y la gestión de control de la empresa. Cabe destacar que a través de esta revisión se detectarán posibles desviaciones de la plantilla a usar para el reporte.

El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”*

## 6.6 Revisión del Plan de Gestión de la Configuración de software

Se aplicará una revisión por pares al plan de gestión de la configuración de software para asegurarse de que este documento cumple con la calidad deseada, para así, obtener múltiples perspectivas y opiniones sobre la calidad y efectividad del plan dicho plan.

El proceso para realizar dicha revisión se encuentra definido en el estándar “Plan de revisión” de la empresa FemmeCode, el cual está elaborado con base en el estándar “IEEE Std 1028-2008: IEEE Standard for Software Reviews and Audits” y el modelo “Software Engineering Institute Capability Maturity Model Integration” (CMMI).

*Para más información sobre el plan de revisión de la empresa ir al anexo 2. “Plan de revisión”*

# Pruebas

El propósito de las pruebas en la fase de construcción de un proyecto es garantizar la calidad y la fiabilidad del software que se está desarrollando.

## 7.1 Pruebas Unitarias

Para llevar a cabo las pruebas unitarias, se debe consultar el plan de pruebas ya creado por la empresa, el cual siguió el estándar “IEEE Std 1008-1987: IEEE Standard for Software Unit Testing” y se conforma de:

* Descripción del entorno de prueba de software
* Descripción de las pruebas que se realizarán, incluyendo objetivos de la prueba, el nivel de prueba (componente o sistema), las clases de prueba, las condiciones y la secuencia, el registro/reducción/análisis de datos
* Cronogramas para las actividades de prueba
* Trazabilidad de los requisitos entre los requisitos del sistema de software/componente y las pruebas que los abordan.

Para la documentación de estas pruebas, se siguen los estándares “IEEE Std 829-2008: IEEE Standard for Software and System Test Documentation” y "Military Standard 498: Software Development and Documentation", con el propósito de que la calidad de los documentos generados sea óptima.

*Para más información sobre el plan de pruebas de la empresa ir al anexo 4. “Plan de pruebas”.*

# Reporte de problemas y acciones correctivas

El reporte de problemas en el cuál se describen las acciones preventivas y correctivas a implementar, debe ser documentado y registrado para asegurar que se hayan implementado y evaluado efectivamente las acciones necesarias para prevenir o corregir problemas o inconformidades durante el desarrollo de software en la fase de construcción.

Dicho reporte se encuentra basado en la norma ISO 9001:2015 la cual establece los requisitos para implementar y mantener un sistema de gestión de calidad en una organización. También establece requisitos específicos para la implementación de acciones correctivas y preventivas y los pasos a seguir para realizar un reporte de lo anterior.

Este reporte debe incluir:

1. Descripción de la inconformidad o problema que requiere acciones: Identificar el problema o la brecha en relación con los requisitos de calidad.
2. Análisis de las causas raíz: Investigar las causas subyacentes de la no conformidad o situación problemática para determinar las razones detrás de ella.
3. Acciones correctivas: Establecer y describir las acciones específicas que se tomarán para corregir la no conformidad y prevenir su recurrencia.
4. Acciones preventivas: Identificar y describir las medidas proactivas que se implementarán para evitar que ocurran problemas similares en el futuro.
5. Responsabilidades y plazos: Asignar roles y responsabilidades claras para la implementación de las acciones correctivas y preventivas, así como establecer plazos específicos para su ejecución.

Los miembros de la unidad de SQA elaborarán estos reportes siguiendo una plantilla elaborada con base en el estándar previamente descrito para registrar aquellos posibles errores encontrados en la fase de construcción del software, que estén relacionados a las acciones preventivas o correctivas implementadas.

*Para más información sobre la plantilla a usar, ir al anexo 6. “Reporte-CAPA”*

# Herramientas, técnicas y metodologías

En esta sección se mencionan las herramientas, técnicas y metodologías que se utilizarán para respaldar el aseguramiento de la calidad del software en la fase de construcción.

## 9.1 Herramientas

● Visual Studio: entorno de desarrollo integrado (IDE) creado por Microsoft. Es ampliamente utilizado para el desarrollo de software en múltiples plataformas y lenguajes de programación.

Esta herramienta se utilizará para agilizar y facilitar el proceso de desarrollo, al igual que para el proceso de desarrollo de las pruebas unitarias.

* Git: Sistema de control de versiones distribuido ampliamente utilizado en el desarrollo de software. Permite rastrear los cambios realizados en los archivos a lo largo del tiempo, coordinar el trabajo de múltiples desarrolladores y facilitar la colaboración en proyectos.

Esta herramienta se utilizará para gestionar y controlar el historial de cambios en el código fuente, el cual permitirá la realización de ramificaciones y fusiones de código, asegurando un control preciso sobre las versiones del software.

* GitHub: Plataforma web basada en la nube que proporciona servicios de alojamiento y control de versiones para proyectos de desarrollo de software utilizando el sistema de control de versiones Git. Permite a los desarrolladores colaborar, compartir y administrar el código fuente de manera eficiente, al tiempo que ofrece características como seguimiento de problemas, gestión de proyectos y control de versiones.

Esta herramienta se utilizará para facilitar el control de versiones del código, el cual permitió a los equipos de desarrollo trabajar de manera conjunta en el proyecto, realizar seguimiento de cambios, gestionar problemas y colaborar eficientemente.

* Code Collaborator: Herramienta de revisión de código colaborativa que permite a los equipos de desarrollo revisar y discutir el código fuente de manera eficiente. Proporciona características como comentarios, seguimiento de cambios y aprobaciones para facilitar el proceso de revisión y mejorar la calidad del código.

Esta herramienta servirá para identificar errores, mejorar la legibilidad, garantizar el cumplimiento de estándares y compartir conocimientos entre los miembros del equipo durante las revisiones de código, facilitando el proceso estructurado y eficiente de estas.

## 9.2 Técnicas

Todas las técnicas por implementar para garantizar la calidad de los elementos establecidos en la fase de construcción se encuentran descritos en el presente plan de SQA. No se requieren técnicas adicionales para poder llevar a cabo la correcta implementación del plan. Por lo tanto, esta sección no aplica.

## 9.3 Metodologías

Todas las metodologías por implementar para garantizar la calidad de los elementos establecidos en la fase de construcción se encuentran descritos en el presente plan de SQA. No se requieren metodologías adicionales para poder llevar a cabo la correcta implementación del plan. Por lo tanto, esta sección no aplica.

# Control de código

El control de código se realiza durante toda la fase de construcción, esto se implementará mediante el control de la configuración. La empresa cuenta con un plan para el control de código donde se describen las actividades que se deben llevar a cabo para esto de manera detallada, este plan está basado en el estándar “ISO/IEC 12207”. En el plan para el control de código se encuentran las actividades que se mencionan a continuación de forma breve.

*Para más información sobre el plan de la empresa, ir al anexo 11. “Plan de control de código”.*

## 10.1 Identificación del código

Para la identificación del código se hará uso de la herramienta Git para el etiquetado de las versiones de código, para el nombramiento de estas versiones se sigue el esquema de versionamiento semántico, que son 3 número separador por puntos de la forma “A.B.C”, donde la A representa los cambios mayores, B nuevas funcionalidades que se hayan integrado y C es para las correcciones de errores. Esto con el fin de poder identificar de forma más rápida los cambios que se realizan.

## 10.2 Localización física del código

El código de software se encontrará en el repositorio de GitHub donde se almacenarán las diferentes versiones del código que hayan sido realizadas posteriormente a la aprobación de la solicitud de cambios.

Otro de los principales usos de Git es que se registran todos los cambios realizados en el código a lo largo del tiempo, lo que permite rastrear y revisar el historial de modificaciones.

## 10.3 Solicitud de cambios

De acuerdo con el plan para el control de código de la empresa, cuando se desee realizar algún cambio en el código se deberá emplear la plantilla para la solicitud de cambios. El comité de control de cambios decidirá posteriormente si la solicitud es aceptada o no, dependiendo de varios factores contenidos en la solicitud de cambios como lo es:

* Descripción del cambio.
* Razón del cambio.
* Impacto del cambio.

Cada una de las solicitudes de cambio deberá ir identificada con el nombre “SC– #”, donde “#” es el número del cambio a realizar, la solicitud de cambios debe contener la documentación que se verá afectada por estos cambios. Al haber terminado la solicitud de cambios, ésta será entregada al comité de control de cambios para que pueda tomar una decisión en cuanto a la aprobación de esa solicitud.

## 

## 10.4 Aprobación o desaprobación de la solicitud

Cada solicitud de cambio que sea aprobada debe ser documentada, esto podrá realizarse en la plantilla de control de cambios que se ha especificado previamente donde se podrá poner si el estado de la solicitud es aprobado o desaprobado. Para la toma de esta decisión se debe tener en cuenta el aseguramiento de la calidad de los cambios que ha sido realizada.

## 10.5 Copias de seguridad

Cuando se hace una copia de seguridad se establece dónde se encontrará ubicada en caso de que se necesite hacer uso de esta, por lo mismo se establece de qué forma serán almacenadas estas copias, ya sea en un servidor local o un repositorio en la nube. De igual forma se establece un plan de mantenimiento para las copias de seguridad, esto con el fin de garantizar que las copias estén actualizadas y sean accesibles. Se lleva un registro del uso de las copias de seguridad, donde se indica la persona que accede junto a la fecha y propósito del acceso.

## 10.6 Distribución de copias

En el caso de distribuir copias del código se seguirán una serie de pasos para garantizar que la distribución sea completa y segura. Para esto primero se empaqueta el código fuente junto con los archivos necesarios para su distribución, posteriormente se utiliza el repositorio de distribución para poner lo que se ha empaquetado que será la copia de la versión de código que se desee distribuir.

De igual forma se documenta la fecha en la que se ha hecho la distribución de la copia y el fin de la realización de esta.

## 10.7 Proceso de liberación de la nueva versión

Para poder liberar una nueva versión primero debe ser aprobada y los cambios deben ser realizados. Se establece una fecha para la liberación de estos cambios, tomando en cuenta ciertas directrices para la realización de esta liberación, que se toman del estándar IEEE-828-2012 y especifican en el plan de control del código mencionado anteriormente.

## 10.8 Aseguramiento de los cambios

Para poder realizar los cambios de forma segura se realizan un conjunto de buenas prácticas que se detallan de manera más específica en el plan del control del código. Estas prácticas son las siguientes:

* Control de versiones: Se hace un control de versiones con el fin de administrar los cambios del código de forma segura, ya que permite revertir los cambios si hay algo mal y se tiene un historial de los cambios realizados.
* Desarrollo basado en ramas: Implementar el desarrollo basado en ramas permite trabajar en ramas separadas para realizar los cambios, de esta forma se pueden probar los cambios sin afectar directamente a la versión principal del código.
* Pruebas de seguridad: Al hacer un cambio significativo en el código, se deben hacer pruebas de seguridad para detectar posibles puntos débiles y asegurar que los cambios no introduzcan riesgos de seguridad.
* Control de acceso: Se limita el acceso al repositorio de código para que solo los desarrolladores autorizados tengan permisos para realizar cambios en el código.
* Monitorización: Monitorear el código para detectar cualquier actividad sospechosa después de implementar los cambios, esto permite identificar problemas de seguridad que puedan ocurrir.

# Control de medios

FemmeCode cuenta con un plan para el control de medios basado en la norma ISO/IEC 27001. En el plan se especifica cómo se almacenan los documentos, manuales, planes y otros elementos necesarios para el desarrollo. Por lo mismo, se describen actividades que se llevan a cabo para realizar el control de la mejor manera posible.

*Para más información sobre el plan de la empresa, ir al anexo 12.” Plan de control de medios”.*

* Almacenamiento: Los documentos que se mencionan en el plan se almacenan en un repositorio en la nube, donde se categorizan en diferentes carpetas indicando qué es lo que contienen. Esto se hace para que las personas autorizadas tengan acceso a estos documentos o medios.
* Respaldo: La empresa FemmeCode cuenta con un servidor local donde se respaldan los documentos del proyecto que esté realizando la empresa, se hace el respaldo cada que se tiene un cambio significativo en algún medio. De igual forma también se hace el respaldo en un disco duro externo para poder tener dos opciones para acceder al respaldo.

Estas y otras actividades se describen de manera más detallada dentro del plan para el control de medios.

## 11.1 Acceso no autorizado

Para garantizar que los medios almacenados no son accedidos por cualquier persona se establecen una serie de métodos para asegurar estos medios y documentos. Algunos de los métodos que se encuentran en el plan se describen a continuación de forma más breve, sin embargo, en el plan se especifica todo de manera más concisa.

* Control de acceso a los medios almacenados: El acceso al repositorio en la nube estará limitado únicamente al personal del proyecto, esto con el fin de evitar que algún documento importante sea distribuido a personas que no deben tener acceso.
* Control de acceso a los respaldos: Para acceder a los respaldos realizados se deberá hacer una solicitud que sólo el jefe de unidad de SQA podrá aprobar, en caso de que se apruebe se brindará acceso al respaldo solicitado. En el caso del acceso de los respaldos del disco duro externo, este tendrá contraseña y se mantendrá en un lugar seguro cuando no esté en uso.
* Gestión de identidad y privilegios: Es esencial tener un sistema adecuado de gestión de identidad y privilegios para garantizar que solo las personas autorizadas del personal de SQA tengan acceso a los medios almacenados. Por lo mismo, se asignan roles y privilegios específicos a los usuarios, estos se deben revocar cuando sea necesario (por ejemplo, cuando un empleado deja la organización) y revisar periódicamente los permisos de acceso para garantizar que sigan siendo apropiados.

## 11.2 Daño o degradación inadvertida

Con el fin de evitar o minimizar el daño o degradación inadvertida en el control de medios se implementan varias medidas y se verifica que su aplicación sea la correcta, por lo que se documenta por parte del personal de SQA. De igual forma, esta sección se detalla mejor en el plan para el control de medios, sin embargo, a continuación, se mencionan algunas de las medidas que se implementan.

* Correcto almacenamiento: Los elementos en los que se almacenan los medios deben mantenerse en un entorno adecuado para prevenir daños o degradación. Esto implica mantenerlos en condiciones ambientales controladas, como temperatura y humedad óptimas, y protegerlos de elementos como polvo, luz solar directa, campos magnéticos fuertes u otras fuentes de interferencia electromagnética.
* Capacitación del personal: Proporcionar capacitación adecuada al personal sobre las mejores prácticas de manejo y almacenamiento de documentos. Educa al personal sobre los riesgos potenciales y cómo evitar daños o degradación inadvertida.
* Manipulación cuidadosa de los medios: Evitar realizar modificaciones accidentales en los documentos o medios almacenados.

# Recopilación, mantenimiento y retención de registros

Los procedimientos siguientes se encuentran descritos conforme al plan “Registros” que la empresa tiene referente a la gestión de la configuración, el cual está basado en la norma ISO/IEC 12207:1997.

*Para más información sobre el plan de la empresa, ir al anexo 7. “Plan de registros”*

## 12.1 Recopilación de registros

Para la recopilación de registros, se deberá tener en cuenta lo siguiente:

* Los registros deben documentar los cambios realizados, los resultados de las pruebas, los problemas identificados, las decisiones tomadas y otros aspectos relevantes del proceso de construcción.
* Los registros deben incluir información sobre la configuración del software, como versiones utilizadas, componentes involucrados, ambientes de desarrollo y pruebas, entre otros.
* ​​Los registros deberán haber estado validados y verificados para que puedan cumplir los estándares y requerimientos impuestos por el cliente.
* Los registros deberán contener información relevante al finalizar el proyecto de software, con el fin de mantener registros históricos.

Los documentos por recopilar deberán ser:

* Registro de rastreo
* Plan de pruebas unitarias
* Plan de verificación y validación del software
* Plan de gestión de la configuración de software
* Reporte de actividades

## 12.2 Mantenimiento de registros

De acuerdo con el plan de la empresa, los registros deben ser gestionados para asegurar su integridad, disponibilidad, accesibilidad y protección contra modificaciones no autorizadas. Esto implica tener controles y procedimientos para el almacenamiento, respaldo, recuperación y conservación de los registros durante un período de tiempo adecuado.

Para lograr lo anterior, los registros deberán ser almacenados en un repositorio privado de Github oficial de la empresa.

Para tener un control de acceso sobre estos registros del software, el jefe de la unidad de desarrollo y mantenimiento junto con el jefe de la unidad de SQA y el líder del proyecto podrán acceder a los registros cada vez que lo requieran sin realizar modificaciones no autorizadas.

## 12.3 Retención de registros

El plan de la empresa menciona que es importante garantizar que los registros se mantengan de forma segura y protegida contra modificaciones no autorizadas o pérdida de datos durante el período de retención. Para ello se implementan medidas de seguridad como el control de accesos y sistemas de almacenamiento confiables, tales como los que se mencionan en el punto 12.2.

# Capacitación

El plan de capacitación a implementar de la empresa FemmeCode se encuentra basado en el modelo “Capability Maturity Model Integration for Development" (CMMI-DEV) de acuerdo con el área de proceso "Capacitación organizacional”, la cual se centra en el desarrollo y la implementación de un plan de capacitación adecuado para el equipo.

*Para más información sobre el plan ir al anexo 8. “Plan de capacitación”*

Dicho plan de capacitación incluye:

Identificación de necesidades: Identifica las habilidades y conocimientos requeridos por el equipo de desarrollo para alcanzar los objetivos del proyecto y cumple con los niveles de madurez establecidos por el modelo CMMI-DEV.

Diseño de programas de capacitación: Programas de capacitación específicos que aborden las brechas identificadas en las habilidades y conocimientos del equipo.

Asignación de recursos: Recursos necesarios para implementar los programas de capacitación, incluyendo el tiempo, el presupuesto y los instructores o facilitadores requeridos.

Implementación y seguimiento: Métricas de seguimiento para medir el impacto de la capacitación en las habilidades y conocimientos del equipo, así como en la calidad y el desempeño del proceso de desarrollo.

Mejora continua: Enfoque de mejora continua para la capacitación. Se revisan regularmente los resultados de la capacitación y se realizan ajustes en los programas según sea necesario. También se documentan las lecciones aprendidas para utilizarlas en futuros planes de capacitación.

Las habilidades requeridas para lograr la correcta implementación del plan de aseguramiento de la calidad de acuerdo con los roles establecidos en la unidad de SQA son:

|  |  |
| --- | --- |
| **Tareas** | **Conocimientos requeridos** |
| Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual. | * Conocer el plan de desarrollo actual, el plan de gestión y las estrategias para implementar de manera adecuada el plan de SQA. * Conocimiento del estándar IEEE 828-2012 para la correcta asignación de roles. |
| Realizar una sesión de revisión para verificar los componentes previamente recolectados. | * Conocimiento del estándar IEEE Std 1028-2008 * Conocimiento del plan de revisiones de la empresa. |
| Documentar la revisión de los componentes previamente revisados. | * Conocimiento del estándar IEEE Std 829-2008 * Conocimiento de las checklist y plantillas usadas por la empresa para la documentación. * Conocimiento de las herramientas y técnica usadas para la documentación |
| Realizar las modificaciones necesarias a los componentes revisados. | * Conocimiento del estándar IEEE Std 829-2008 * Conocimiento de los lenguajes de programación utilizados en los elementos a revisar. * Conocimiento técnico de los elementos a revisar |
| Definir y aplicar pruebas unitarias para verificar que el funcionamiento de cada componente esté acorde con la parte detallada del proceso de construcción. | * Conocimiento del estándar IEEE Std 1008-1987 * Conocimiento de los requerimientos funcionales * Conocimiento del plan de pruebas unitarias de la empresa * Conocimiento técnico requerido para verificar el funcionamiento de cada componente. |
| Corregir los defectos encontrados hasta lograr pruebas unitarias exitosas (sin defectos). | * Conocimiento del estándar IEEE Std 1008-1987 * Conocimiento del plan de pruebas unitarias de la empresa |
| Realizar la sesión de revisión para la verificación del Plan de pruebas unitarias. | * Conocimiento del estándar IEEE Std 1028-2008 * Conocimiento del plan de revisiones de la empresa. * Conocimiento del plan de pruebas unitarias de la empresa |
| Actualizar el Registro de Rastreo, incorporando los componentes construidos o modificados. | * Conocimiento del estándar “IEEE Std 1012-2016” * Conocimiento de las plantillas y checklist a usar * Conocimiento de las herramientas y técnicas a usar por la empresa. |
| Corregir los defectos encontrados en el Registro de Rastreo con base en el Reporte de Verificación y obtener la aprobación de las correcciones. | * Conocimiento de la documentación a implementar del registro de rastreo. * Conocimiento del estándar y plantilla a usar para el reporte de verificación * Conocer el plan de verificación y validación de la empresa |
| Incorporar Componentes y Registro de Rastreo como líneas base a la Configuración de Software. | * Conocimiento del estándar IEEE Std 1028-2008 * Conocer el documento de control de la configuración del software |
| Elaborar el Reporte de Actividades. | * Conocimiento del estándar “IEEE Std 1012-2016” * Conocimiento de la plantilla para elaborar el reporte de actividades * Conocimiento de la documentación para añadir el reporte de actividades. |

*Tabla 4.0 Conocimientos requeridos por la organización*

# Gestión de riesgos

La empresa FemmeCode tiene un plan para la gestión de riesgos basado en la norma ISO 31000 donde se describen las áreas de riesgo en la fase de construcción, igual se menciona las razones por las cuales el personal de SQA piensa que es un área de riesgo, las personas con las que ha consultado el personal de SQA, así como las razones por las que se cree que esta es un área de riesgo.

*Para más información sobre el plan de la empresa ir al anexo 13.” Plan de gestión de riesgos”*

En el plan para la gestión de riesgos se tiene la forma en la que se documentan estos riesgos, de forma más detallada, aquí se pone la descripción, su impacto, quien lo supervisa, entre otros aspectos. Entre los principales riesgos que se describen en el plan, son los que se describen brevemente a continuación.

* Errores de programación: Estos errores pueden causar fallas en el software, comportamientos inesperados o vulnerabilidades de seguridad.
* Falta de realización de pruebas adecuadas: Si no se realizan pruebas exhaustivas durante la fase de construcción, es probable que se pasen por alto errores y problemas en el código. La falta de pruebas adecuadas puede resultar en software con errores y con un rendimiento deficiente.
* Vulnerabilidades de seguridad: Durante la fase de construcción, pueden introducirse vulnerabilidades de seguridad en el código. Esto puede incluir prácticas de codificación inseguras, falta de validación de entrada, gestión inadecuada de la memoria y otros problemas que podrían ser explotados por atacantes.
* Falta de control de versiones de código: Si no se tiene un adecuado control de versiones, puede ser difícil rastrear los cambios realizados en el código durante la fase de construcción. Esto dificulta la reversión de cambios no deseados, la colaboración en equipo y la gestión de versiones del software.
* Incumplimiento de estándares: Si los desarrolladores no siguen estándares y buenas prácticas de codificación, como el uso de patrones de diseño, estructuras de datos adecuadas y técnicas de programación seguras, el software resultante puede ser propenso a errores y más difícil de mantener.

En caso de que los riesgos cambien, se tendrán que actualizar y documentar de la forma establecida en el plan para la gestión de riesgos.

# Anexos

Toda la documentación referenciada y utilizada para la elaboración del plan de aseguramiento de la calidad del software se encuentra almacenada en el repositorio de Github del proyecto, al igual que los estándares referenciados.

Link: <https://github.com/ver0rivera/Plan-SQA>

1. Plantilla Registro de rastreo

Link:

<https://github.com/ver0rivera/Plan-SQA/blob/main/Plantillas/Reportes/PlantillaReporteRastreo.docx>

1. Plan de revisión

Link:

<https://github.com/ver0rivera/Plan-SQA/blob/main/FemmeCode/PlanRevision.pdf>

1. Plantillas para plan de pruebas

Link:

<https://github.com/ver0rivera/Plan-SQA/tree/main/Plantillas/Reportes/Plantillas%20de%20pruebas>

1. Plan de pruebas

Link:

<https://github.com/ver0rivera/Plan-SQA/blob/main/FemmeCode/PlanPruebas.pdf>

1. Plantilla del informe de estado del proyecto

Link:

<https://github.com/ver0rivera/Plan-SQA/blob/main/Plantillas/InformeEstadoProyecto.pdf>

1. Reporte-CAPA

Link:

<https://github.com/ver0rivera/Plan-SQA/blob/main/Plantillas/Reportes/ReporteCAPA_Plantilla.pdf>

1. Plan de registros

Link:

<https://github.com/ver0rivera/Plan-SQA/blob/main/FemmeCode/PlanRegistros.pdf>

1. Plan de Capacitación

Link:

<https://github.com/ver0rivera/Plan-SQA/blob/main/FemmeCode/PlanCapacitacion.pdf>

1. Plan de verificación y validación

Link:

<https://github.com/ver0rivera/Plan-SQA/blob/main/FemmeCode/Plan%20de%20verificaci%C3%B3n%20y%20validaci%C3%B3n%20de%20Software.pdf>

1. Plantilla verificación y validación

Link:

<https://github.com/ver0rivera/Plan-SQA/blob/main/Plantillas/Reportes/Reporte%20de%20Verificaci%C3%B3n%20y%20Validaci%C3%B3n%20del%20Software.pdf>

1. Plan de control de código

Link:

<https://github.com/ver0rivera/Plan-SQA/blob/main/FemmeCode/PlanControlCodigo.pdf>

1. Plan de control de medios

Link:

<https://github.com/ver0rivera/Plan-SQA/blob/main/FemmeCode/PlanControlMedios.pdf>

1. Plan de gestión de riesgos

Link: <https://github.com/ver0rivera/Plan-SQA/blob/main/FemmeCode/PlanGestionRiesgos.pdf>