

# **Plan de Aseguramiento de la calidad para RiskMap**

**Fecha de elaboración:**

30 de mayo de 2022

Maestro: Edwin Jesús León Bojórquez

**Versión 0.3**

**Miembros del equipo:**

**Ana Pérez**

**Lenin Rosas**

**Pablo Rosas**

# Control de Documentación

## Control de Configuración

Título:	Plan de Aseguramiento de la calidad para RiskMap
Referencia:	<a href="https://github.com/PabloR9080/Documentacion-PlanSQA">https://github.com/PabloR9080/Documentacion-PlanSQA</a>
Autor:	Pablo Rosas
Fecha:	08 de mayo del 2022

## Histórico de versiones

Versión	Fecha	Estado	Responsable	Nombre de archivo
0.3	27/05/2022	B	Equipo	SQAPlanV0.3.docx
0.2	18/05/2022	A	Equipo	SQAPlanV0.2.docx
0.2	15/05/2022	B	Equipo	SQAPlanV0.2.docx
0.1	08/05/2022	A	Equipo	SQAPlanV0.1.docx
0.1	08/05/2022	B	Equipo	SQAPlanV0.1.docx

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

## Histórico de cambios

Versión	Fecha	Cambios
0.3	27/05/2022	Se realizaron las correcciones necesarias al apartado de Propósito, Gestión, Documentación, Estándares, prácticas, convenciones y métricas, Revisiones y auditorías, Pruebas y Control del código de acuerdo a los comentarios de la versión 0.2 del documento.
0.2	18/05/2022	Se definieron las herramientas, así como las metodologías a utilizar. Los procesos para el control de código, medios y documentos finales se han especificado en esta versión. Por último, se han agregado las acciones correctivas, así como los métodos de entrenamiento, en el caso necesitar una preparación para los integrantes de la organización.
0.2	15/05/2022	Se realizaron los puntos 10,11 y 12 remarcando los puntos del control de distintos objetos dentro del plan.
0.1	09/05/2022	Creación del repositorio para el control de versiones en Github y vinculación de los documentos referenciados con el repositorio.
0.1	08/05/2022	Se agregan algunas herramientas, en este caso herramientas que serán utilizadas para las pruebas

0.1	09/05/2022	Se realizan los puntos 6-7 para definir cómo se realizarán las auditorías y revisiones, así como también las pruebas que se realizarán a lo largo del plan de aseguramiento de la calidad para el software RiskMap.
0.1	09/05/2022	Se ha realizado la definición del punto 1-5 del documento en donde se describe el propósito del documento actual y los documentos referenciados para la creación de este. Por otra parte, se empieza a describir los puntos de gestión, documentación y estándares, prácticas, convenciones y métricas para comenzar a definir el plan de SQA.
0.1	08/05/2022	Se define el proyecto sobre el cual se aplicará el plan de SQA.

## Tabla de contenidos

Control de Documentación .....	2
1. Propósito .....	6
2. Documentos referenciados.....	6
3. Gestión.....	8
3.1 Organización .....	8
Roles y responsabilidades .....	10
3.2 Tareas .....	11
3.2.1 Tareas generales.....	11
3.2.2 Requerimientos .....	12
3.2.3 Análisis y Diseño .....	13
3.2.4 Implementación .....	14
3.2.5 Mantenimiento .....	15
4. Documentación .....	16
4.1 Especificación de Requerimientos de Software (ERS) .....	17
4.2 Descripción de Diseño de Software (DDS) .....	17
4.3 Plan de Pruebas del Sistema .....	18
4.5 Manual de Usuario .....	19
4.6 Registro de Rastreo .....	19
4.7 Manual de Mantenimiento de Software (MMS) .....	20
4.8 Manual de Operación.....	21
4.9 Plan de Pruebas de Integración.....	21
4.10 Plan de Gestión de la Configuración del Software (PGCS) .....	21
5. Estándares, prácticas, convenciones y métricas .....	22
5.1 Propósito. ....	22
5.2 Contenido .....	22
5.2.1 Fase de requerimientos.....	22
5.2.2 Fase de análisis y diseño .....	23
5.2.3 Fase de Construcción .....	23
5.2.4 Fase de Integración y Pruebas .....	24
5.3 Documentación .....	24
5.4 Métricas.....	24
6. Revisiones y auditorías.....	25
6.1 Propósito .....	25
6.2. Requisitos mínimos.....	25
6.2.1 Revisión de la especificación de software .....	25
6.2.2 Revisión del plan de pruebas del sistema .....	25

6.2.3 Revisión del manual de usuario .....	26
6.2.4 Revisión del Registro de rastreo .....	26
6.2.5 Revisión del diseño .....	26
6.2.6 Revisión del plan de pruebas de integración .....	26
6.2.7 Revisión del manual de operación .....	27
6.2.8 Revisión del manual de mantenimiento .....	27
6.2.9 Revisión del plan de gestión de la configuración del software .....	27
7. Pruebas .....	27
7.1 Proceso de realización de las pruebas: .....	27
7.2 Pruebas de unidad .....	29
7.3 Pruebas de integración .....	29
7.4 Pruebas de sistema .....	29
8. Reporte de problemas y acciones correctivas .....	30
8.1 Responsabilidades .....	30
8.2 Contenido .....	31
9. Herramientas, técnicas y metodologías .....	31
9.1 Herramientas .....	31
9.2 Técnicas .....	35
9.3 Metodologías .....	36
10. Control de código .....	37
10.1 Solicitudes de cambio .....	37
10.2 Aseguramiento de la calidad de los cambios .....	37
10.3 Proceso de liberación .....	38
10.4 Localización del código .....	38
11. Control de medios .....	39
11.1 Acceso no autorizado .....	39
11.2 Daño o degradación desapercibida .....	40
12. Recolección de registros, mantenimiento y retención .....	40
12.1 Recolección de registros .....	41
12.2 Mantenimiento de registros. ....	41
12.3 Retención de registros. ....	41
13. Entrenamiento .....	42
14. Anexos .....	43

## 1. Propósito

En este documento se describen las actividades que el personal de aseguramiento de la calidad de software (SQA) debe llevar a cabo con el objetivo de dar seguimiento al proceso de desarrollo y reportar las deficiencias detectadas en los procesos y productos generados.

Este Plan de Aseguramiento de la Calidad de Software será aplicado en el desarrollo del proyecto Riskmap, una aplicación que ayuda a las personas con la identificación de las zonas de más peligrosas de su localidad por medio de un mapa delictivo que es generado con los reportes de los delitos registrados por los propios usuarios de la aplicación.

En el siguiente plan se especifican las estrategias para controlar la calidad de los productos generados en todas las etapas del ciclo de vida (planeación, requerimientos, análisis y diseño, implementación y mantenimiento), incluyendo las verificaciones y validaciones que aplican a cada uno de ellos y su conformidad con los lineamientos establecidos en el modelo de procesos de software.

Los elementos cubiertos por este plan son:

- Especificación de Requerimientos de Software
- Descripción de Diseño de Software
- Código y Pruebas
- Planes y Reportes de Verificación y Validación
- Plan de Gestión de la Configuración de Software
- Documentación desarrollada

Este documento está dirigido al equipo de aseguramiento de calidad de software, 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.

## 2. Documentos referenciados

**IEEE Std 1028-2008. Standard for Software Reviews and Audits.** Este estándar define los requerimientos mínimos aceptables para las revisiones y auditorías

sistemáticas aplicables a la adquisición, suministro, desarrollo y mantenimiento del software.

**IEEE Std 830-1998. Recommended Practice for Software Requirements Specifications.** Este estándar describe el contenido y cualidades que debe contener una buena especificación de software.

**IEEE Std 1016-2009. Standard for Information Technology – Systems Design – Software Design Descriptions.** Este estándar describe los diseños de software y establece el contenido de la información y organización de una descripción del diseño de software.

**IEEE Std 829-1998. Standard for Software Test Documentation.** Este estándar describe un conjunto de documentos de pruebas de software básicos y especifica el formato y contenido de los documentos de pruebas individuales.

**IEEE Std 1063-2001. Standard for Software User Documentation.** Este estándar establece los requerimientos mínimos de la estructura y contenido de la información de la documentación del usuario.

**IEEE Std 828-1990. Standard for Software Configuration Management Plans.** Este estándar establece el contenido mínimo requerido para el Plan de la gestión de la configuración del software y define las tareas específicas a realizar y sus requerimientos para cualquier parte del ciclo de vida de un producto de software.

**IEEE Std. 1008. Standard for Software Unit Testing.** Este estándar define un acercamiento integrado a las pruebas unitarias sistemáticas y documentadas.

**ISO/IEC 14764-2006. Software Life Cycle Processes – Maintenance.** Este estándar define las actividades y tareas del mantenimiento de software, además proporciona los requerimientos para la planificación del mantenimiento.

**ISO/IEC 15939-2007. Systems and software engineering – Measurement process.** Este estándar define un proceso de medición aplicable a las disciplinas de ingeniería y gestión de los sistemas y del software.

**IEEE Std 1012-2016. Standard for System, Software and Hardware Verification and Validation.** Este estándar abarca todos los procesos del ciclo de vida del

sistema, software y hardware para determinar si los productos de una determinada actividad satisfacen los requisitos y necesidades del usuario.

**ISO/IEC 29119. Software Testing Standard.** Este estándar contiene un conjunto de documentos definidos internacionalmente que tienen como objetivo cubrir todo el ciclo de vida de las pruebas de sistemas de software incluyendo aspectos relacionados a la organización, gestión, diseño y ejecución de las pruebas. Disponible en:

Sánchez, P., (2015), *Pruebas de Software. Fundamentos y Técnicas*. Recuperado el 16/05/2022 <https://github.com/PabloR9080/Documentacion-PlanSQA/blob/main/Documentos-Estandares/Pruebas-Software-Fundamentos-Tecnicas.pdf>

Martin, R., (2009), *Clean Code A Handbook of Agile Software Craftsmanship*. Prentice Hall. <https://github.com/PabloR9080/Documentacion-PlanSQA/blob/main/Documentos-Estandares/CleanCode.pdf>

Oracle, (1997), *Java Code Conventions*. <https://github.com/PabloR9080/Documentacion-PlanSQA/blob/main/Documentos-Estandares/Java-code-conventions.pdf>

### **3. Gestión**

#### **3.1 Organización**

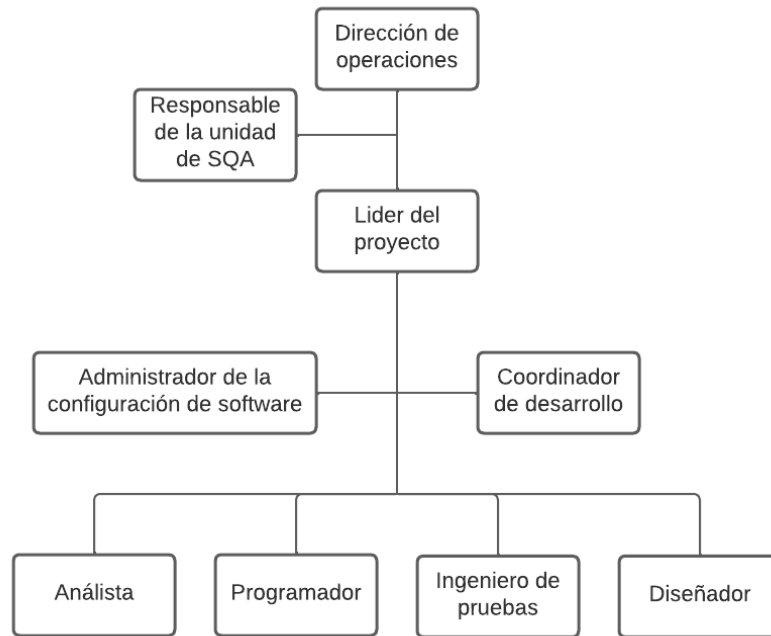
La actividad de SQA se desglosa en distintas tareas a lo largo del ciclo de vida del software a realizar. El propósito de este equipo de acuerdo con el plan es minimizar los errores que el equipo comete a través de las tareas que son descritas en este plan.

Por lo cual, una organización es requerida para la coordinación en realizar las tareas, generando así una noción sobre las actividades que sea han realizado, dejando pendiente las actividades restantes.

El equipo de SQA es un grupo de personas con mayor importancia sobre el proyecto ya que, el negocio no podría avanzar exitosamente sin este. Es vital la comunicación entre los integrantes de este para que prevalezca el éxito en el proyecto.

#### **Organigrama**





Los elementos del organigrama que interactúan directamente con el elemento de SQA son la Dirección de operaciones y el Líder del proyecto. Esta interacción se da cuando el Responsable de la unidad de SQA detecta que la calidad del producto está en peligro, reportando las anomalías y no conformidades al siguiente nivel (Líder del proyecto) o al nivel más alto (Dirección de operaciones) en la cadena de liderazgo en la organización del proyecto. Estas tres entidades, trabajan en conjunto para establecer un programa de calidad para el proyecto, así como también llevar a cabo la revisión y aprobación del Plan de Aseguramiento de Calidad (SQAP).

Una vez identificadas las actividades de SQA a llevarse a cabo y aprobado el SQAP, el Líder del proyecto se encarga de aplicar el programa de calidad delegando la responsabilidad de supervisar que los roles del último nivel del organigrama realicen sus actividades de acuerdo al plan de calidad previamente establecido. El Líder de proyecto trabaja con cada uno de los roles en la definición de los planes necesarios para las diferentes etapas del ciclo de vida, así como los reportes de control del proceso resultantes.

El Analista, Programador, Ingeniero de Pruebas y Diseñador implementan la calidad en los procesos y productos que le corresponden a cada uno de acuerdo al Plan de SQA y los documentos de planeación, resolviendo y dando seguimiento a cualquier asunto de calidad que tenga relación con los productos de salida. Estos roles deberán

también comentar acerca del Plan de SQA, haciendo llegar sus observaciones y resultados tras realizar las actividades que les fueron asignadas al Líder del Proyecto que se encuentra al pendiente para atender a los reportes y comunicarlo de vuelta a los niveles superiores de la organización.

Por último, el Administrador de la configuración implementa las practicas, procesos y procedimientos definidos en el Plan de Gestión de la Configuración de la Configuración y en otros planes o documentos complementarios, definidas en conjunto con la Unidad de desarrollo de software y la unidad de SQA. De igual manera, comenta el plan de SQA para el proyecto e implementa las actividades de calidad definidas en este, resolviendo también los problemas detectados por SQA relacionados con SCM.

### **Roles y responsabilidades**

A continuación, se presentan las responsabilidades de cada uno de los roles desde la perspectiva de aseguramiento de la calidad:

Rol	Abreviatura	Responsabilidad
Dirección de operaciones	DIOP	Encargado de establecer un programa de calidad para los procesos, revisión y aprobación de planes de aseguramiento de calidad. Además, deberá seleccionar al personal responsable del aseguramiento de la calidad e identificar los factores o metas de calidad a cubrir.
Responsable de la unidad de SQA	RUSQ	Rol responsable de verificar y validar que las actividades sean realizadas acorde a lo establecido en el plan de SQA.
Líder de proyecto	LPR	Aplicar el programa de calidad, identificar las actividades necesarias para cumplir, revisar y aprobar el plan de calidad del proyecto. Atiende a los reportes de problemas generados durante verificaciones y auditorias y mantiene los planes asociados al proyecto.

Coordinador de desarrollo	CD	Verificar que los roles AN, PR e IPR cumplan con las actividades designadas de acuerdo al plan de calidad establecido. Realizar actividades de capacitación y control en el equipo de desarrollo.
Analista	AN	Dar revisión al plan de aseguramiento de calidad del proyecto, así como identificar, implementar y evaluar los factores de calidad definidos en el plan SQA.
Programador	PR	Implementar las especificaciones de diseño de software, revisar y comentar el plan de SQA, verificar los factores de calidad a implementar en el sistema.
Ingeniero de pruebas	IPR	Implementar el programa de calidad definido para el proyecto, verificar y validar cada una de las funcionalidades acordes a la especificación de requisitos.
Administrador de la configuración de software	ACS	Llevar a cabo todas las prácticas relacionadas con la administración de la configuración y control de cambios.
Diseñador	DI	Encargado de realizar las actividades de análisis y diseño. Implementa las prácticas y procesos a tomar en cuenta en el diseño del software definidos en el plan de SQA.

## 3.2 Tareas

### 3.2.1 Tareas generales

**Criterio de entrada:** ninguno.

**Criterio de salida:** ninguno.

#### Tareas

Actividad	Roles	Responsabilidades
-----------	-------	-------------------

Realizar los planes, definir los procesos para que el proyecto tenga un nivel de calidad aceptable.	DIOP	DIOP: encargado de poner en marcha la creación de los planes, empezar a definir los procesos y crear los roles de los responsables.
Revisar y evaluar la calidad de las actividades del proyecto conforme a los lineamientos del aseguramiento de la calidad.	LPR	LPR: verifica que las actividades que se están realizando estén correctas y ordenadas.
Planificar todas las tareas definidas en el plan de SQA.	RUSQ	RUSQ: se encarga de revisar que todas las tareas estén siendo incluidas en el plan, así como la asignación de actividades y roles a los integrantes.
Mantener contacto y cercanía con el cliente a lo largo del ciclo de vida del proyecto.	LPR	LPR: se encarga de realizar juntas para mantener al usuario al tanto sobre los avances del proyecto.

### 3.2.2 Requerimientos

**Criterio de entrada:** ninguno.

**Criterio de salida:** documento de especificación de requerimientos y plan de pruebas.

#### Tareas

Actividad	Roles	Responsabilidades
Asegurar que los stakeholders e integrantes de la compañía estén presentes en los procesos para la educación de los requisitos.	LPR	LPR: encargado de agendar las citas con los stakeholders.
Asegurar que los integrantes del equipo reciban una	CD	CD: encargado de entregarle la capacitación al equipo de la fase de requerimientos.

capacitación en el caso de ser necesario.		
Asegurar que los miembros del equipo que estará involucrado en la fase de requerimientos tengan los conocimientos suficientes para llevar a cabo esta fase.	CD	CD: verifica que los participantes involucrados en la fase de requerimientos tengan los conocimientos adecuados a través de un cuestionario.
Monitorear la ejecución de las técnicas de elicitación elegidas.	LPR, CD	LPR: verifica que las técnicas a realizar estén de acuerdo a las actividades planeadas. CD: se encarga del monitoreo para que la actividad de educación de requisitos sea llevada a cabo.
Auditar el documento de especificación de requerimientos y plan de pruebas.	RUSQ, LPR	RUSQ: se encarga que la auditoría se lleve a cabo conforme a las revisiones descritas en el plan de SQA. LPR: realiza la auditoría a partir de lo descrito en el Plan de SQA.
Asegurarse que se lleven a cabo las revisiones y auditorías necesarias documentadas en el punto 6 del actual plan de SQA.	RUSQ, LPR	RUSQ: verifica que las revisiones y auditorías se están llevando a cabo como lo describe el punto 6 del plan. LPR: lleva a cabo las revisiones y auditorías necesarias.
Evaluar los productos generados en la fase de requerimientos para su validación.	LPR, ACS	LPR: asegura los productos tengan una correctitud a partir de su validación. ACS: revisa y almacena el producto generado en la fase de requerimientos.

### 3.2.3 Análisis y Diseño

**Criterio de entrada:** documento de especificación de requerimientos y plan de pruebas.

**Criterio de salida:** documento de especificación de diseño, documento de diseño detallado.

### Tareas

Actividad	Roles	Responsabilidades
Asegurar que los integrantes del equipo reciban una capacitación en el caso de ser necesario.	CD	CD: encargado de entregarle la capacitación al equipo de la fase de requerimientos.
Evaluar y revisar el proceso preliminar de documentar la especificación de diseño.	LPR	LPR: realiza la revisión sobre el documento preliminar de la especificación de diseño.
Asegurarse que se lleven a cabo las revisiones y auditorías necesarias documentadas en el punto 6 del actual plan de SQA para la fase.	RUSQ, LPR	RUSQ: verifica que las revisiones y auditorías se están llevando a cabo como lo describe el punto 6 del plan. LPR: lleva a cabo las revisiones y auditorías necesarias.

### 3.2.4 Implementación

**Criterio de entrada:** documento de especificación de requerimientos, plan de pruebas y documento de especificación de diseño.

**Criterio de salida:** código fuente, pruebas unitarias, entorno para despliegue.

### Tareas

Actividad	Roles	Responsabilidades
Asegurar que el programador cuenta con los conocimientos suficientes sobre el proyecto y la implementación a realizar.	CD, PR	CD: realiza un cuestionario al programador para asegurar que cuenta con los conocimientos. PR: realiza el cuestionario impuesto por el CD.
Revisar que el proceso de implementación sea ejecutado de manera correcta siguiendo	CD	CD: verifica que las actividades que realiza el programador estén de acuerdo

el documento de especificación de requerimientos.		al documento de especificación de requerimientos.
Evaluar la consistencia del proceso con lo que se tiene documentado en el plan de SQA.	LPR, RUSQ	LPR: encargado de llevar un control de los reportes de problemas durante la fase de implementación. RUSQ: verifica que el proceso que se está llevando a cabo está de acuerdo al plan de SQA.
Realizar revisiones periódicas en el proceso.	LPR, CD	LPR: se encarga de agendar y ejecutar las revisiones periódicas. CD: reúne los que se tiene hecho del producto hasta el momento de la revisión para llevarla a cabo.
Auditar el control de cambios en el código fuente o los productos de software que se hayan generado en iteraciones anteriores.	ACS	ACS: revisa el código fuente y genera el reporte de revisión, cuando se considera pertinente, almacena la versión del producto.

### 3.2.5 Mantenimiento

**Criterio de entrada:** peticiones de cambio, usuarios para usar el software y reportar los problemas, plan de pruebas.

**Criterio de salida:** problemas en el código y cambios realizados para pasar a producción.

#### Tareas

Actividad	Roles	Responsabilidades
Realizar una revisión posterior sobre el código fuente resultante de la implementación.	ACS	ACS: realiza la revisión para poder pasar al mantenimiento. Reporta los errores encontrados al LPR.

Recabar las peticiones o cambios necesarios para realizar el mantenimiento.	LPR	LPR: encargado de recolectar las peticiones de cambio, definir los cambios que se realizarán y los errores por reparar.
Asegurarse que los involucrados en la fase tengan los documentos necesarios para empezar el mantenimiento.	LPR, PR	LPR: junta los documentos que se llevarán a cabo en el mantenimiento y se implementarán en la siguiente versión del proyecto. PR: recibe y analiza los documentos necesarios antes de realizar el mantenimiento.
Realiza los cambios necesarios sobre el código fuente.	PR	PR: lleva a cabo el mantenimiento a partir de los documentos entregados por el LPR.
Revisar y controlar los cambios necesarios se estén realizando sobre el código fuente.	CD	LPR: entrega los documentos al CD, mismos que se fueron entregados al PR. CD: revisa que los cambios que se están haciendo están de acuerdo con los documentos entregados al PR.
Realizar una verificación antes de proceder con la validación sobre los cambios hechos.	ACS	ACS: revisa que los cambios hechos, estén correctos y no haya ningún defecto en el código fuente.
Asegurarse que las pruebas sean realizadas conforme al plan de pruebas definido y no haya ningún error en el proceso.	CD, IPR	CD: se asegura que el IPR sigue la ejecución de las pruebas descritas en el plan de pruebas para el sistema RiskMap. IPR: lleva a cabo las pruebas como lo dicta el plan de pruebas.

#### 4. Documentación

La implementación de todo proceso de desarrollo trae consigo la elaboración de documentos cuya calidad debe de ser asegurada. Para esto, será importante definir



la manera en la que serán revisados o auditados, así como los criterios de revisión o auditoría que deberán aplicarse.

#### **4.1 Especificación de Requerimientos de Software (ERS)**

El documento de la especificación de requerimientos de software debe describir todos los requisitos del sistema, estos necesitan ser claros y no dejar ningún tipo de ambigüedad (cada requisito debe tener una única interpretación), además debe ser redactado en lenguaje informal, debido a que el cliente debe ser capaz de entender gran parte del documento y el equipo de desarrollo debe tener la información suficiente para la creación del sistema. Por lo tanto, se hará uso del estándar IEEE 830-1998 *“Recommended Practices for Software Requirements”*, el cual establece el contenido que el documento ERS debería tener como mínimo para considerarse bueno.

La verificación del documento de Especificación de Requerimientos de Software será con base en una revisión de diseño formal, con el objetivo de verificar la claridad de redacción de la Especificación de Requerimientos y su consistencia de Descripción del Producto y con el estándar. Adicionalmente revisa que los requerimientos sean completos y no ambiguos o contradictorios. Para la elaboración del Plan de Revisión se seguirá el estándar IEEE Std 1028-2008 *“Standard for Software Reviews and Audits”*. También se usará el estándar IEEE Std 1012-2016 *“Standard for System, Software and Hardware Verification and Validation”* como guía para el proceso de verificación.

La validación del documento de Especificación de Requerimientos de Software será con base en pruebas funcionales del sistema de tipo Alfa, con el objetivo de 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. Para la elaboración del Plan de Pruebas se seguirá el estándar IEEE 829-1998 *“Standard for Software Test Documentation”*. También se usará el estándar IEEE Std 1012-2016 *“Standard for System, Software and Hardware Verification and Validation”* como guía para el proceso de validación.

#### **4.2 Descripción de Diseño de Software (DDS)**

El documento de Descripción de Diseño de Software (Análisis y Diseño) describe las características del sistema en cuanto a las interfaces externas e internas, bases de datos y en general todo lo relacionado con el diseño y que serán dirigidos por los requerimientos del ERS para que estos sean satisfechos. Incluye distintos apartados que sirven para que quede una idea más clara de cómo se va a organizar todo el sistema y las relaciones que habrá entre cada componente., por lo tanto, para la elaboración de este documento se tomará como referencia el estándar IEEE 1016-2009 “*Standard for Information Technology – Systems Design – Software Design Descriptions*”, que define distintos puntos de vista del diseño según su uso, como el lógico, las dependencias, interfaces, estructuras, interacciones, entre otros.

La verificación del documento de Descripción de Diseño de Software será con base en una revisión por pares del tipo recorrido, con el objetivo de verificar la claridad de la Descripción de Diseño de Software, su factibilidad y la consistencia con la Especificación de Requerimientos y con el estándar. Para la elaboración del Plan de Revisión se seguirá el estándar IEEE Std 1028-2008 “*Standard for Software Reviews and Audits*”. También se usará el estándar IEEE Std 1012-2016 “*Standard for System, Software and Hardware Verification and Validation*” como guía para el proceso de verificación.

La validación del documento Descripción de Diseño de Software será con base en pruebas funcionales del sistema de tipo Caja Negra, con el objetivo de validar que la Descripción de Diseño de Software cumple con las necesidades y expectativas acordadas con el cliente. Para la elaboración del Plan de Pruebas se seguirá el estándar IEEE 829-1998 “*Standard for Software Test Documentation*”. También se usará el estándar IEEE Std 1012-2016 “*Standard for System, Software and Hardware Verification and Validation*” como guía para el proceso de validación.

### **4.3 Plan de Pruebas del Sistema**

El documento del Plan de Pruebas del Sistema define las pruebas necesarias para asegurar que el producto cumple con todos los requerimientos establecidos y que estén correctamente desarrollados. El plan funciona como guía para las actividades de prueba, teniendo objetivos y estrategias de trabajo. Para la

elaboración de este documento se tomará como referencia el estándar ISO/IEC 29119 *“Software Testing Standard”* que define los procesos, el vocabulario, documentación, técnicas y un modelo de evaluación de los procesos para las pruebas para ser usado en cualquier etapa del ciclo de vida de desarrollo del software.

La verificación del documento del Plan de Pruebas del Sistema será con base en una revisión por pares de tipo inspección con el objetivo de verificar la consistencia del Plan de Pruebas de Sistema con la Especificación de Requerimientos del Sistema y con el estándar. Para la elaboración del Plan de Revisión se seguirá el estándar IEEE Std 1028-2008 *“Standard for Software Reviews and Audits.”* También se usará el estándar IEEE Std 1012-2016 *“Standard for System, Software and Hardware Verification and Validation”* como guía para el proceso de verificación.

#### **4.5 Manual de Usuario**

El documento del Manual de Usuario funciona como una guía para los usuarios. No están enfocada solo al usuario final, sino que incluye todos los que estén relacionados al sistema, como vendedores o los encargados del mantenimiento. Estos documentos pueden abarcar temas como la instalación, gestión del software y su mantenimiento. Deben estar completos y no omitir ningún tipo de información que pueda resultar útil para los usuarios, por lo tanto, para la elaboración de este documento se tomará como referencia el estándar IEEE Std 1063-1987 *“Standard for Software User Documentation”* que establece los requerimientos mínimos de la estructura y contenido de la información de la documentación del usuario, ya sea impresa o electrónica.

La verificación del documento del Manual de Usuario será con base en una revisión de diseño forma, con el objetivo de verificar consistencia del Manual de Usuario con el Sistema de Software y con el estándar. Para la elaboración del Plan de Revisión se seguirá el estándar IEEE Std 1028-2008 *“Standard for Software Reviews and Audits.”* También se usará el estándar IEEE Std 1012-2016 *“Standard for System, Software and Hardware Verification and Validation”* como guía para el proceso de verificación.

#### **4.6 Registro de Rastreo**

El documento de Registro de Rastreo contiene las relaciones de los requerimientos, elementos de análisis y diseño componentes y planes de pruebas, esta información facilita la localización y obtención de información de problemas comunes en el software. Para la elaboración de este documento se tomará como referencia el Anexo 4. *Formato para el Registro de Rastreo* que contiene una tabla con distintos campos relacionados a los requisitos funcionales y no funcionales y al documento de diseño.

La verificación del documento del Registro de Rastreo constará de dos revisiones, la primera será con base en una revisión de diseño formal al finalizar la etapa de análisis y diseño. La segunda, se realizará con una revisión de diseño formal para la verificación de la claridad de la documentación del Análisis y diseño, su factibilidad y la consistencia con la Especificación de Requerimientos y con el estándar. Ambas se hacen con el objetivo de verificar que el Registro de Rastreo contenga las relaciones adecuadas entre los elementos de Análisis y Diseño y los componentes. Para la elaboración del Plan de Revisión se seguirá el estándar IEEE Std 1028-2008 “*Standard for Software Reviews and Audits*”. También se usará el estándar IEEE Std 1012-2016 “*Standard for System, Software and Hardware Verification and Validation*” como guía para el proceso de verificación.

#### **4.7 Manual de Mantenimiento de Software (MMS)**

El documento de Manual de Mantenimiento de Software especifica las instrucciones del sistema para que se le pueda proveer de mantenimiento. Como mínimo debe incluir las especificaciones del software, del hardware, los procedimientos correctivos, instalaciones y pruebas de todos los cambios. Para la elaboración de este documento se tomará como referencia el estándar ISO/IEC 14764-2006 “*Software Life Cycle Processes – Maintenance*” que define las actividades y tareas del mantenimiento de software, además proporciona los requerimientos para la planificación del mantenimiento.

La verificación del documento del Manual de Mantenimiento de Software será con base en una revisión de diseño formal, con el objetivo de verificar la consistencia del Manual de Mantenimiento con la Configuración de Software y con el estándar. Para la elaboración del Plan de Revisión se seguirá el estándar IEEE Std 1028-2008 “*Standard for Software Reviews and Audits*”. También se usará el estándar

IEEE Std 1012-2016 “*Standard for System, Software and Hardware Verification and Validation*” como guía para el proceso de verificación.

#### **4.8 Manual de Operación**

El documento de Manual de Operación funciona como un apoyo para la instalación y administración del software, incluyendo información técnica del ambiente en el que opera el software. Para la elaboración de este documento se tomará como referencia el Anexo 5. *Formato para el Manual de Operación* que establece la estructura y contenido mínimo para el manual.

La verificación del documento del Manual de Operación será con base en una revisión de diseño formal, con el objetivo de verificar la consistencia del Manual de Operación con el Software y con anexo 5. Para la elaboración del Plan de Revisión se seguirá el estándar IEEE Std 1028-2008 “*Standard for Software Reviews and Audits*”. También se usará el estándar IEEE Std 1012-2016 “*Standard for System, Software and Hardware Verification and Validation*” como guía para el proceso de verificación.

#### **4.9 Plan de Pruebas de Integración**

El documento de Plan de Pruebas de Integración establece las pruebas que se ejecutarán para verificar que los diferentes componentes resultantes de la fase de análisis y diseño se integran de manera adecuada, es decir, se verifica que no haya problemas en su interacción, por lo cual, para la elaboración de este documento se tomará como referencia al Anexo 6. *Formato para el Plan de Pruebas de Integración*, que define la estructura y formato para documentar cada una de las pruebas.

La verificación del documento del Plan de Pruebas de Integración será con base en una revisión de diseño formal, con el objetivo de verificar consistencia del Plan de Pruebas de Integración con el Análisis y Diseño y con el anexo 6. Para la elaboración del Plan de Revisión se seguirá el estándar IEEE Std 1028-2008 “*Standard for Software Reviews and Audits*”. También se usará el estándar IEEE Std 1012-2016 “*Standard for System, Software and Hardware Verification and Validation*” como guía para el proceso de verificación.

#### **4.10 Plan de Gestión de la Configuración del Software (PGCS)**

El documento del Plan de Gestión de la Configuración del Software define las actividades que se deben realizar para la gestión de la configuración. El plan debe incluir temas como las tareas, métodos (cada método que se implementa para la identificación de los elementos de la configuración de software, para el control e implementación de cambios, las auditorías, los reportes, revisiones y para la elección del personal de mantenimiento), responsables, calendarios, herramientas y recursos. Para la elaboración de este documento se tomará como referencia el estándar IEEE Std 828-1990 “*Standard for Software Configuration Management Plans*” que establece el contenido mínimo del plan, define las tareas específicas a realizar y sus requerimientos para cualquiera de las fases del ciclo de vida de un producto de software.

La verificación del documento del Plan de Gestión de la Configuración del Software será con base en una revisión de diseño formal. Para la elaboración del Plan de Revisión se seguirá el estándar IEEE Std 1028-2008 “*Standard for Software Reviews and Audits*”. También se usará el estándar IEEE Std 1012-2016 “*Standard for System, Software and Hardware Verification and Validation*” como guía para el proceso de verificación.

## **5. 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**

<b>Estándar</b>	<b>Descripción de uso</b>
IEEE Std 830-1998	Elaboración del documento de Especificación de Requerimientos.
IEEE Std 1028-2008	Elaboración de los Planes de Revisión para todos los documentos de esta fase que realicen verificación.

IEEE 829-1998	Elaboración de los Planes de Pruebas para todos los documentos de esta fase que realicen validación.
IEEE Std 828-1990	Elaboración del Plan de la Gestión de la Configuración de Software.
IEEE Std 1063-1987	Elaboración para el Manual de Usuario.
ISO/IEC 29119	Elaboración del Plan de Pruebas del Sistema.
IEEE Std 1012-2016	Guía para las actividades de verificación y validación, según corresponda.

### 5.2.2 Fase de análisis y diseño

<b>Estándar</b>	<b>Descripción de uso</b>
IEEE 1016-2009	Elaboración del documento de Diseño de Software.
IEEE Std 1028-2008	Elaboración de los Planes de Revisión para todos los documentos de esta fase que realicen verificación.
IEEE 829-1998	Elaboración de los Planes de Pruebas para todos los documentos de esta fase que realicen validación.
IEEE Std 1012-2016	Guía para las actividades de verificación y validación, según corresponda.

### 5.2.3 Fase de Construcción

<b>Estándar</b>	<b>Descripción de uso</b>
IEEE Std. 1008	Elaboración de las pruebas unitarias.
<b>Convención</b>	<b>Descripción de uso</b>
Java Code Conventions	Estándar de codificación del lenguaje de Programación Java, utilizado para mejorar la legibilidad del software para un

	entendimiento más rápido y completo, además de aportar limpieza al código.
Clean Code	Utilizado por los patrones y prácticas para tener un código más limpio y para la transformación de un mal código a uno mejor, con mayor legibilidad y para un correcto manejo de los errores dentro del código.

#### 5.2.4 Fase de Integración y Pruebas

Estándar	Descripción de uso
ISO/IEC 14764-2006	Elaboración del Manual de Mantenimiento.
IEEE Std 1028-2008	Elaboración de los Planes de Revisión para todos los documentos de esta fase que realicen verificación.
IEEE 829-1998	Elaboración de los Planes de Pruebas para todos los documentos de esta fase que realicen validación.
IEEE Std 1063-1987	Correcciones finales del Manual de Usuario.
IEEE Std 1012-2016	Guía para las actividades de verificación y validación, según corresponda.

#### 5.3 Documentación

Los estándares implementados se mencionan brevemente en la sección 4. Documentación, además de los que se han especificado para los productos durante las fases del ciclo de vida.

Para tener más detalles de los estándares acudir a la sección 2. Documentos referenciados.

#### 5.4 Métricas

Para la recolección de datos de métricas de sobre la calidad de software se implementará el estándar ISO/IEC 15939-2007.



## **6. Revisiones y auditorías**

### **6.1 Propósito**

En el apartado Revisiones y Auditorías se define la forma de realizar revisiones y/o auditorías para los documentos en los que es requerido estas evaluaciones.

De igual manera se indican las acciones necesarias, así como las adicionales para asegurar que todo está de acuerdo con lo establecido.

### **6.2. Requisitos mínimos**

Para cubrir de forma aceptable las revisiones y auditorías se debe contar como mínimo con las siguientes revisiones:

#### **6.2.1 Revisión de la especificación de software**

Se aplicará una revisión de diseño formal sobre el documento de la especificación de software para verificar la claridad de redacción de la Especificación de Requerimientos y su consistencia de Descripción del Producto y con el estándar de documentación requerido en el Proceso Especifico. Adicionalmente se revisarán que los requerimientos sean completos y no ambiguos o contradictorios. En caso de encontrar defectos serán documentados en un Reporte de Verificación.

El proceso para ejecutar esta revisión se definirá en el Plan de Revisión para RiskMap que será elaborado con base en el estándar IEEE Std 1028-2008 y el estándar IEEE Std 830-1998.

#### **6.2.2 Revisión del plan de pruebas del sistema**

Para el documento de pruebas del sistema se llevará a cabo una revisión por pares del tipo inspección en la que se debe 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.

Para asegurar que todas las pruebas se lleven a cabo, se deberá utilizar un check-list en donde se listen las pruebas a realizar. El proceso para ejecutar esta revisión se definirá en el Plan de Revisión para RiskMap que será elaborado con base en el estándar IEEE Std 1028-2008 y el estándar ISO/IEC 29119.

### **6.2.3 Revisión del manual de usuario**

Se realizará una revisión de diseño formal para verificar la consistencia del Plan de Pruebas de Sistema con la Especificación de Requerimientos y con el estándar de documentación requerido en el Proceso Específico.

Las observaciones encontradas se documentan en un Reporte de Verificación. El proceso para ejecutar esta revisión se definirá en el Plan de Revisión para RiskMap que será elaborado con base en el estándar IEEE Std 1028-2008 y el estándar IEEE Std 1063-1987.

### **6.2.4 Revisión del Registro de rastreo**

Se aplicará una revisión de diseño formal al concluir con la fase de análisis y diseño para verificar que el Registro de Rastreo contenga las relaciones adecuadas entre los requerimientos y los elementos de Análisis y Diseño.

Por otra parte, se hará una revisión del mismo tipo para verificar la 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.

Para ambas revisiones se deberán registrar los defectos u observaciones encontradas en un Reporte de Verificación. El proceso para ejecutar esta revisión se definirá en el Plan de Revisión para RiskMap que será elaborado con base en el estándar IEEE Std 1028-2008.

### **6.2.5 Revisión del diseño**

Se llevará a cabo una revisión por pares de tipo recorrido con el cliente para validar que el Análisis y Diseño cumple con las necesidades y expectativas acordadas en el contrato.

Los defectos o comentarios encontrados se documentan en un Reporte de Validación. El proceso para ejecutar esta revisión se definirá en el Plan de Revisión para RiskMap que será elaborado con base en el estándar IEEE Std 1028-2008.

### **6.2.6 Revisión del plan de pruebas de integración**

Se aplicará una revisión de diseño formal para verificar consistencia del Plan de Pruebas de Integración con el Análisis y Diseño y con el estándar de documentación requerido en el Proceso Específico. El proceso para ejecutar esta revisión se definirá

en el Plan de Revisión para RiskMap que será elaborado con base en el estándar IEEE Std 1028-2008.

#### **6.2.7 Revisión del manual de operación**

El manual de usuario será revisado por medio de una revisión del diseño formal en donde se verificará la consistencia del Manual de Operación con el Software y con el estándar de documentación requerido en el Proceso Específico. El proceso para ejecutar esta revisión se definirá en el Plan de Revisión para RiskMap que será elaborado con base en el estándar IEEE Std 1028-2008.

#### **6.2.8 Revisión del manual de mantenimiento**

Se hará una revisión de diseño formal sobre el manual de mantenimiento con el objetivo de Verificar consistencia del Manual de Mantenimiento con la Configuración de Software y con el estándar de documentación requerida en el Proceso Específico. El proceso para ejecutar esta revisión se definirá en el Plan de Revisión para RiskMap que será elaborado con base en el estándar IEEE Std 1028-2008.

#### **6.2.9 Revisión del plan de gestión de la configuración del software**

Se llevará a cabo una revisión formal de diseño para verificar que los procedimientos de gestión de la configuración de software sean adecuados y proporcionen el control necesario sobre la documentación y el código. El proceso para ejecutar esta revisión se definirá en el Plan de Revisión de Configuración del Software con base en el estandar IEEE Std 1028-2008 y el estándar IEEE Std 828-1990.

### **7. Pruebas**

El propósito de las pruebas es garantizar que los productos de software generados sean de calidad, evidenciando con las pruebas los defectos del software y validando que el producto esté acorde a lo que se especificó.

#### **7.1 Proceso de realización de las pruebas:**

##### **1. Configurar pruebas**

Como primer paso será necesario planear y preparar todo lo necesario para las pruebas:

- Plan de pruebas
- Casos de prueba

- Procedimientos para realizar las pruebas
  - Herramientas para las pruebas
  - Ambiente de pruebas
2. Realizar pruebas

Se llevan a cabo las pruebas que fueron planeadas y se obtienen los resultados de las pruebas.
  3. Evaluar resultados

Se verifica si los resultados de los casos de prueba coinciden con los resultados esperados.
  4. Reportar defectos

Se documentan los defectos que fueron encontrados en un Reporte de defectos.
  5. Realizar correcciones

Con base en el Reporte de defectos, se realizan las correcciones pertinentes y se realizan nuevamente las pruebas para asegurar que no se haya introducido otro defecto como efecto secundario.

Las actividades de aseguramiento de calidad en este proceso incluyen la revisión de los siguientes artefactos:

- Plan de pruebas: en este se indican los componentes que serán probados, las características consideradas en la prueba, las tareas a realizar y los riesgos asociados a la prueba.
- Casos de prueba: relacionados directamente con los casos de uso, deben indicar condiciones de ejecución, datos de entrada y datos de salida.
- Procedimiento de pruebas: puede referirse a un caso de prueba en particular o bien aplicarse a varios, define como ejecutar la prueba y en qué condiciones.
- Reporte de incidentes: informa los resultados de la realización de pruebas y debe contener:
  - Entradas
  - Resultados esperados
  - Resultados presentados

- Anomalías
- Fecha y hora de ejecución
- Procedimiento aplicado
- Condiciones de ejecución
- Número de veces que se realizó
- Observaciones

Las revisiones realizadas y las observaciones resultantes deberán de ser documentadas y comunicadas al LPR para su correspondiente atención.

El RUSQ deberá asegurarse de que las observaciones realizadas sean atendidas.

## **7.2 Pruebas de unidad**

Los componentes deberán ser probados por separados para verificar que estos se comporten según fueron diseñados. Para la implementación de las pruebas unitarias se deberá seguir el estándar *IEEE 1008 – Standard for Software Unit Testing*.

El conjunto de pruebas unitarias se debe de hacer con base en el Plan de Pruebas Unitarias elaborado siguiendo el estándar *IEEE 829-1998 – Standard for Software Test Documentation*, por lo tanto, para ver el procedimiento completo que se seguirá para la aplicación de estas pruebas se deberá acudir a este plan.

## **7.3 Pruebas de integración**

Una vez probados los componentes de manera individual, probamos la interacción entre ellos para garantizar que funcionan de acuerdo con lo especificado.

Se deberá de elaborar un Plan de Pruebas de Integración con base en el estándar *IEEE 829-1998 – Standard for Software Test Documentation* durante la fase de Análisis y Diseño. Toda la información necesaria para llevar a cabo estas pruebas se encuentra disponible en este plan que servirá como guía para su posterior implementación contemplada en la fase de Integración y Pruebas, documentando los resultados en un Reporte de Pruebas de Integración.

El tipo de pruebas de integración a realizarse es la incremental, por lo tanto, conforme los componentes son sometidos a pruebas unitarias deberán de ser probados con el conjunto de componentes a los que ya se la aplicaron las pruebas de integración.

## **7.4 Pruebas de sistema**

Se tiene como objetivo probar los flujos de trabajo que involucran al sistema completo. Son pruebas de integración del sistema de información completo, y permiten probar el sistema en su conjunto y con otros sistemas con los que se relaciona para verificar que las especificaciones funcionales y técnicas se cumplen.

Es por ello por lo que, durante la fase de requerimientos, se deberá generar un plan de pruebas cuyo contenido debe de incluir una visión general de las metodologías, cronogramas y recursos para probar el software.

La definición de los procedimientos que se utilizarán para llevar a cabo estas pruebas estará definida en el Plan de pruebas de Sistema con base en el estándar *ISO / IEC 29119 – Software Testing Standard* que describe la información básica necesaria y resultados de las pruebas de software.

## **8. Reporte de problemas y acciones correctivas**

Sección dedicada a la identificación de técnicas (y las responsabilidades de la organización para su ejecución) desarrolladas para la creación de los reportes, el seguimiento y la resolución de problemas.

Es importante mencionar que el alcance de esta actividad no es solamente para la fase de desarrollo de software, sino que también se debe contemplar el proceso de mantenimiento.

### **8.1 Responsabilidades**

Para este plan, el equipo encargado de que se cumplan las acciones correctivas (siguiendo las normas y requerimientos del proyecto), además de darle un adecuado seguimiento hasta que se llegue a una resolución del problema, será el equipo de aseguramiento de calidad, los cuales deben contar con las siguientes características:

- Conocimiento del proceso o producto
- Autoridad para resolver el problema e implementar las acciones correctivas
- Habilidades en las disciplinas técnicas
- Un líder del equipo designado

Cualquier problema que sea encontrado durante el proyecto durante las revisiones de la documentación, en la fase de desarrollo, instalaciones, pruebas y/o mantenimiento debe ser reportado inmediatamente a este equipo.

## 8.2 Contenido

Para los reportes de problemas o las acciones correctivas, se recomienda que como mínimo contengan:

- **Información general:** información básica acerca del reporte o la acción correctiva.
- **Objetos afectados:** objetos que son afectados por el problema o por la acción correctiva.
- **Tareas:** checklist de las tareas relacionadas al reporte o la acción correctiva.
- **Anexos:** archivos, enlaces o mayor información sobre el problema o la acción correctiva.
- **Análisis del impacto:** vista de cómo el problema o la acción correctiva impacta en otros objetos.
- **Relaciones:** asociaciones entre el reporte del problema o la acción correctiva con cualquier otro objeto de negocio.
- **Seguridad:** responsables quienes tendrán acceso.
- **Historial:** registro de la secuencia de acciones que se han realizado.

## 9. Herramientas, técnicas y metodologías

Para dar soporte a las actividades realizadas durante el desarrollo e implementación del plan de aseguramiento de calidad se aplicaron técnicas y metodologías que involucran diferentes herramientas de soporte para el control del versionado, la ejecución de las pruebas de software y herramientas para la administración e implementación del proyecto que se detallan a continuación.

### 9.1 Herramientas

Para la elección de las herramientas de soporte se consideraron diferentes opciones, analizando las ventajas y desventajas, y seleccionando aquellas que se adecuen más al proyecto. En el caso de los programas a ser usados, se priorizó el uso de software libre dado el presupuesto y las restricciones para el proyecto.

#### Herramientas para el Control de versiones

##### Git

Para el control de versiones se decidió hacer uso de un sistema de control de versiones distribuido (SVN) como lo es Git. Este incluye un repositorio central y una serie de repositorios locales en donde cada usuario puede tener una copia completa del proyecto, haciendo que el acceso a la historia de cada uno sea extremadamente rápido. De esta manera, se puede trabajar de forma remota fácilmente sin conexión a internet. La velocidad es otro de los puntos fuertes de Git frente a otros sistemas de control de versiones, ya que necesita menos capacidad de procesamiento y gestión al poder realizar las operaciones en local.

Cabe mencionar que Git es compatible con casi todos los entornos de desarrollo y las herramientas de líneas de comandos de Git se ejecutan en todos los sistemas operativos principales.

Esta herramienta será fundamental para el trabajo en equipo, ya que permite tener un desarrollo en paralelo para un proyecto con acceso compartido sin estar físicamente cerca, así como identificar que usuario y cuando ha realizado cada modificación. Además, podremos comparar las diferentes versiones y restaurarlas en caso de que algo salga mal.

## **GitHub**

GitHub es un servicio de alojamiento de repositorios de software que usa Git y que ha sido pensado para compartir código de una manera fácil y ágil. Esta herramienta nos permite alojar proyectos de código abierto de forma gratuita ofreciéndonos un seguimiento de errores, búsqueda rápida y una amplia comunidad de desarrolladores alrededor del mundo.

Esta plataforma cuenta también con funciones de organización y gestión de proyectos. Se pueden asignar tareas a individuos o grupos, establecer permisos y roles para los colaboradores y usar la moderación de comentarios para mantener a todos en la tarea.

## **Herramientas para las Pruebas de Software**

### **JUnit**

¿Cuándo utilizar JUnit? Se hará uso de Junit cuando se tenga que realizar pruebas con una única clase.



Permite evaluar el resultado de la ejecución de un método. Con esta herramienta es posible comparar el resultado esperado, con el que realmente se obtuvo después de ejecutar el método.

### **Mockito**

¿Cuándo utilizar Mockito? Cuando las pruebas a una clase utilicen otras clases.

Con Mockito se tiene la posibilidad de simular la respuesta de otro método necesario para ejecutar el método que necesitamos probar. Esto permite un enfoque único en el método que deseamos probar.

### **Bugzilla**

Herramienta que permitirá que los desarrolladores y evaluadores puedan dar seguimiento de los defectos pendientes. Bugzilla permite organizar en múltiples formas los defectos de software, permitiendo el seguimiento de múltiples productos con diferentes versiones, a su vez compuestos de múltiples componentes. Permite además categorizar los defectos de software de acuerdo con su prioridad y gravedad, así como asignarles versiones para su solución.

También permiten anexar comentarios, propuestas de solución, designar a responsables a los que asignar la resolución y el tipo de solución que se aplicó al defecto, todo ello llevando un seguimiento de fechas en las cuáles sucede cada evento y, si se configura adecuadamente, enviando mensajes de correo a los interesados en el error.

### **Herramientas de soporte para las revisiones**

Las revisiones podrán ser guiadas a través de listas de verificación que deben contener aspectos o características que deben estar presentes en los diferentes artefactos generados durante la realización de un proyecto y deberán de especificar lo siguiente:

- Descripción breve
- Proyecto e identificación del documento
- Fecha de revisión
- Encargado de aplicar la revisión
- Característica verificada y la calificación correspondiente

Las calificaciones que se pueden asignar son las siguientes:

Completa y correcta (CC)	La característica que se revisa está presente y fue elaborada de forma adecuada
Completa, pero con errores (CE)	La característica que se revisa está presente, pero se necesita corregir algunos aspectos
Incompleta pero correcta (IC)	La característica que se revisa no está completa pero la parte elaborada es correcta
Incompleta y con errores	La característica que se revisa no está completa y la parte elaborada es incorrecta
No considerada	La característica que se revisa no está presente en el artefacto revisado

- Observaciones

## Herramientas para la Administración del proyecto

### Trello

Utilizamos Trello como software de administración de proyectos ya que esta nos permite organizar, coordinar y gestionar las tareas en tiempo real y con velocidad. Mediante este software asignaremos las actividades a cada uno de los miembros del equipo de trabajo basado en la metodología Kanban.

Consiste en un sistema de tableros, listas y tarjetas que corresponden a las actividades que deben realizarse y en las que es posible añadir imágenes, plazos de entrega, lista de tareas, etiquetas separadas por color y comentarios asociados a cada tarea. Las tarjetas pueden ser editadas y reordenadas, arrastrándolas y soltándolas entre las listas para comunicar el avance de cada una de ellas. De esta manera, una persona que ingrese a Trello estará al tanto del desarrollo de las tareas de un proyecto.

## Herramientas para la Implementación de la aplicación

### Android Studio

Dado que el software que se desea asegurar con este plan de SQA corresponde a una aplicación móvil para dispositivos Android, usaremos el IDE oficial para desarrollar aplicaciones para este sistema operativo, Android Studio. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, como las siguientes:

1. Sistema de compilación flexible basado en Gradle
2. Emulador rápido y cargado de funciones
3. Entorno unificado donde puedes desarrollar para todos los dispositivos Android
4. Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también incorporar códigos de muestra
5. Variedad de marcos de trabajo y herramientas de prueba
6. Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros

## 9.2 Técnicas

Las técnicas que se implementarán con el fin de evaluar y mejorar la calidad de software son:

- Pruebas. Para mayor detalle de los momentos en que estas distintas técnicas se aplican y para qué tipo de pruebas se implementan, además de los estándares implementados ver la sección 7. *Pruebas*. En resumen, se hará uso de cada una de estas:
  - Alfa: pruebas ejecutadas en las oficinas del desarrollador del producto por un grupo de personas que representan al cliente final, con el fin de registrar errores y problemas de uso.
  - Rendimiento: técnica para medir la velocidad de procesamiento y el tiempo de respuesta del sistema.
  - Estrés: técnica en la cual se seleccionan actividades a probar en un sitio para ejecutarlas en un tiempo determinado desde una ubicación remota.
  - Regresión: pruebas que se realizan cuando un software ha sufrido un cambio, ya sea que se agregó algún módulo nuevo o para solucionar algún problema.

- Caja negra: técnica que utiliza el análisis de la especificación, funcional y no funcional, sin tener en cuenta la estructura interna del programa.
- Caja blanca: técnica de en la que se prueba la estructura interna, el diseño y la codificación del software.
- Revisiones: técnica para la detección de errores en el producto de software, puede aplicarse desde etapas tempranas del desarrollo.
- Auditorías: actividad de revisión para verificar el cumplimiento de un sistema establecido y su efectividad o área y mejora o de acción correctiva.
- Uso de métricas según estándar ISO/IEC 15939-2007

### 9.3 Metodologías

Las técnicas en conjunto con las herramientas mencionadas anteriormente conforman las metodologías a aplicarse para llevar a cabo las diferentes actividades que dan soporte al aseguramiento de la calidad del software.

Con el objetivo de asegurar la calidad que los productos de software generados y controlados con Git y GitHub y la satisfacción de los requerimientos establecidos, se deberá implementar revisiones de los documentos mencionados en el apartado 6. *Revisiones* del presente plan por cada versión controlada, así como las auditorías correspondientes para detectar los errores y así puedan presentarse las propuestas de mejora o acciones correctivas que eviten la propagación en futuras versiones.

Además, será muy importante hacer uso de las métricas y pruebas de software, las cuáles deben de estar conforme a lo establecido en el apartado 7. *Pruebas* en las que se menciona la aplicación de pruebas a los módulos de manera individual al momento de ser generados o modificados y la integración incremental de estos con apoyo de las herramientas JUnit y Mockito para la ejecución y Bugzilla para el seguimiento de los defectos encontrados.

Es importante que tanto para las revisiones de los documentos como las revisiones derivadas de las pruebas se considere el uso de las listas de verificación detalladas en las *Herramientas de soporte para las revisiones* y recopilación de evidencia que soporten las actividades realizadas.

## 10. Control de código

Para llevar un control del código generado en la fase de construcción y mantenimiento de software, los cambios surgidos, evaluados y aprobados se llevarán a cabo un proceso de control de la configuración que incluye las actividades que se detallan a continuación.

### 10.1 Solicitudes de cambio

Para cada cambio se deberá realizar una solicitud de cambio siguiendo la plantilla disponible en el *Anexo 1. Formato para Solicitud de cambio*, la cual será posteriormente examinada y aprobada o rechazada por el ACS según los siguientes factores:

- Contribución esperada de la aplicación del cambio propuesto
- Urgencia del cambio
- Efecto del cambio en los cronogramas del proyecto, nivel de servicio, etc.
- Esfuerzos necesarios para hacer operativo el cambio
- Esfuerzos necesarios para asegurar la calidad del software
- Estimación de recursos profesionales y costos para realizar el cambio

Para llevar a cabo una nueva versión el LPR tendrá que estar de acuerdo que todas las unidades del equipo hayan acabado con sus tareas para que posteriormente registre la fecha, el número de versión y describa las adiciones hechas en esta nueva versión, con el fin de proveer copias al cliente en caso de ser necesarios, y también para controlar las versiones del software.

Cada cambio en el código fuente o en un módulo/unidad de software deberá ser documentado mediante los comentarios de las nuevas adiciones, de igual manera deberá ser comunicado a los demás integrantes del equipo para estar atentos a estos cambios.

### 10.2 Aseguramiento de la calidad de los cambios

Se deberá asegurar la calidad de cada nueva versión de configuración del software antes de que entre en funcionamiento. Para ello, se deberán contemplar el aseguramiento de la calidad en los siguientes dos niveles:

- Aseguramiento de la calidad en cada Objeto de Configuración cambiado

El IPR deberá elaborar un Plan de Revisiones con base en el estándar *IEEE-1028-2008* y Pruebas (sección 7 del presente plan) apropiadas al tipo de cambio introducido.

- Aseguramiento de la calidad de la versión nueva de software

Se considera que se ha completado una nueva versión de software una vez que los SCI modificados reemplazan a los SCI anteriores. A menos que se pruebe toda la nueva versión, o al menos todas las partes del software que podrían verse afectadas, la posibilidad de que el software funcione correctamente es escasa.

### 10.3 Proceso de liberación

Se deberá elaborar un Plan de la Configuración de Software con base en el estándar *IEEE-828-2012* cuya finalidad es la de agendar cuando se liberarán las versiones, qué recursos serán necesarios y cuál es el progreso de las actividades de liberación actual.

Para la documentación de las versiones de configuración de software se deben de cumplir dos tareas importantes:

- Documentación de las versiones de los Objetos de Configuración

Cada vez que una Solicitud de cambio sea aprobada, se deberá generar un documento que identifique la versión siguiendo la plantilla disponible en el *Anexo 2. Formato de la Documentación de las versiones de los objetos de configuración*.

- Documentación de las liberaciones de la Configuración de Software

De igual manera, se deberá documentar el historial de las versiones de cada liberación con ayuda de la plantilla disponible en el *Anexo 3. Formato de la Documentación de las liberaciones de la configuración de software*.

### 10.4 Localización del código

El código fuente, así como los módulos serán desarrollados bajo el uso del controlador de versiones Git y un repositorio de GitHub. Este servirá para el control general de código y el manejo de las versiones del código. El etiquetado a utilizarse llevará los siguientes datos: **nombreDelModulo\_equipoAutorizado\_apellidoDelEncargado**,

representado en el trabajo con ramas. Este etiquetado nos da la noción del pedazo del código, equipo del encargado y el encargado para que durante el tiempo del ciclo de ese código la seguridad sea auditada con el fin de asegurarse que el control se lleve a cabo.

El desarrollo del software indica la creación de código, por lo que, la localización de este será descentralizada, es decir, cada encargado tendrá disponible el código en un repositorio en la nube.

## **11. Control de medios**

En este punto se abordará todo medio en el que se estarán guardando el proyecto de software.

El control de estos medios se deberá hacer a partir de los puntos que se encuentran posteriormente definidos como “*Acceso no autorizado*” y “*daño o degradación desapercibida*”.

Para el caso del sistema RiskMap se hará la documentación del control de medios para llevar a cabo este control sobre el software, así como también, el resguardo de las copias. Los procedimientos por implementar serán descritos a continuación.

El equipo de seguridad será encargado de llevar a cabo el *plan de seguridad del software*. El procedimiento de control empieza por la parte de etiquetado en donde se presentarán los mismos datos que en el control de código, estos son:

- Nombre del medio
- Personal autorizado
- Encargados del control

Lo anterior descrito servirá como el preámbulo del control de medios que, a su vez, nos ayuda para documentación posterior de estos.

### **11.1 Acceso no autorizado**

Se deberá hacer uso de procedimientos para resguardar los medios de accesos no autorizados.

La organización deberá utilizar una contraseña única para cada tipo de usuario dependiendo del rango de acceso ya sea parcial o total. Estas contraseñas deberán

ser generadas y entregadas al integrante según el rango que el encargado considere apropiado.

Antes de poder generar estas contraseñas, los encargados de SQA deberán clasificar los medios, asegurar su almacenamiento y definir las restricciones de la siguiente manera:

**Software.** El software deberá estar almacenado y disponible para su recuperación. El sitio en donde se almacena el software deberá ser un lugar seguro con las características que se mencionan en el punto 11.2. Para el acceso y recuperación de este solo deberá ser permitido para el equipo de desarrollo o quién lo solicite además de controlar el rango de acceso para este.

**Copia.** La copia del software deberá estar almacenado y disponible para su recuperación. El sitio en donde se almacena la copia del software deberá ser un lugar seguro con las características que se mencionan en el punto 11.2. Para el acceso y recuperación de este solo deberá ser permitido para el equipo de desarrollo o quién lo solicite además de controlar el rango de acceso para este.

### **11.2 Daño o degradación desapercibida**

Para evitar los daños o la degradación de los medios de almacenamiento se hará uso de técnicas adecuadas de la administración de la configuración.

De igual manera, se deberá considerar lugares seguros para su almacenamiento, estos deben contar con las siguientes características:

1. Lugar a prueba de fuego
2. Con temperaturas internas de hasta 25° Centígrados

El personal de SQA deberá implementar actividades como la revisión periódica de los medios con el fin de documentar el estado de estos.

## **12. Recolección de registros, mantenimiento y retención**

Los siguientes procedimientos están conforme al plan de administración de la configuración del software aprobado por la organización.

Para garantizar la retención e integridad de estos archivos sus accesos serán limitados a el líder del proyecto o responsable del proyecto específico, el cliente para



su verificación y validación, por último, los encargados de llevar el control de documentos y la configuración del software.

### **12.1 Recolección de registros**

Los documentos tendrán que cumplir con los siguientes puntos para realizar una recolección para el resguardo de estos:

- Los documentos deberán haber estado validados y verificados con que cumplan los estándares y requerimientos impuestos por el cliente.
- Los documentos resultantes al finalizar el proyecto del software, esto implica la terminación del mantenimiento. Esto con fines de mantener registros históricos.

A continuación, se presenta una lista de los documentos que tendrán que recolectar para su resguardo:

- 4 Manuales
- 5 Diseño específico
- 6 Plan de pruebas
- 7 Plan de Aseguramiento de la calidad.
- 8 ERS
- 9 Documento de configuración del software

### **12.2 Mantenimiento de registros.**

Los documentos finales o registros estarán almacenados físicamente en un almacén seguro, a prueba de fuego y con seguridad activa en el lugar. Por otro lado, se tendrán los archivos originales de estos documentos almacenados en un repositorio oficial de la organización en la plataforma en la nube GitHub.

Para controlar el acceso sobre estos registros del software, el acceso solo se le dará al líder del desarrollo, el cual será encargado de recuperar el o las partes que el interesado necesita y devolver este registro en caso de finalización de uso.

### **12.3 Retención de registros.**

Los documentos estarán retenidos o almacenados para la recuperación de datos históricos sobre proyectos realizados con el tiempo. Estos documentos son los mismos que se describen para su recolección en el punto 12.1.

En el caso de ser solicitada una destrucción por parte del cliente o la llegada del tiempo límite propuesto como plazo de retención, se llevará a cabo su destrucción del documento y se suspenderá su retención.

### 13. Entrenamiento

En esta sección se tiene como objetivo el identificar los conocimientos y habilidades necesarias para poder llevar a cabo las actividades de aseguramiento de la calidad de manera efectiva.

Para esto, se deberá definir un Plan de Entrenamiento aplicado a las personas que formarán parte de la unidad de aseguramiento de la calidad. En este se identificarán las actividades de formación requeridas para lograr una implementación exitosa del Plan de SQA.

A continuación, se presentan habilidades requeridas para realizar las tareas orientadas a las revisiones:

Tarea	Habilidades o conocimientos requeridos
Revisión de código	<ul style="list-style-type: none"><li>• Lenguaje de programación aplicado</li><li>• Framework utilizado en la etapa de implementación</li></ul>
Revisión de la documentación	<ul style="list-style-type: none"><li>• Modelo de calidad de referencia</li><li>• Metodología de desarrollo de software</li><li>• Guías institucionales de elaboración</li></ul>
Auditoría del proceso de desarrollo	<ul style="list-style-type: none"><li>• Proceso de desarrollo de software aplicado</li><li>• Estándares que fueron tomados como referencia</li></ul>
Auditoría del proceso de Pruebas de Software	<ul style="list-style-type: none"><li>• Niveles y técnicas de prueba</li><li>• Estrategia de pruebas</li><li>• Herramientas de soporte</li></ul>
Auditoría al proceso de control de cambios	<ul style="list-style-type: none"><li>• Administración de la configuración (registro, seguimiento y atención)</li></ul>
Auditoría al proceso de control de riesgos	<ul style="list-style-type: none"><li>• Administración de riesgos (identificación, análisis, atención)</li></ul>

En el caso del Plan de Entrenamiento, es fundamental que sea incluya la siguiente información:

- Descripción del personal a ser capacitado
- Objetivos de la capacitación
- Contenido a cubrir en el entrenamiento
- Cantidad estimada de todos los recursos necesarios para llevar a cabo el entrenamiento
- Procedimientos para evaluar la efectividad del entrenamiento y para hacer modificaciones al entrenamiento
- Calendarización de las sesiones de entrenamiento

El personal debe de ser capacitado de acuerdo al objetivo que debe alcanzar, de tal forma que, si su asignación es la generación del Documento de Especificación de Requisitos, conocerá el estándar o estándares de referencia y los lineamientos que debe seguir para generarlo.

## **14. Anexos**

### **Anexo 1. Formato para Solicitud de cambio**

#### ***(1) Change principles***

- The initiator
- The date the SCR was presented
- The character of the change
- The goals
- The expected contribution to the project / system
- The urgency of performance

#### ***(2) Change details***

- Description of the proposed change
- A list of the SCIs to be changed
- Expected effect on other SCIs
- Expected effect on interfaces with other software systems and hardware firmware

- Expected delays in development completion schedules and expected disturbances to services to customers

***(3) Change timetable and resources estimates***

- Timetable for implementation
- Estimated required professional resources
- Other resources required
- Estimated total cost of the requested change

**Anexo 2. Formato de la Documentación de las versiones de los objetos de configuración**

***Identification***

- SCI Version number
- Names(s) of software engineer(s) who implemented the change
- Date the new version was completed and approved

***Changes in the new version***

- Former SCI version number
- Short description of the introduced changes
- List of other SCIs that had to be changed as a result of the current changes
- List of SCOs included in the new version
- List of software problem reports resolved by the new version
- Operational as well as other implications of the changes introduced in the new version

**Anexo 3. Formato de la Documentación de las liberaciones de la configuración de software**

***Identification and installations***

- Release version and revision number
- Date of the new version's release
- List of installations where the release was entered (site, date, name of technician who installed the version), if applicable

***Configuration of the released version***

- List of SCIs in the released version, including identification of each SCI version

- List of hardware configuration items required for operating the specified version, including specification of each hardware configuration item
- List of interfacing software systems (including version) and hardware systems (including model)
- Installation instructions for the new release

#### ***Changes in the new version***

- Previous software configuration version
- List of SCIs that have been changed, new SCIs introduced for the first time, and deleted SCIs
- Short description of introduced changes
- Operational and other implications of changes introduced in the new release

#### ***Further development issues***

- List of software system problems that have not been solved in the new version
- List of SCRs and proposals for development of the software system for which implementation of development was delayed

### **Anexo 4. Formato para el Registro de Rastreo**

Formato para el Registro de Rastreo.xlsx –  
<https://github.com/PabloR9080/Documentacion-PlanSQA/blob/main/Documentos-Estandares/Formato%20para%20el%20Registro%20de%20Rastreo.xlsx>

### **Anexo 5. Formato para el Manual de Operación**

Formato\_ManualOperaciones.docx –  
[https://github.com/PabloR9080/Documentacion-PlanSQA/blob/main/Documentos-Estandares/Formato\\_ManualdeOperaciones.docx](https://github.com/PabloR9080/Documentacion-PlanSQA/blob/main/Documentos-Estandares/Formato_ManualdeOperaciones.docx)

### **Anexo 6. Formato para el Plan de Pruebas de Integración**

Formato\_PlanPruebasIntegración.xlsx –  
[https://github.com/PabloR9080/Documentacion-PlanSQA/blob/main/Documentos-Estandares/Formato\\_PlanPruebasIntegracion.xlsx](https://github.com/PabloR9080/Documentacion-PlanSQA/blob/main/Documentos-Estandares/Formato_PlanPruebasIntegracion.xlsx)