

**APLICACIÓN MÓVIL PARA EL SEGUIMIENTO Y CONTROL DE  
PACIENTES DE LA CLÍNICA “CLINUD” EN SANTIAGO DE CHILE**

**Plan de Configuración**

**Versión 1.0**

**Historia de revisiones**

Fecha	Versión	Descripción	Autor
03/06/2024	1.0	Versión inicial del documento	Juan Matinez

## Contenido

<b>1. INTRODUCCIÓN .....</b>	<b>3</b>
1.1. PROPÓSITO .....	3
1.2. ALCANCE .....	3
1.3. TERMINOLOGÍA .....	3
1.4. REFERENCIAS .....	4
<b>2. GESTIÓN DE SCM.....</b>	<b>4</b>
2.1. ORGANIZACIÓN .....	4
2.2. RESPONSABILIDADES .....	6
2.3. POLÍTICAS, DIRECTIVAS Y PROCEDIMIENTOS APLICABLES .....	8
<b>3. ACTIVIDADES DE SCM .....</b>	<b>12</b>
3.1. IDENTIFICACIÓN DE LA CONFIGURACIÓN .....	12
3.1.1. <i>Elementos de configuración</i> .....	12
3.1.2. <i>Nomenclatura de Elementos</i> .....	13
3.1.3. <i>Elementos de la Línea Base del Proyecto</i> .....	16
3.1.4. <i>Recuperación de los Elementos de configuración</i> .....	19
3.2. CONTROL DE CONFIGURACIÓN .....	23
3.2.1. <i>Solicitud de cambios</i> .....	23
3.2.2. <i>Evaluación de cambios o Análisis de Impacto</i> .....	23
3.2.3. <i>Aprobación o desaprobación de cambios</i> .....	24
3.2.4. <i>Implementación de cambios</i> .....	24
3.3. ESTADO DE LA CONFIGURACIÓN .....	24
3.4. AUDITORIAS Y REVISIONES DE CONFIGURACIÓN.....	25
3.5. CONTROL DE INTERFASES .....	25
3.6. CONTROL DE SUBCONTRATOS Y VENDEDORES .....	25
<b>4. CALENDARIO .....</b>	<b>25</b>
<b>5. RECURSOS.....</b>	<b>27</b>
<b>6. MANTENIMIENTO DEL PLAN DE SCM.....</b>	<b>29</b>

## **1. Introducción**

Este documento describe al plan de configuración para el desarrollo de la aplicación móvil destinada al seguimiento y control de pacientes de la clínica “CLINUD” en Santiago de Chile

### **1.1. Propósito**

Este documento describe las actividades de gestión de configuración de software que deben ser llevadas a cabo durante el proceso de desarrollo del proyecto. Aquí se definen tanto los productos que se pondrán bajo control de configuración como los procedimientos que deben ser seguidos por los integrantes del equipo de trabajo.

### **1.2. Alcance**

El Plan de configuración está basado en algunos supuestos que se detallarán:

- El tiempo de duración del proyecto está limitado a 13 semanas, por lo tanto se busca una rápida respuesta a los cambios, tratando que este procedimiento sea lo menos burocrático posible.
- El Modelo de Proceso se basa en un desarrollo incremental, dado por las distintas iteraciones. Resulta importante tener control sobre cada una de las iteraciones y fases, de los productos generados en estas y de los cambios surgidos, evaluados y aprobados.
- Se deben incluir en control de configuración la mayor cantidad de productos posibles, tomando en cuenta siempre las restricciones dadas por la duración del proyecto y por la capacidad organizativa del grupo.
- La elección de los elementos de configuración se realizará en base a los entregables, siendo ésta responsabilidad del Responsable de SCM, apoyado por los integrantes de cada disciplina.

### **1.3. Terminología**

- CCB (Configuration Control Board) Comité de Control de Configuración.
- CI (Configuration Item) elemento bajo gestión de Configuración.
- SCA (Software Change Authorization) Autorización de Cambio en el Software.
- SCM (Software Configuration Management) Gestión de Configuración del Software.
- SCMR (SCM Responsable) Responsable de SCM.

- SCR (System/Software Change Request) Petición de Cambio en el Sistema/Software.
- SQA (Software Quality Assurance) Aseguramiento de la Calidad del Software.
- SQAR (SQA Responsable) Responsable de SQA.

#### 1.4. Referencias

[1] ANSI/IEEE Std 828-1990, IEEE Standard for Software Configuration Management Plans.

[2] 2002, Modelo de Proceso.

## 2. Gestión de SCM

Juan Martínez, responsable de gestión de configuración SCM

Dereck Jaramillo, desarrollador principal

Fabián Yépez, responsable de pruebas y calidad

### 2.1. Organización

#### 1. Planificación de la Gestión de la Configuración:

- **Roles y Responsabilidades:** Designar claramente los roles y responsabilidades del equipo de SCM y los desarrolladores en el proyecto.
- **Documentación del Plan de SCM:** Crear un plan detallado que describa los procesos, herramientas y recursos necesarios para la gestión de la configuración.
- **Definición de la Política de SCM:** Establecer políticas y procedimientos para la gestión de la configuración, alineándolos con los objetivos del proyecto y las mejores prácticas de la industria.

#### 2. Gestión de los Elementos de Configuración:

- **Identificación de los Elementos de Configuración:** Definir qué componentes del software (código fuente, documentación, scripts, etc.) serán gestionados y controlados.
- **Asignación de Identificadores Únicos:** Asignar identificadores únicos a cada elemento de configuración para facilitar su seguimiento y control.

#### 3. Control de Versiones:

- **Implementación de un Sistema de Control de Versiones (VCS):** Utilizar herramientas de control de versiones (como Git, Subversion, Mercurial) para gestionar el historial de cambios en el código y otros elementos de configuración.
- **Definición de Estrategias de Ramas:** Establecer políticas para el uso de ramas en el repositorio de código, como ramas para desarrollo, pruebas y producción.
- **Fusión y Resolución de Conflictos:** Implementar procedimientos para la fusión de ramas y la resolución de conflictos de código.

#### 4. Control de Cambios:

- **Proceso de Solicitud de Cambios:** Establecer un proceso formal para solicitar, revisar, aprobar y registrar cambios en los elementos de configuración.
- **Evaluación de Impacto:** Evaluar el impacto de los cambios propuestos en el sistema y en los otros elementos de configuración.
- **Registro de Cambios:** Mantener un registro detallado de todos los cambios aprobados y su implementación.

#### **5. Construcción y Automatización:**

- **Definición de Procesos de Construcción:** Establecer procedimientos para la construcción del software, asegurando que los builds sean consistentes y reproducibles.
- **Automatización de Builds y Pruebas:** Implementar herramientas de automatización (como Jenkins, Bamboo, Travis CI) para la construcción, prueba y despliegue automáticos del software.

#### **6. Auditoría y Control de Configuración:**

- **Revisión de Configuración:** Realizar revisiones periódicas para asegurar que todos los elementos de configuración están en conformidad con las especificaciones y estándares del proyecto.
- **Auditoría de SCM:** Llevar a cabo auditorías para verificar que los procesos de SCM se están siguiendo adecuadamente y que el estado del sistema es conforme con la documentación y registros.

#### **7. Gestión de Librerías de Software:**

- **Almacenamiento Seguro de Versiones:** Almacenar versiones controladas y aprobadas de software y documentación en una librería de software (repositorio).
- **Acceso y Recuperación:** Asegurar que el acceso a los elementos de la librería está controlado y que se puede recuperar cualquier versión anterior si es necesario.

#### **8. Gestión de la Liberación y Despliegue:**

- **Planificación de Liberaciones:** Definir la estrategia para las liberaciones de software, incluyendo la frecuencia y el contenido de cada liberación.
- **Gestión del Despliegue:** Coordinar el despliegue de nuevas versiones del software, asegurando que se siguen los procedimientos establecidos y que las actualizaciones son exitosas.

#### **9. Capacitación y Soporte:**

- **Entrenamiento en SCM:** Proporcionar capacitación a los miembros del equipo sobre los procedimientos de SCM y el uso de las herramientas.
- **Soporte Continuo:** Ofrecer soporte continuo para resolver problemas relacionados con la gestión de la configuración y asegurar que se sigan las mejores prácticas.

#### **10. Monitoreo y Mejora Continua:**

- **Monitoreo de Procesos:** Supervisar los procesos de SCM para identificar áreas de mejora.

- **Revisión y Ajuste de Políticas:** Revisar y ajustar regularmente las políticas y procedimientos de SCM para adaptarse a las necesidades cambiantes del proyecto y las lecciones aprendidas.

## 2.2. Responsabilidades

### 2.2.1. Gerente de Proyecto

- **Visión y Estrategia:** Definir y comunicar la visión general del proyecto y cómo el SCM encaja dentro de esa visión.
- **Planificación de SCM:** Aprobar el plan de gestión de la configuración y asegurar que se alinee con los objetivos del proyecto.
- **Gestión de Recursos:** Asignar los recursos necesarios (humanos y tecnológicos) para la ejecución efectiva del SCM.
- **Supervisión General:** Monitorear el progreso del SCM y asegurar que se cumplan los hitos del proyecto.
- **Gestión de Riesgos:** Identificar y mitigar los riesgos relacionados con la gestión de la configuración.

### 2.2.2. Líder del Equipo de SCM

- **Implementación de Procesos de SCM:** Desarrollar e implementar procedimientos y políticas de SCM.
- **Coordinación del Equipo de SCM:** Supervisar y coordinar las actividades del equipo de SCM.
- **Evaluación y Control de Cambios:** Revisar y aprobar las solicitudes de cambio en la configuración del software.
- **Comunicación:** Actuar como punto de contacto principal para todas las cuestiones relacionadas con SCM y facilitar la comunicación entre equipos.
- **Capacitación:** Asegurar que todos los miembros del equipo estén capacitados en las herramientas y procedimientos de SCM.

### 2.2.3. Ingeniero de SCM / Administrador de Configuración

- **Configuración de Herramientas:** Configurar y mantener las herramientas de control de versiones y automatización utilizadas en el proyecto.
- **Control de Versiones:** Gestionar las ramas de código y asegurar que las políticas de control de versiones se sigan correctamente.
- **Automatización de Builds:** Implementar y mantener scripts de automatización para la construcción y pruebas del software.
- **Monitoreo y Mantenimiento:** Supervisar el estado del sistema de control de versiones y resolver problemas técnicos que surjan.

- **Documentación:** Mantener registros detallados de la configuración y cambios en el software.

#### 2.2.4. Desarrolladores de Software

- **Adherencia a las Políticas de SCM:** Seguir las políticas y procedimientos de SCM al desarrollar y comprometer código.
- **Uso de Control de Versiones:** Usar el sistema de control de versiones para gestionar el código fuente y colaborar con otros desarrolladores.
- **Documentación de Cambios:** Documentar adecuadamente los cambios realizados en el código y proporcionar descripciones claras en los commits.
- **Revisión de Código:** Participar en revisiones de código y fusiones para asegurar la calidad y conformidad con los estándares del proyecto.
- **Pruebas y Validación:** Ejecutar pruebas iniciales y validar que los cambios funcionan como se espera antes de integrarlos en ramas principales.

#### 2.2.5. Tester / Ingeniero de Calidad

- **Validación de Cambios:** Probar los cambios en el software para asegurar que cumplan con los requisitos y no introduzcan errores.
- **Automatización de Pruebas:** Colaborar con el equipo de SCM para implementar pruebas automatizadas que se ejecuten durante los procesos de integración y despliegue.
- **Gestión de Incidencias:** Reportar y rastrear problemas encontrados durante las pruebas en el sistema de seguimiento de incidencias.
- **Revisión de Builds:** Verificar que las construcciones de software sean coherentes y estén libres de defectos antes de la liberación.

#### 2.2.6. Equipo de Operaciones / DevOps

- **Despliegue de Software:** Implementar y mantener los entornos de despliegue para desarrollo, prueba y producción.
- **Automatización de Despliegue:** Configurar y gestionar pipelines de CI/CD para automatizar el despliegue de la aplicación.
- **Monitoreo de Producción:** Monitorear el rendimiento y la disponibilidad de la aplicación en producción.
- **Gestión de Incidentes:** Responder a incidentes en producción y coordinar con el equipo de desarrollo para resolver problemas.

#### 2.2.7. Comité de Control de Cambios (CCB)

- **Revisión de Solicitudes de Cambio:** Evaluar todas las solicitudes de cambio en la configuración del software y decidir sobre su aprobación o rechazo.

- **Evaluación de Impacto:** Analizar el impacto potencial de los cambios en el proyecto y en los otros elementos de configuración.
- **Priorización de Cambios:** Determinar la prioridad de los cambios aprobados y planificar su implementación.

#### 2.2.8. Stakeholders (Clínica “CLINUD”)

- **Requisitos y Expectativas:** Proporcionar requisitos claros y expectativas sobre la funcionalidad y el rendimiento de la aplicación.
- **Revisión de Progresos:** Revisar el progreso del desarrollo y proporcionar retroalimentación continua.
- **Aprobación de Liberaciones:** Aprobar las versiones del software antes de su despliegue final en producción.

#### 2.2.9. Administrador de Librerías de Software

- **Gestión de Librerías:** Asegurar que todas las versiones del software y sus dependencias se almacenan y gestionan adecuadamente en el repositorio.
- **Control de Acceso:** Gestionar los permisos de acceso al repositorio de librerías para asegurar la seguridad y integridad de los componentes almacenados.
- **Actualización de Dependencias:** Mantener actualizadas las dependencias externas y asegurarse de que sean compatibles con el sistema.

#### 2.2.10. Consultor de Seguridad

- **Evaluación de Seguridad:** Evaluar los componentes del software para identificar posibles vulnerabilidades de seguridad.
- **Recomendaciones de Mejora:** Proporcionar recomendaciones sobre cómo mejorar la seguridad del software y las prácticas de SCM.
- **Monitoreo de Conformidad:** Asegurar que las prácticas de SCM cumplan con las políticas de seguridad de la organización y las regulaciones aplicables.

### 2.3. Políticas, directivas y procedimientos aplicables

#### Políticas de SCM

##### 1. Política de Control de Versiones

- **Uso de Herramientas de Control de Versiones:** Todo el código fuente y los artefactos de configuración deben ser gestionados mediante un sistema de control de versiones (Git).
- **Reglas de Nomenclatura:** Todas las ramas, etiquetas y commits deben seguir una convención de nomenclatura estandarizada.



- **Frecuencia de Commits:** Los desarrolladores deben hacer commits frecuentes para mantener el historial de cambios detallado y actualizado.

## 2. Política de Control de Cambios

- **Proceso Formal de Solicitud de Cambios:** Todos los cambios deben ser solicitados, evaluados y aprobados mediante un proceso formal documentado.
- **Evaluación de Impacto de Cambios:** Cada solicitud de cambio debe incluir una evaluación del impacto en el sistema y en los usuarios.
- **Autorización de Cambios:** Solo los cambios aprobados por el Comité de Control de Cambios (CCB) pueden ser implementados.

## 3. Política de Gestión de la Configuración

- **Identificación de Elementos de Configuración:** Todos los elementos de configuración deben ser claramente identificados y documentados.
- **Gestión de Librerías:** Las librerías de software deben ser gestionadas y almacenadas de manera segura y accesible solo para el personal autorizado.
- **Revisión y Auditoría:** La configuración del software debe ser revisada y auditada periódicamente para asegurar la conformidad con los estándares establecidos.

## 4. Política de Seguridad

- **Acceso Controlado:** El acceso a los sistemas de SCM debe estar restringido y controlado mediante autenticación segura.
- **Revisión de Seguridad:** Todo el código y los cambios deben pasar por una revisión de seguridad antes de ser integrados en las ramas principales.
- **Cumplimiento de Normativas:** Todas las prácticas de SCM deben cumplir con las normativas de protección de datos y seguridad de la información aplicables.

## 5. Política de Gestión de Incidencias

- **Registro de Incidencias:** Todas las incidencias deben ser registradas en un sistema de seguimiento (como JIRA) y categorizadas según su impacto y urgencia.
- **Resolución Prioritaria:** Las incidencias críticas que afecten la funcionalidad principal o la seguridad de la aplicación deben ser abordadas con prioridad.

- **Documentación de Resoluciones:** Todas las resoluciones de incidencias deben ser documentadas y revisadas para prevenir futuros problemas similares.

## Directivas de SCM

### 1. Directiva de Integración Continua

- **Automatización de Builds:** Las construcciones de software deben ser automatizadas utilizando herramientas de CI/CD (como Jenkins o GitHub Actions).
- **Pruebas Automáticas:** Todas las integraciones deben pasar por pruebas automáticas para validar la calidad y la funcionalidad del código.
- **Notificación de Errores:** Los fallos en la integración continua deben ser notificados inmediatamente al equipo de desarrollo.

### 2. Directiva de Control de Calidad

- **Estándares de Codificación:** El código debe adherirse a los estándares de codificación definidos y ser revisado regularmente para asegurar la calidad.
- **Revisión de Código:** Todas las fusiones deben ser revisadas por al menos otro miembro del equipo para asegurar la conformidad con los estándares de calidad.
- **Pruebas de Aceptación:** El software debe pasar por pruebas de aceptación antes de ser liberado a los usuarios finales.

### 3. Directiva de Documentación

- **Documentación Completa:** Todo el código y las configuraciones deben estar adecuadamente documentados para facilitar la comprensión y el mantenimiento.
- **Actualización Continua:** La documentación debe ser actualizada continuamente para reflejar los cambios en el sistema.
- **Disponibilidad de Documentación:** Toda la documentación relevante debe estar fácilmente accesible para todos los miembros del equipo.

### 4. Directiva de Despliegue

- **Procedimientos de Despliegue:** El despliegue de la aplicación a los entornos de prueba y producción debe seguir procedimientos documentados y controlados.
- **Ventanas de Mantenimiento:** El despliegue a producción debe realizarse en ventanas de mantenimiento planificadas para minimizar el impacto en los usuarios.

- **Validación Post-Despliegue:** Después del despliegue, se deben realizar validaciones para asegurar que el sistema funcione correctamente.

## 5. Directiva de Seguridad de la Información

- **Gestión de Contraseñas:** Todas las contraseñas y credenciales deben ser gestionadas de forma segura y estar cifradas en reposo.
- **Protección de Datos:** Los datos sensibles de los pacientes deben estar protegidos y cumplir con las regulaciones de privacidad aplicables (como la Ley de Protección de Datos Personales en Chile).
- **Revisión de Vulnerabilidades:** Se deben realizar revisiones periódicas para identificar y mitigar vulnerabilidades de seguridad en la aplicación.

## Procedimientos de SCM

### 1. Procedimiento de Control de Versiones

- **Creación de Ramas:** Crear ramas de feature/\* para nuevas funcionalidades, release/\* para preparaciones de liberación, y hotfix/\* para correcciones urgentes.
- **Combinación de Ramas:** Las ramas feature/\* deben fusionarse en develop mediante pull requests revisados, y las ramas release/\* y hotfix/\* deben fusionarse en main después de la aprobación.
- **Etiquetado de Versiones:** Etiquetar las versiones de producción en la rama main con identificadores de versión únicos.

### 2. Procedimiento de Solicitud y Aprobación de Cambios

- **Iniciar una Solicitud de Cambio:** Crear una solicitud de cambio en el sistema de seguimiento de incidencias detallando la descripción, el impacto y la justificación del cambio.
- **Revisión del Cambio:** El Comité de Control de Cambios (CCB) revisa la solicitud y evalúa el impacto, la viabilidad y la alineación con los objetivos del proyecto.
- **Decisión y Registro:** El CCB aprueba o rechaza el cambio, y se registra la decisión junto con cualquier comentario o condición.

### 3. Procedimiento de Construcción y Pruebas

- **Construcción Automática:** Configurar las herramientas de CI para ejecutar construcciones automáticas del software cada vez que se fusionen cambios en develop o main.
- **Ejecución de Pruebas:** Automatizar las pruebas unitarias, de integración y de aceptación para ejecutarse durante el proceso de construcción.

- **Revisión de Resultados:** Revisar los resultados de las pruebas y solucionar cualquier problema antes de proceder con el despliegue.

#### 4. Procedimiento de Despliegue

- **Preparación del Despliegue:** Asegurar que todos los cambios han sido probados y aprobados antes de planificar el despliegue.
- **Despliegue a Entornos de Prueba:** Implementar primero en un entorno de prueba y validar la funcionalidad y el rendimiento.
- **Despliegue a Producción:** Realizar el despliegue a producción siguiendo el procedimiento documentado y dentro de las ventanas de mantenimiento planificadas.
- **Verificación Post-Despliegue:** Realizar verificaciones para asegurar que la aplicación esté funcionando correctamente y sin errores.

#### 5. Procedimiento de Gestión de Incidencias

- **Registro de Incidencias:** Registrar todas las incidencias detectadas en el sistema de seguimiento, clasificándolas por severidad y prioridad.
- **Investigación y Diagnóstico:** Investigar la causa raíz de las incidencias y proponer soluciones.
- **Resolución y Cierre:** Implementar soluciones para resolver las incidencias y cerrar los tickets en el sistema de seguimiento una vez confirmada la resolución.

#### 6. Procedimiento de Revisión y Auditoría

- **Revisiones Periódicas:** Realizar revisiones regulares de la configuración y los procedimientos de SCM para asegurar su conformidad y efectividad.
- **Auditorías de Configuración:** Llevar a cabo auditorías para verificar que todos los elementos de configuración están correctamente gestionados y documentados.
- **Informes de Auditoría:** Documentar los hallazgos de las auditorías y las acciones correctivas recomendadas.

### 3. Actividades de SCM

Identifica todas las actividades y tareas que se requieren para el manejo de la configuración del sistema. Estas deben ser tanto actividades técnicas como de gestión de SCM, así como las actividades generales del proyecto que tengan implicancia sobre el manejo de configuración.

#### 3.1. Identificación de la configuración

##### 3.1.1. Elementos de configuración

Para este proyecto los elementos de configuración se corresponderán con los entregables definidos en el Modelo de Proceso, aunque no necesariamente todos los entregables deben ser elementos de configuración.

La decisión de cuales de los entregables serán elementos de configuración será tomada por el SCMR, quién deberá tomar en cuenta qué productos serán necesarios cuando se quiera recuperar una versión completa del sistema.

Se debe generar una línea base por iteración en cada Fase, de acuerdo a lo siguiente:

- Los eventos que dan origen a la línea base.
- Los elementos que serán controlados en la línea base.
- Los procedimientos usados para establecer y cambiar la línea base.
- La autorización requerida para aprobar cambios a los documentos de la línea base.

### 3.1.2. Nomenclatura de Elementos

En esta sección se especifican la identificación y descripción única de cada elemento de configuración.

Además se especifica como se distinguirán las diferentes versiones de cada elemento.

Para todos los elementos de configuración se les deberá agregar, después del nombre del mismo, información acerca del grupo al que corresponde el elemento y la versión del mismo.

El formato para esta nomenclatura es: **NomenclaturaGXvY**.extensión, donde:

- Nomenclatura es la especificada mas abajo para cada elemento.
- X es un número de 1 dígito que identifica al grupo.
- Y indica la versión del elemento de configuración o entregable.
- Extensión indica la extensión del elemento de configuración o entregable.

[Ejemplo: RQALSG1v2.doc, es como se deberá llamar el entregable "Alcance del Sistema" correspondiente al grupo 1 y cuya versión del documento es la 2.]

Para los entregables, se deberá identificar a que Fase e iteración corresponden en forma manual. Esto es: para los elementos bajo control de configuración se los almacenará de forma que se puedan recuperar dada la Fase e iteración a la que corresponden, y para los elementos que no se encuentran bajo control de configuración podrán ser almacenados por ejemplo en carpetas que identifiquen la Fase e iteración a la que pertenecen.

Se indica la siguiente nomenclatura para cada entregable en el modelo de proceso, según la disciplina (en caso que exista algún elemento de configuración que se agregue a los que se detallan abajo, se deberá incluir en las tablas siguientes de acuerdo a la disciplina a la que pertenece, indicando la nomenclatura usada):

#### Requerimientos:

Nomenclatura	Entregable
RQACT	Acta de Reunión de Requerimientos
RQDRQ	Especificación de Requerimientos
RQMOD	Modelo de Casos de Uso
RQRSU	Requerimientos Suplementarios
RQDVC	Documento de Validación con el Cliente
RQPIU	Pautas para Interfase de Usuario
RQRCA	Requerimientos Candidatos
RQALS	Alcance del Sistema
RQGLO	Glosario
RQOOMDO	Modelo de Dominio
RQOODRP	Documento de Requerimientos para el Prototipo
RQGXMOM	Nomenclatura

Diseño:

Nomenclatura	Entregable
DSMDI	Modelo de Diseño
DSARQ	Descripción de la Arquitectura
DSOOMDA	Modelo de Datos
DSOODDP	Documento de Diseño del Prototipo

Implementación:

Nomenclatura	Entregable
IMEDT	Estándar de Documentación Técnica
IMEI	Estándar de Implementación
IMPR	Prototipo
IMIIN	Informe de Integración
IMDT	Documentación técnica
IMIVU	Informe de Verificación Unitaria
IMOOPH	Plan de Integración de la Iteración
IMOOMIM	Modelo de Implementación
IMOOEJI	Ejecutable de la Iteración
IMOORRP	Reporte de Revisión por Pares
IMOOCVU	Clases de la Verificación Unitaria de Módulo
IMGXICO	Informe de Consolidación
IMGXEST	BC Con Estilos
IMGXCON	BC Consolidado
IMGXNUC	BC Núcleo
IMGXMOD	BC Módulo

Verificación:

Nomenclatura	Entregable
VRPVV	Plan de Verificación y Validación
VRDAP	Documento de Evaluación y Ajuste del Plan de V & V

VRPVI	Plan de Verificación de la Iteración
VRMCP	Modelo de Casos de Prueba
VRIVD	Informe de Verificación de Documento
VRIVI	Informe de Verificación de Integración
VRIVS	Informe de Verificación del Sistema
VRRPR	Reportes de Pruebas
VREV	Evaluación de la Verificación
VRIFV	Informe Final de Verificación

Implantación (IP):

Nomenclatura	Entregable
IPMSU	Materiales para Soporte al Usuario (Se pueden usar sufijos para identificar cada ítem dentro del material Ej. IPMSUMU para Manual de Usuario)
IPMCA	Materiales para Capacitación
IPPS	Presentación del Sistema
IPPLA	Plan de Implantación
IPVPR	Versión del Producto
IPOOEDU	Estándar de Documentación de Usuario
IPOORFPA	Reporte Final de Pruebas de Aceptación

Gestión de Configuración y Control de Cambios (SCM):

Nomenclatura	Entregable
SCMPLA	Plan de Configuración
SCMMAC	Manejo del Ambiente Controlado
SCMGC	Gestión de Cambios
SCMRV	Registro de Versiones
SCMILB	Informe de la Línea Base del Proyecto
SCMIF	Informe Final de SCM

Gestión de Calidad (SQA):

Nomenclatura	Entregable
SQAPLA	Plan de Calidad
SQADAP	Documento de Evaluación y Ajuste del Plan de Calidad
SQARTF	Informe de RTF
SQAES	Entrega Semanal de SQA
SQAIR	Informe de Revisión de SQA
SQADV	Descripción de la Versión
SQANV	Notas de la Versión
SQAIF	Informe Final de SQA

Gestión de Proyecto (GP):

Nomenclatura	Entregable
GPPLA	Plan de Proyecto

GPISP	Informe de Situación del Proyecto
GPEM	Estimaciones y Mediciones
GPDRI	Documento de Riesgos
GPRAC	Registro de Actividades
GPIFP	Informe Final de Proyecto
GPARE	Acta de la Reunión de Equipo
GPPIT	Plan de la Iteración
GPPDE	Plan de Desarrollo
GPICF	Informe de Conclusiones de la Fase
GPPDIP	Presentación en Diapositivas del Proyecto
GPPDP	Presentación al Director del Proyecto
GPARD	Acta de la Reunión con el Director del Proyecto
GPOODAP	Documento de Evaluación y Ajuste al Plan de Proyecto
GPIARI	Acta de la Reunión de Integración

Comunicación (COM):

Nomenclatura	Entregable
COMDI	Documento Informativo
COMENS	Encuesta de Satisfacción del Cliente
COMEVS	Evaluación de Satisfacción del Cliente

### 3.1.3. Elementos de la Línea Base del Proyecto

[En esta sección se debe detallar la Línea Base. Esto es, los elementos que pertenecen a la Línea Base del Proyecto, especificados por Fase del Proyecto y por iteraciones dentro de cada Fase.]

#### *FASE: Iniciación*

#### **ITERACIÓN: 1**

Elemento	Descripción	Disciplina
<b>Documento de Alcance</b>	Define los objetivos, alcance, y delimitaciones del proyecto.	Gestión de Proyecto
<b>Requisitos de Usuarios</b>	Detalla las necesidades y expectativas de los usuarios finales.	Análisis de Requisitos
<b>Acta de Constitución</b>	Documento formal que autoriza el inicio del proyecto.	Gestión de Proyecto
<b>Plan de Comunicación</b>	Estrategias y canales de comunicación con stakeholders.	Gestión de Proyecto



## *FASE: Planificación*

### **ITERACIÓN: 1**

<b>Elemento</b>	<b>Descripción</b>	<b>Disciplina</b>
<b>Plan del Proyecto</b>	Documenta la planificación global del proyecto.	Gestión de Proyecto
<b>Cronograma del Proyecto</b>	Define las fechas clave y los plazos de entregables.	Gestión de Proyecto
<b>Plan de Gestión de Riesgos</b>	Identifica y evalúa los riesgos potenciales del proyecto.	Gestión de Proyecto
<b>Plan de Recursos</b>	Detalla los recursos necesarios (humanos y materiales).	Gestión de Proyecto
<b>Matriz de Trazabilidad de Requisitos</b>	Mapas de requisitos de usuarios a funcionalidades.	Análisis de Requisitos

### **ITERACIÓN: 2**

<b>Elemento</b>	<b>Descripción</b>	<b>Disciplina</b>
<b>Documentos de Requisitos Funcionales</b>	Lista detallada de las funcionalidades requeridas.	Análisis de Requisitos
<b>Documentos de Requisitos No Funcionales</b>	Especificaciones de rendimiento, seguridad, etc.	Análisis de Requisitos
<b>Estudios de Viabilidad Técnica</b>	Evaluación de la factibilidad técnica de las soluciones.	Arquitectura de Sistemas
<b>Prototipos Iniciales</b>	Prototipos de baja fidelidad para validar conceptos de UI/UX.	Diseño de UI/UX

## *FASE: Desarrollo*

### **ITERACIÓN: 1**

<b>Elemento</b>	<b>Descripción</b>	<b>Disciplina</b>
<b>Arquitectura del Sistema</b>	Diseño general de la arquitectura del software y su estructura.	Arquitectura de Sistemas
<b>Especificación Técnica</b>	Detalles técnicos de cómo se implementarán las funcionalidades.	Desarrollo de Software
<b>Plan de Integración Continua</b>	Estrategia y herramientas para la CI/CD.	Desarrollo de Software
<b>Entorno de Desarrollo</b>	Configuración del entorno de desarrollo incluyendo herramientas.	Desarrollo de Software
<b>Bases de Datos</b>	Diseño de la estructura de la base de datos.	Desarrollo de Software

## ITERACIÓN: 2

Elemento	Descripción	Disciplina
Código Fuente	Código fuente inicial de la aplicación móvil.	Desarrollo de Software
Pruebas Unitarias	Planes y casos de pruebas unitarias para el código desarrollado.	Aseguramiento de Calidad
Configuraciones de Entorno	Configuraciones específicas para entornos de prueba y desarrollo.	Desarrollo de Software
Documentación Técnica	Documentación de la implementación y APIs del sistema.	Desarrollo de Software
Plan de Pruebas	Estrategia y casos de prueba para la fase de desarrollo.	Aseguramiento de Calidad

## FASE: Pruebas

## ITERACIÓN: 1

Elemento	Descripción	Disciplina
Casos de Prueba de Integración	Casos de prueba para la integración de diferentes módulos.	Aseguramiento de Calidad
Ambiente de Pruebas	Configuración del entorno de pruebas incluyendo bases de datos.	Aseguramiento de Calidad
Resultados de Pruebas Unitarias	Resultados y análisis de las pruebas unitarias realizadas.	Aseguramiento de Calidad
Informes de Incidencias	Documentación de cualquier defecto encontrado durante las pruebas.	Aseguramiento de Calidad
Manual de Pruebas	Procedimientos y criterios para la realización de pruebas.	Aseguramiento de Calidad

## ITERACIÓN: 2

Elemento	Descripción	Disciplina
Pruebas de Aceptación	Casos y resultados de pruebas para validar la aceptación por parte del cliente.	Aseguramiento de Calidad
Informe de Validación de Usuario	Documentación de la validación del sistema por parte de usuarios clave.	Aseguramiento de Calidad
Reporte de Calidad	Evaluación final de la calidad del software antes del despliegue.	Aseguramiento de Calidad
Lista de Verificación de Despliegue	Checklist para asegurar que todos los requisitos se cumplen antes del despliegue.	Aseguramiento de Calidad
Documentación de Despliegue	Guía y procedimientos para la implementación en producción.	Aseguramiento de Calidad

## **FASE: Despliegue**

### **ITERACIÓN: 1**

<b>Elemento</b>	<b>Descripción</b>	<b>Disciplina</b>
<b>Plan de Despliegue</b>	Estrategia y procedimientos detallados para el despliegue del sistema.	Gestión de Proyecto
<b>Paquete de Liberación</b>	Versión final de la aplicación preparada para su despliegue.	Desarrollo de Software
<b>Manual de Usuario Final</b>	Guía completa para que los usuarios finales operen la aplicación.	Documentación
<b>Capacitación de Usuarios</b>	Materiales y plan para la formación de los usuarios.	Gestión del Cambio
<b>Checklist de Post-Despliegue</b>	Tareas a realizar inmediatamente después del despliegue para asegurar el correcto funcionamiento.	Operaciones

### **ITERACIÓN: 2**

<b>Elemento</b>	<b>Descripción</b>	<b>Disciplina</b>
<b>Validación Post-Despliegue</b>	Confirmación de que la aplicación está funcionando como se espera en producción.	Operaciones
<b>Informe de Desempeño en Producción</b>	Evaluación del rendimiento y estabilidad del sistema en el entorno de producción.	Operaciones
<b>Plan de Mantenimiento</b>	Estrategia para la gestión del mantenimiento continuo de la aplicación.	Gestión del Cambio
<b>Recolección de Feedback</b>	Proceso para recoger la retroalimentación de los usuarios después del despliegue.	Gestión de Proyecto
<b>Plan de Actualización</b>	Plan para futuras actualizaciones y mejoras basadas en el feedback recibido.	Gestión del Cambio

#### **3.1.4. Recuperación de los Elementos de configuración**

##### **Objetivos de la Recuperación de Elementos de Configuración**

1. **Garantizar la Disponibilidad:** Asegurar que todos los elementos de configuración pueden ser recuperados rápidamente y de manera completa en caso de pérdida o daño.
2. **Minimizar el Tiempo de Inactividad:** Reducir el tiempo necesario para restaurar los elementos de configuración a su estado operativo en caso de incidentes.
3. **Asegurar la Integridad de los Datos:** Garantizar que los elementos de configuración recuperados son completos y no han sido comprometidos.
4. **Mantener la Continuidad del Proyecto:** Permitir que el proyecto continúe sin interrupciones significativas después de una recuperación.

## Procedimientos de Recuperación

Para estructurar el plan de recuperación, se pueden detallar los procedimientos por tipo de elemento de configuración:

### 1. Código Fuente

- **Almacenamiento Primario:** El código fuente se gestiona a través de un sistema de control de versiones (por ejemplo, Git) y se almacena en un repositorio remoto seguro (como GitHub, GitLab o Bitbucket).
- **Procedimiento de Recuperación:**
  1. **Identificar el Punto de Restauración:** Determinar la versión o commit específico que se necesita restaurar.
  2. **Clonar o Extraer del Repositorio:** Usar el comando de clonación o extracción para obtener el código desde el repositorio remoto.
  3. **Validar la Integridad:** Verificar que todos los archivos han sido recuperados y que el código se compila y ejecuta correctamente.

### 2. Documentación del Proyecto

- **Almacenamiento Primario:** Documentación guardada en una plataforma de colaboración (como Confluence, Google Drive o SharePoint) con versiones controladas.
- **Procedimiento de Recuperación:**
  1. **Acceder a la Plataforma de Colaboración:** Ingresar al sistema donde se almacena la documentación.
  2. **Seleccionar la Versión Necesaria:** Buscar la versión de la documentación que se necesita restaurar.
  3. **Descargar o Restaurar:** Descargar el documento o restaurar la versión directamente en la plataforma.

### 3. Configuraciones de Entorno

- **Almacenamiento Primario:** Archivos de configuración guardados en un sistema de gestión de configuraciones o en el repositorio de control de versiones.
- **Procedimiento de Recuperación:**
  1. **Localizar los Archivos de Configuración:** Acceder al sistema donde se almacenan los archivos de configuración.
  2. **Obtener la Versión Correcta:** Recuperar la versión específica del archivo de configuración que corresponde al entorno a restaurar.

3. **Aplicar la Configuración:** Copiar los archivos de configuración al entorno apropiado y reiniciar los servicios si es necesario.

#### 4. Bases de Datos

- **Almacenamiento Primario:** Datos almacenados en sistemas de gestión de bases de datos (como MySQL, PostgreSQL, MongoDB) y respaldos periódicos almacenados en un servidor seguro o servicio de nube.
- **Procedimiento de Recuperación:**
  1. **Acceder al Respaldo de la Base de Datos:** Localizar el archivo de respaldo más reciente o el punto de recuperación.
  2. **Restaurar la Base de Datos:** Usar herramientas de gestión de bases de datos para importar el respaldo a la base de datos en el servidor.
  3. **Verificar la Integridad de los Datos:** Realizar pruebas para asegurar que los datos han sido restaurados correctamente.

#### 5. Paquetes de Construcción y Artefactos de Despliegue

- **Almacenamiento Primario:** Artefactos almacenados en un repositorio de binarios (como JFrog Artifactory, Nexus Repository) o en un servicio de CI/CD.
- **Procedimiento de Recuperación:**
  1. **Acceder al Repositorio de Binarios:** Ingresar al sistema donde se almacenan los artefactos de construcción.
  2. **Identificar la Versión Correcta:** Buscar y seleccionar la versión específica del artefacto que se necesita desplegar.
  3. **Descargar y Desplegar:** Descargar el artefacto y seguir los procedimientos de despliegue estándar para implementar el artefacto en el entorno deseado.

#### 6. Planes y Resultados de Pruebas

- **Almacenamiento Primario:** Planes y resultados de pruebas almacenados en herramientas de gestión de pruebas (como TestRail, JIRA, o ALM) o en sistemas de almacenamiento de documentos.
- **Procedimiento de Recuperación:**
  1. **Acceder a la Herramienta de Gestión de Pruebas:** Ingresar al sistema donde se almacenan los planes y resultados de las pruebas.
  2. **Seleccionar la Prueba o Resultado Necesario:** Buscar el plan de pruebas o resultado específico que se requiere.
  3. **Descargar o Visualizar:** Descargar el documento o revisar los resultados directamente en la plataforma.

## Estrategias de Copia de Seguridad y Recuperación

Para asegurar que los elementos de configuración están disponibles en caso de desastre, se deben seguir las siguientes estrategias:

### 1. Copia de Seguridad Regular:

- **Frecuencia:** Realizar copias de seguridad de los elementos de configuración críticos de forma diaria o semanal, dependiendo de la importancia y la frecuencia de los cambios.
- **Almacenamiento Seguro:** Guardar las copias de seguridad en múltiples ubicaciones (local y en la nube) para mitigar el riesgo de pérdida total.

### 2. Pruebas de Recuperación:

- **Simulacros de Recuperación:** Realizar pruebas periódicas de recuperación para asegurar que los procedimientos funcionan como se espera.
- **Evaluación y Mejora:** Revisar los resultados de las pruebas y ajustar los procedimientos según sea necesario para mejorar la eficiencia y la efectividad.

### 3. Control de Acceso y Seguridad:

- **Autenticación y Autorización:** Implementar controles estrictos de acceso a los sistemas de almacenamiento de copias de seguridad y de recuperación.
- **Encriptación:** Utilizar encriptación para proteger los datos sensibles en tránsito y en reposo.

### 4. Automatización de Copias de Seguridad:

- **Herramientas de Backup:** Usar herramientas automatizadas para realizar copias de seguridad y gestionarlasy, reduciendo la dependencia de procesos manuales.

### 5. Monitoreo y Alerta:

- **Sistemas de Monitoreo:** Implementar sistemas de monitoreo para detectar fallos en el proceso de copia de seguridad y recuperación.
- **Notificaciones de Incidentes:** Configurar alertas para notificar a los responsables en caso de fallos o incidentes.

## Roles y Responsabilidades en la Recuperación

- **Administrador de Sistemas:** Responsable de la gestión y recuperación de configuraciones del entorno y bases de datos.

- **Ingeniero de DevOps:** Encargado de la recuperación de los paquetes de construcción y artefactos de despliegue.
- **Gestor de Proyectos:** Supervisa la recuperación de la documentación del proyecto y asegura la disponibilidad de los planes y resultados de pruebas.
- **Equipo de Desarrollo:** Recupera el código fuente y verifica la integridad después de la restauración.
- **Responsable de SCM:** Coordina y verifica que todos los elementos de configuración críticos están recuperados y funcionales.

### 3.2. Control de configuración

En esta sección se detallan las actividades de solicitud, evaluación, aprobación e implementación de cambios a los elementos de la línea base.

Los cambios apuntan tanto a la corrección como al mejoramiento.

El procedimiento que se describe a continuación es el que se utilizará cada vez que se precise introducir un cambio al sistema.

Se entiende por cambio al sistema, las modificaciones que afecten a la línea base del sistema, como pueden ser:

- Cambios en los Requerimientos.
- Cambios en el Diseño.
- Cambios en la Arquitectura.
- Cambios en las herramientas de desarrollo.
- Cambios en la documentación del proyecto. (agregar nuevos documentos o modificar la estructura de los existentes)

#### 3.2.1. Solicitud de cambios

Cuando se realiza la solicitud de un cambio, se actualiza el documento de “Solicitud de cambio” para registrar esta solicitud.

Se debe ingresar toda la información necesaria, detallada en el documento.

#### 3.2.2. Evaluación de cambios o Análisis de Impacto

La evaluación del cambio involucra determinar qué es necesario hacer para implementar el cambio y la estimación de sus costos y plazos.

Se realiza en 2 pasos:

1. Planificación de la evaluación del cambio que involucra:
  - Revisar la solicitud de cambio para entender su alcance. (Si es necesario se discute con el originador para aclarar el alcance de lo propuesto y los motivos de la solicitud.
  - Determinar las personas del proyecto que deben realizar el análisis de evaluación del cambio e involucrarlas.
  - Desarrollar un Plan para la evaluación del cambio.

- Si el cambio involucra al Cliente, obtener el acuerdo de éste con el Plan.

## 2. Evaluar el cambio:

Dependiendo de las características del cambio, la evaluación del cambio puede ser realizado por el Administrador o ser delegado a otras personas del proyecto.

Se debe determinar el impacto en:

- Los productos técnicos.
- Los Planes de proyecto.
- Los acuerdos con el Cliente.
- Los Riesgos del proyecto.

### 3.2.3. Aprobación o desaprobación de cambios

Se debe formar el “Comité de Control de Configuración” y determinar su autoridad para la aprobación de cambios.

La composición de este comité puede variar según el tipo de cambio y las líneas de trabajo involucradas en él.

Se sugieren como posibles integrantes:

- Administrador (obligatorio)
- Arquitecto (opcional)
- Analista (opcional)
- Implementador (opcional)
- SCM (obligatorio)
- Cliente (opcional)

Se define un comité de Control de Configuración de nivel superior, compuesto por el Gerente de proyecto, al cual se elevarán las solicitudes de cambios cuya aprobación o desaprobación no se pueda resolver por el primer comité.

### 3.2.4. Implementación de cambios

Una vez realizada la evaluación del cambio, se decide en qué momento implementarlo. Esta etapa involucra los procesos necesarios para implementar la solicitud y monitorear el progreso del trabajo.

Además se especificará el momento de liberación del cambio; así como también los responsables de las actividades que involucra el cambio.

Recordando que nos basamos en un proceso de desarrollo incremental e iterativo, donde en cada iteración se realizan tareas de Análisis de requerimientos, Diseño, Implementación y Verificación; se debe introducir el cambio en el área que lo originó y continuar con las actividades del ciclo (Requerimientos, Análisis, Diseño, Implementación, Verificación) que impactarán los elementos de la línea base correspondientes a cada actividad.

### 3.3. Estado de la configuración

[Las actividades de control de estado son para reunir información y reportar el estado de los elementos de configuración.

Se debe especificar lo siguiente:



- Qué elementos serán revisados de la línea base y por cambios a realizarse.
- Qué tipos de reportes de estado serán generados y con qué frecuencia.
- Como la información será obtenida, guardada, procesada, y reportada.
- Como será controlado el acceso a los datos de estado.

Si se utiliza una herramienta automática deberá ser especificada su funcionalidad y modo de uso explícitamente o por referencia.

En los reportes de estado de los elementos de configuración se debe incluir como mínimo la siguiente información:

- Su primer versión aprobada.
- El estado de los cambios solicitados.
- El estado de implementación de los cambios aprobados.]

### **3.4. Auditorias y revisiones de configuración**

Se realizarán auditorias de la línea base antes de una liberación de ésta o de una actualización de la versión de un componente prioritario de ésta.

Estas auditorias incluirán:

- Objetivo: el objetivo de todas las auditorías es verificar que en un momento dado la línea base se compone de una colección consistente y bien definida de productos.
- Elementos de configuración bajo auditoría: se elegirán uno o mas elementos de configuración de mayor prioridad en la línea base.
- Agenda de auditorías: antes de la liberación o actualización.
- Conducción: las auditorías serán dirigidas por el SCMR.
- Participantes: SCMR y los autores de los elementos de configuración a auditar.
- Documentos Requeridos: Documentos de SCR y reportes de estado de la configuración generados.
- Reportes de Deficiencias y Acciones Correctivas: determinadas por los participantes.
- Criterio de Aprobación: lo determina el SCMR.

### **3.5. Control de Interfases**

Las actividades de Control de Interfases controlan los cambios a los elementos de configuración del proyecto, que modifican las interfases con elementos fuera del alcance del Plan.

Este control será llevado por el SCMR como parte del control de la configuración.

### **3.6. Control de subcontratos y vendedores**

N/a.

## **4. Calendario**

## Actividades y Eventos Relevantes

### 1. Definición de la Línea Base

- **Fecha de Inicio:** 9 de julio de 2024
- **Fecha de Finalización:** 15 de julio de 2024
- **Descripción:** Establecimiento de la línea base que incluye la configuración inicial de todos los elementos críticos del proyecto, como el código fuente, la documentación y las configuraciones del entorno.

### 2. Implementación de Control de Cambios

- **Fecha de Inicio:** 16 de julio de 2024
- **Fecha de Finalización:** 22 de julio de 2024
- **Descripción:** Despliegue y puesta en funcionamiento del proceso formal de control de cambios para gestionar todas las modificaciones en los elementos de configuración del proyecto.

### 3. Auditorías

- **Primera Auditoría**
  - **Fecha:** 30 de julio de 2024
  - **Objetivo:** Revisión inicial para verificar la conformidad con los estándares de gestión de configuración y procedimientos establecidos.
- **Segunda Auditoría**
  - **Fecha:** 15 de agosto de 2024
  - **Objetivo:** Evaluación posterior a la implementación para asegurar la eficacia de las prácticas de SCM y realizar ajustes si es necesario.

## Dependencias entre Actividades de SCM y Hitos del Proyecto

### • Inicio del Desarrollo de Software

- Dependencia de la definición de la línea base para tener una configuración inicial estable y consistente.
- Fecha de inicio: 23 de julio de 2024

### • Prueba de Aceptación del Cliente

- Dependencia de la implementación del control de cambios para asegurar que las versiones entregadas sean controladas y aprobadas adecuadamente.

- Fecha prevista: 2 de septiembre de 2024

- **Despliegue en Producción**

- Dependencia de las auditorías para garantizar que todas las configuraciones y procesos estén validados.
- Fecha prevista: 9 de septiembre de 2024

## 5. Recursos

### Herramientas de Software

#### 1. Sistema de Control de Versiones (VCS):

- **Ejemplo:** Git (GitHub, GitLab, Bitbucket)
- **Descripción:** Para la gestión del código fuente, seguimiento de cambios y colaboración en el desarrollo.

#### 2. Herramienta de Gestión de Configuraciones:

- **Ejemplo:** Ansible, Puppet, Chef
- **Descripción:** Automatización de la configuración de sistemas y despliegues para mantener la coherencia del entorno.

#### 3. Herramienta de Gestión de Tareas y Proyectos:

- **Ejemplo:** JIRA, Trello, Asana
- **Descripción:** Para la planificación, seguimiento y asignación de tareas relacionadas con la gestión de configuración y el desarrollo.

#### 4. Plataforma de Integración Continua y Despliegue Continuo (CI/CD):

- **Ejemplo:** Jenkins, CircleCI, GitLab CI/CD
- **Descripción:** Automatización de pruebas, integración y despliegue de aplicaciones para asegurar la calidad y la entrega continua.

#### 5. Herramienta de Monitoreo y Seguimiento:

- **Ejemplo:** Prometheus, ELK Stack (Elasticsearch, Logstash, Kibana)
- **Descripción:** Monitoreo de infraestructura y aplicaciones para identificar problemas y mejorar el rendimiento.

### Técnicas

- **Prácticas de Gestión de Configuración:**

- Establecimiento de procesos formales para la gestión de versiones, control de cambios, auditorías y gestión de liberaciones.

- **Automatización de Procesos:**
  - Implementación de scripts y herramientas de automatización para agilizar tareas repetitivas como compilación, prueba y despliegue.

## **Equipamiento**

- **Servidores y Almacenamiento:**
  - Infraestructura para alojar repositorios de código, artefactos de construcción, y bases de datos.
- **Estaciones de Trabajo:**
  - Equipos de desarrollo con capacidad adecuada para ejecutar herramientas de desarrollo y pruebas.

## **Personal**

- **Responsable de SCM:**
  - Persona encargada de supervisar y ejecutar las actividades de gestión de configuración, asegurando la implementación correcta de políticas y procedimientos.
- **Equipo de Desarrollo:**
  - Ingenieros y desarrolladores responsables de contribuir al repositorio de código y seguir las prácticas de SCM establecidas.
- **Equipo de Operaciones (Ops):**
  - Administradores de sistemas y DevOps que implementan y mantienen la infraestructura necesaria para el entorno de desarrollo y producción.

## **Capacitación Necesaria**

- **Formación en Herramientas y Técnicas:**
  - Capacitación en el uso efectivo de herramientas como Git, Ansible, Jenkins, y prácticas de CI/CD.
- **Políticas y Procedimientos de SCM:**
  - Entrenamiento en las políticas y procedimientos específicos de gestión de configuración y control de cambios adoptados por el proyecto.

## **Gestión de Recursos**

- **Asignación de Recursos:** Designar roles y responsabilidades claras para cada miembro del equipo de desarrollo y operaciones.

- **Optimización de Recursos:** Utilizar herramientas de automatización para maximizar la eficiencia y minimizar errores en las operaciones de SCM.
- **Monitoreo y Mejora Continua:** Evaluar regularmente el rendimiento de las herramientas y procesos, y ajustar según sea necesario para mejorar la productividad y la calidad del software.

## 6. Mantenimiento del Plan de SCM

### Responsabilidades

- **Responsable de Monitoreo del Plan de SCM:**
  - Fabián Yépez, Responsable de SCM designado en el equipo del proyecto será responsable de monitorear y gestionar el cumplimiento del Plan de SCM.

### Frecuencia de Modificaciones al Plan

- **Frecuencia:**
  - Las modificaciones al Plan de SCM pueden ser necesarias en función de cambios en el proyecto, actualizaciones de herramientas, feedback del equipo, o revisiones periódicas de la eficacia de los procesos.

### Evaluación y Aprobación de Cambios al Plan

- **Proceso de Evaluación:**
  - Los cambios propuestos al Plan de SCM serán evaluados por el Responsable de SCM en colaboración con el equipo de gestión del proyecto y los expertos pertinentes.
  - Se considerarán aspectos como la necesidad del cambio, impacto potencial en los procesos existentes, y alineación con los objetivos del proyecto.
- **Aprobación:**
  - Los cambios al Plan de SCM serán aprobados por el Director del Proyecto o el Comité de Gestión del Proyecto, según las políticas establecidas.

### Implementación y Comunicación de Cambios al Plan

- **Implementación:**
  - Una vez aprobados, los cambios al Plan de SCM serán implementados por el Responsable de SCM y el equipo pertinente según los procedimientos definidos.
- **Comunicación:**

- Se comunicarán los cambios al Plan de SCM a todos los miembros del equipo de desarrollo y operaciones involucrados.
- Se utilizarán herramientas de comunicación interna como correos electrónicos, reuniones de equipo, o plataformas de gestión de proyectos para asegurar que todos estén informados y actualizados.