Ingeniería de Software

**Gestión de la Configuración (CM)**

***Definición de un Proceso de Control del Cambio***

MIEMBROS DEL GRUPO 1.2:

* Nicolás Barcia Quintela
* Jesús Campos Manjón
* Guillermo Madriñán González
* Roi Martínez Enríquez

FECHA DE ENTREGA: 18/03/2024

|  |  |  |
| --- | --- | --- |
| **CONTROL DE VERSIONES** | | |
| **VERSION** | **FECHA** | **DESCRIPCIÓN DEL CAMBIO** |
| 1.0 | 30/01/24 | Creación del documento y cambios iniciales. |
| 2.0 | 07/02/24 | Modificación del documento tras las correcciones del grupo 1.1. |
| 3.0 | 20/02/24 | Readaptación del documento a la inclusión de Git |
| 4.0 | 17/03/24 | Inclusión de los apartados relacionados con la ejecución del proceso |

**ÍNDICE**

[Información sobre la práctica a realizar 1](#_Toc161681969)

[Descripción de la práctica 1](#_Toc161681970)

[Descripción del grupo de trabajo 1](#_Toc161681971)

[Seguimiento de la práctica 1](#_Toc161681972)

[DOCUMENTACIÓN DE LA PRÁCTICA 2](#_Toc161681973)

[Descripción del proceso de control de cambios 2](#_Toc161681974)

[Descripción del proceso de control de versiones con Git 2](#_Toc161681975)

[Descripción del repositorio 2](#_Toc161681976)

[Descripción de uso de Póker Scrum 4](#_Toc161681977)

[Diagrama de actividades 5](#_Toc161681978)

[Definición de Actividades 6](#_Toc161681979)

[Identificación del problema y solicitud de cambio 6](#_Toc161681980)

[Evaluación del desarrollador 7](#_Toc161681981)

[Evaluación del CCB 8](#_Toc161681982)

[Notificación de rechazo 10](#_Toc161681983)

[Planificación 11](#_Toc161681984)

[Implementación 13](#_Toc161681985)

[Pruebas 15](#_Toc161681986)

[Revisión 16](#_Toc161681987)

[Cierre 17](#_Toc161681988)

[Plantillas del proceso 19](#_Toc161681989)

[Plantilla de Identificación del problema y Solicitud de cambio 19](#_Toc161681990)

[Plantilla de Evaluación del desarrollador 20](#_Toc161681991)

[Plantilla de Evaluación del CCB 22](#_Toc161681992)

[Plantilla de Notificación de rechazo 23](#_Toc161681993)

[Plantilla de Planificación 25](#_Toc161681994)

[Plantilla de Implementación 27](#_Toc161681995)

[Plantilla de Pruebas 29](#_Toc161681996)

[Plantilla de Revisión 31](#_Toc161681997)

[Plantilla de Cierre 32](#_Toc161681998)

[ANEXOS 35](#_Toc161681999)

[Anexo 1.- Cambios propuestos por cada miembro del grupo 35](#_Toc161682000)

[Nicolás Barcia Quintela 35](#_Toc161682001)

[Jesús Campos Manjón 35](#_Toc161682002)

[Guillermo Madriñán González 35](#_Toc161682003)

[Roi Martínez Enríquez 35](#_Toc161682004)

[Anexo 2.- Ejecución del proceso 36](#_Toc161682005)

[Organización del repositorio 36](#_Toc161682006)

[Distribución del trabajo 36](#_Toc161682007)

[Cambio de planes respecto a la versión 3 del documento 37](#_Toc161682008)

[Problemas encontrados y soluciones 38](#_Toc161682009)

[Gestión de varias solicitudes de cambio en el proceso de análisis 38](#_Toc161682010)

[Detalles de la estimación con Scrum Poker 39](#_Toc161682011)

[1.2 Anexo 3.- Bibliografía y material utilizado 43](#_Toc161682012)

[Anexo 4.- Relatorio de documentos asociados a éste 43](#_Toc161682013)

# Información sobre la práctica a realizar

## Descripción de la práctica

En esta práctica, sexta de la asignatura Ingeniería de Software, realizaremos un enfoque en diferentes técnicas de ingeniería particularizadas en el proceso de Gestión de la Configuración.

La primera tarea que abordaremos consiste en la creación de un proceso genérico para el control de cambios, que servirá como cimiento para las futuras actividades dentro de la misma práctica, aprovechando una línea de base ya establecida.

Para el desarrollo de nuestro proceso de control de cambios, es esencial desglosar una serie de actividades necesarias que facilitarán su implementación de manera efectiva. Asimismo, proponemos la inclusión de plantillas en blanco cuando sea pertinente, las cuales actuarán como herramientas para dar seguimiento a cada etapa del proceso de manera adecuada.

Optamos por presentar los detalles de cada actividad de manera estructurada, dividiéndolos en secciones que incluyen una breve descripción, los involucrados en la actividad, las entradas requeridas, los productos de trabajo necesarios y su ubicación correspondiente, así como los criterios de entrada y salida.

En lo que respecta a las plantillas desarrolladas, proporcionaremos una explicación detallada sobre su utilidad y justificaremos cada campo incluido para asegurar una comprensión clara de su propósito.

## Descripción del grupo de trabajo

|  |  |
| --- | --- |
| **Integrante** | **Rol** |
| **Nicolás Barcia Quintela** | Gestor del cambio |
| **Jesús Campos Manjón** | Gestor del cambio |
| **Guillermo Madriñán González** | Gestor del cambio |
| **Roi Martínez Enríquez** | Jefe de proyecto, Gestor del cambio |

## Seguimiento de la práctica

El seguimiento detallado de las tareas y tiempo dedicado por cada alumno en esta práctica se encuentra adjunto en un proyecto de *ProjectLibre*.

# DOCUMENTACIÓN DE LA PRÁCTICA

## Descripción del proceso de control de cambios

La gestión de cambios constituye un procedimiento esencial para administrar las solicitudes de modificación en proyectos y otras iniciativas críticas. Integrada como parte fundamental de un plan de gestión de cambios, esta práctica establece roles específicos encargados de supervisar y gestionar las modificaciones dentro de un equipo o empresa.

El proceso detallado en este documento abarca diversas actividades específicas diseñadas para supervisar detalladamente cada fase relacionada con la implementación de un cambio, independientemente de si este finalmente se lleva a cabo o no, en el contexto de un proyecto. Hemos identificado nueve actividades clave en nuestro proceso de control de cambios.

A continuación, en el desarrollo de cada actividad, exploraremos su propósito, los roles involucrados, las entradas necesarias, los criterios de entrada y salida, los productos de trabajo generados, así como cualquier subactividad relevante asociada. Este análisis detallado nos permitirá comprender a fondo el flujo y la estructura de nuestro proceso de gestión de cambios.

## Descripción del proceso de control de versiones con Git

Con el objetivo de reforzar la gestión de versiones en el proyecto, se realizará una reestructuración del proceso de control de cambios, asumiendo la adopción de Git en la definición de actividades. Esta revisión implica la incorporación de Git como herramienta central para gestionar las versiones de los archivos del proyecto. La redefinición de las actividades comprenderá la descripción detallada del repositorio Git, especificando quién tiene la autoridad para realizar cambios, cuándo y cómo se llevan a cabo, así como la implementación de etiquetado de estados del proyecto. Esta adaptación busca aprovechar las ventajas de Git para lograr un control más eficiente y detallado de las versiones, mejorando la trazabilidad y la calidad en el desarrollo del proyecto.

### Descripción del repositorio

El repositorio de Git se erige como el punto central para almacenar y rastrear la evolución de los archivos vinculados al proceso de gestión de control de cambios. Su estructura abarca una disposición cuidadosa de carpetas, no solo destinada a albergar los archivos inherentes al proceso, sino también a gestionar metadatos esenciales que posibiliten un control preciso de las versiones y cambios realizados a lo largo del tiempo. Dentro del repositorio, la estructura se ajustará de manera específica para dar soporte a las nueve actividades del proceso, ajustando la disposición de carpetas de manera en que refleje la secuencia de estas actividades:

* *01\_Identificacion\_Solicitud\_Cambio*: Una carpeta dedicada para almacenar documentos relacionados con la identificación del problema y las solicitudes de cambio, facilitando la rápida recuperación de información inicial.
* *02\_Evaluacion\_Desarrollador*: Una sección destinada a contener archivos relacionados con la evaluación realizada por los desarrolladores, incluyendo documentación y registros pertinentes.
* *03\_Evaluacion\_CCB*: Una carpeta específica para albergar documentos vinculados a la evaluación llevada a cabo por el CCB, proporcionando un espacio centralizado para estos registros.
* *04\_Notificacion\_Rechazo*: Una sección designada para almacenar documentos relacionados con notificaciones de rechazo, asegurando un acceso fácil y ordenado a esta información.
* *05\_Planificacion*: Una carpeta dedicada para la planificación de cambios, con subcarpetas que contienen documentos detallados sobre la planificación de cada cambio propuesto.
* *06\_Implementacion*: Una sección específica para albergar archivos relacionados con la implementación de cambios, asegurando una gestión clara de los recursos utilizados durante este proceso.
* *07\_Pruebas*: Una carpeta destinada a documentos relacionados con las pruebas realizadas durante el proceso de gestión de cambios, facilitando la revisión y evaluación de los resultados.
* *08\_Revision*: Una sección dedicada para archivos relacionados con las revisiones realizadas durante el proceso, incluyendo informes y documentación asociada.
* *09\_Cierre*: Una carpeta específica para documentación y registros relacionados con la fase de cierre, proporcionando un lugar centralizado para la conclusión y seguimiento de cambios implementados.

Dentro del repositorio, cada carpeta mencionada previamente estará enriquecida con una subcarpeta específica denominada 'Plantillas', que actuará como un repositorio centralizado para albergar plantillas en blanco destinadas a los productos de trabajo generados durante cada actividad del proceso de gestión de cambios.

Cabe destacar que en cada carpeta se incorporará un documento de identificación de los involucrados en la respectiva actividad. Este documento asignará un identificador único a cada usuario con la capacidad de modificar la documentación almacenada en el repositorio Git. Dichos identificadores serán utilizados por los usuarios al realizar commits, garantizando la rastreabilidad y trazabilidad de los responsables de los cambios. De esta manera, la inclusión de identificadores en los mensajes de commit permitirá una gestión efectiva de la autoría de los cambios, facilitando la comprensión y seguimiento de las contribuciones de cada usuario en el proceso de gestión de cambios.

En las descripciones detalladas de cada actividad a lo largo del presente documento, se proporcionará información específica sobre el proceso de cambio en el repositorio, abordando quién tiene la autoridad para realizar cambios, cuándo se pueden llevar a cabo y cómo se ejecutan. Asimismo, se detallará el procedimiento para etiquetar un estado del proyecto, indicando quién tiene la responsabilidad de realizar esta acción, cuándo se lleva a cabo y cómo se implementa.

## Descripción de uso de Póker Scrum

Para abordar un ajuste necesario de manera colectiva, optaremos por el método Póker Scrum para evaluar tanto el esfuerzo requerido como la complejidad de este. El funcionamiento del proceso es el siguiente:

Cada participante recibe un conjunto de cartas numeradas que representan diferentes niveles de esfuerzo, complejidad o tiempo requerido, para indicar la magnitud del esfuerzo. También se incluyen algunas cartas especiales, como puede ser una interrogación para indicar desconocimiento.

Después, se presenta una historia de usuario al equipo, explicando cual es el proceso que realizar sin detallar demasiado. Previo a comenzar con las votaciones, los integrantes del equipo plantean preguntas sobre la historia. El objetivo es asegurarse de que todos entiendan la tarea de la misma manera.

Después de la ronda de preguntas, cada miembro del equipo elige una de sus cartas para indicar el esfuerzo necesario que considera que llevará la propuesta. Las cartas de todos son reveladas de manera simultánea para evitar que las elecciones de unos influyan sobre las de otros.

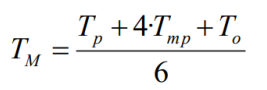
Tras votar, los miembros del equipo con las estimaciones más altas y bajas explican su razonamiento. El objetivo de esto es llegar a un punto común y evitar que haya miembros con percepciones incorrectas que influyan en el resultado final.

Finalmente, se hace una segunda ronda de votación. Esta repetición de la votación junto con el debate previo podría repetirse de forma indefinida hasta conseguir un punto lo más consensuado posible. Se considera que hay convergencia cuando todas las votaciones se corresponden con 3 cartas consecutivas.

Como tiempo final se propone el tiempo *Pert*.

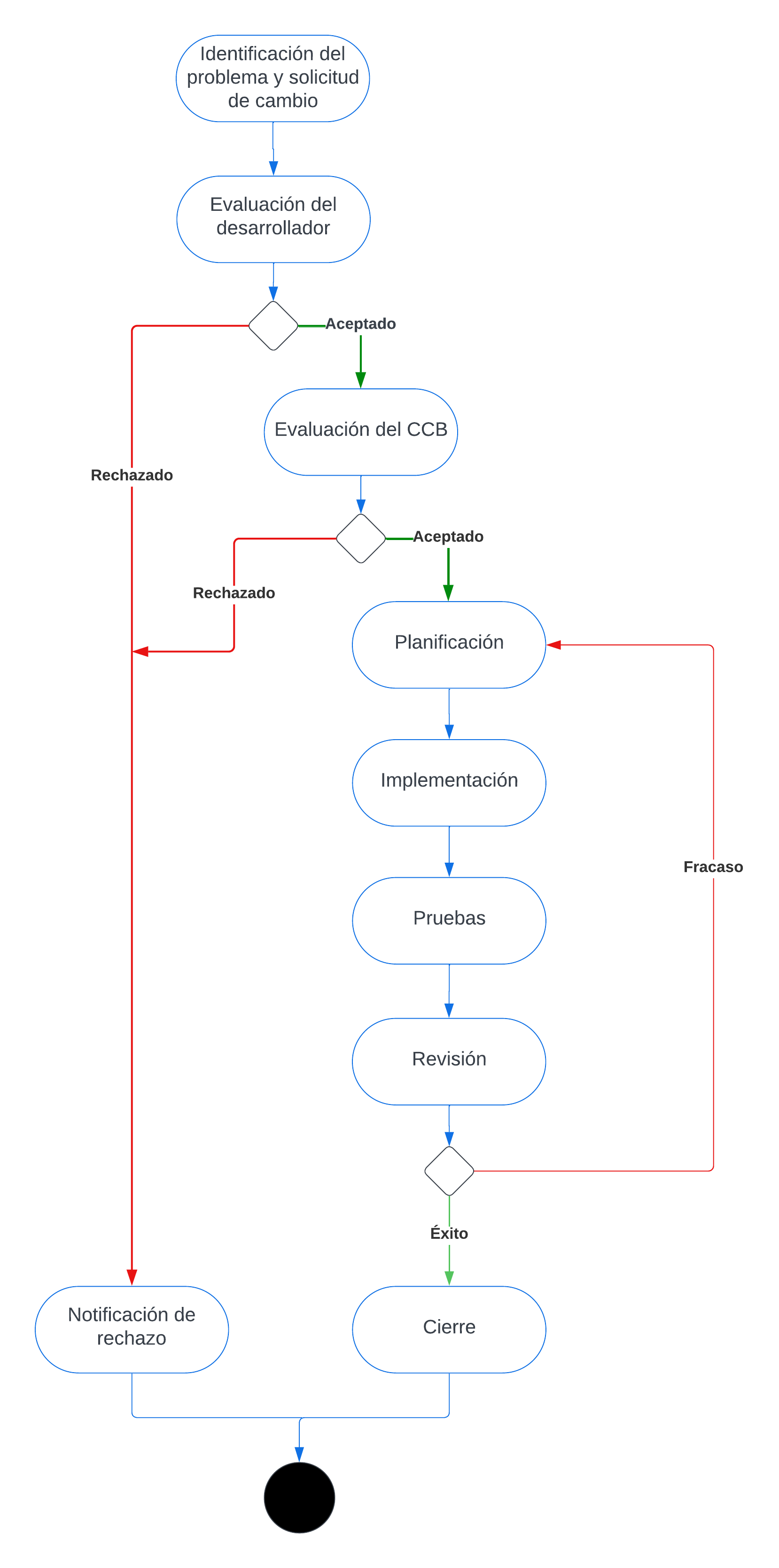
* Tiempo pesimista (*Tp*): Tiempo propuesto más largo
* Tiempo optimista (*To*): Tiempo propuesto más corto
* Tiempo más probable (*Tmp*): Tiempo que más veces se repite

Se aplica la fórmula:



El *PokerScrum* fomenta el consenso, la colaboración, el debate y permite alcanzar estimaciones de esfuerzo más realistas.

## Diagrama de actividades



## Definición de Actividades

### Identificación del problema y solicitud de cambio

Descripción y propósito

Esta actividad tiene como objetivo recopilar información sobre problemas identificados y formalizar solicitudes de cambio, siendo la identificación del problema y la solicitud de cambio un proceso integrado. Durante este proceso, los interesados en el proyecto, como usuarios finales y personal de atención al cliente, describen cualquier problema encontrado o sugieren cambios en el software. La actividad se inicia con la identificación del problema, y la solicitud de cambio se genera implícitamente en este proceso.

Involucrados

Todos los interesados en el proyecto, excluido el equipo de desarrollo. Incluye al personal del servicio de atención al cliente, usuarios finales y cualquier persona que identifique problemas o proponga cambios.

Entradas requeridas

* Observaciones de usuarios, informes de errores, comentarios, análisis de rendimiento u otros documentos que describan problemas o cambios potenciales.
* Plantilla "Identificación del Problema" en blanco disponible en la ruta *01\_Identificacion\_Solicitud/Plantillas/IdfSlc.xlsx* del repositorio Git.

Productos de trabajo creados

Documento combinado que surge de completar la plantilla “Identificación del problema” y que incluye la descripción del problema identificado y la formalización de la solicitud de cambio (RFC).

Se almacenará como hoja exportada de Excel en formato *pdf* en la ruta *01\_Identificacion\_Solicitud/Solicitudes* del repositorio Git, nombrándolo como *[Num]*.pdf siendo *Num* el número de identificación correlativo indicado en la plantilla.

El personal de servicio de atención al cliente será responsable de completar la plantilla y de subir el documento al repositorio Git al concluir las subactividades correspondientes. Estos cambios se registrarán mediante commits en Git, identificando al personal de atención al cliente según el documento de identificación en la carpeta y proporcionando un mensaje breve que describa la acción realizada.

Criterios de entrada

Recepción de una llamada, FAX o correo electrónico que aún no ha sido procesado y requiere atención.

Criterios de salida

La tarea se considera completada cuando se ha generado un documento combinado que incluye la descripción del problema identificado y la solicitud de cambio y ha sido subido al repositorio Git. Este documento sirve como punto de partida para la revisión técnica y la toma de decisiones posteriores.

Subactividades

#### Revisar identificación

Antes de registrar la solicitud, se realiza una revisión preliminar de la identificación del problema para asegurar su integridad y completitud. En caso de identificar información faltante o insuficiente, se solicita al remitente que proporcione la información necesaria para completar la solicitud y garantizar una evaluación completa y precisa.

#### Registrar la solicitud en el sistema

La solicitud de cambio se registra en un sistema de seguimiento para facilitar el monitoreo del estado y la asignación a los evaluadores. Además, una integración con el repositorio Git asegura que cada solicitud sea rastreable y gestionada eficientemente a lo largo del proceso de control de cambio.

### Evaluación del desarrollador

Descripción y propósito

Esta actividad tiene como objetivo evaluar la viabilidad técnica de la solicitud de cambio, así como estimar el esfuerzo y los recursos necesarios para su implementación. El desarrollador asignado revisa la solicitud y proporciona una evaluación preliminar.

Involucrados

Desarrolladores asignados.

Entradas requeridas

* Documento formal de solicitud de cambio en estado pendiente proveniente de la actividad anterior.
* Plantilla "Evaluación del desarrollador" en blanco disponible en la ruta *02\_Evaluacion\_Desarrollador/Plantillas/EvalDes.xlsx* del repositorio Git.

Productos de trabajo creados

Documento sobre la evaluación realizada por el desarrollador, que surge de completar la plantilla “Evaluación del desarrollador” e incluye la viabilidad técnica, estimación del esfuerzo y recursos necesarios.

Se almacenará como hoja exportada de Excel en formato *pdf* en la carpeta *Aceptadas* o *Rechazadas,* según la decisión tomada, en la ruta *02\_Evaluacion\_Desarrollador*, nombrándolo como *[Num]*.pdf siendo *Num* el número de identificación correlativo indicado en la plantilla.

El desarrollador responsable de la evaluación será encargado de completar la plantilla y cargar el documento al repositorio Git al concluir las subactividades correspondientes. Los cambios se registrarán mediante commits en Git, identificando al desarrollador según el documento de identificación de la carpeta y proporcionando un mensaje breve que describa la acción realizada.

Criterios de entrada

La actividad puede comenzar cuando se recibe la solicitud formal de cambio.

Criterios de salida

La tarea se considera completada cuando se ha realizado una evaluación técnica de la solicitud de cambio, que incluye la viabilidad técnica y una estimación del esfuerzo y recursos requeridos, la plantilla ha sido cubierta con éxito y subida al repositorio Git. Además, la etiqueta de la solicitud de cambio implicada debe modificarse a “En proceso”.

Subactividades

#### Revisión técnica

El desarrollador revisa la solicitud de cambio para evaluar su factibilidad técnica, considerando la arquitectura existente, las dependencias y la complejidad técnica.

#### Estimación de esfuerzo

El desarrollador proporciona una estimación del esfuerzo necesario para implementar el cambio. Esto incluye el tiempo estimado y cualquier recurso adicional necesario.

#### Identificación de riesgos

Se identifican posibles riesgos técnicos asociados con la implementación del cambio. Estos riesgos se documentan para su consideración en etapas posteriores del proceso.

### Evaluación del CCB

Descripción y propósito

El objetivo fundamental de esta actividad es revisar las solicitudes de cambio que surgieron durante la identificación del problema y fueron aceptadas por uno de los desarrolladores evaluadores. El CCB (*Change Control Board*), conformado por expertos con amplia experiencia y conocimiento integral del proyecto, lleva a cabo un análisis detallado para tomar decisiones informadas sobre la aprobación o rechazo de cada solicitud. Es crucial destacar que la toma de decisiones del CCB se basa en las anotaciones y evaluaciones realizadas por los desarrolladores durante la fase previa. Estas evaluaciones actúan como un vínculo crucial entre el problema identificado por los usuarios y la información presentada al CCB, garantizando que las decisiones estén respaldadas por una comprensión profunda de la viabilidad técnica y los recursos necesarios.

Involucrados

Miembros del *CCB*, que pueden incluir líderes de proyecto, desarrolladores senior y representantes de diferentes áreas del proyecto.

Entradas requeridas

* Documento combinado que incluye la descripción del problema identificado y la solicitud de cambio proveniente de la actividad inicial, etiquetado “En proceso”.
* Documento resultante de la evaluación previa del desarrollador, disponible en la ruta *02\_Evaluacion\_Desarrollador/Aceptadas* del repositorio Git.
* Plantilla "Evaluación del CCB" en blanco disponible en la ruta *03\_Evaluacion\_CCB/Plantillas/EvalCCB.xlsx* del repositorio Git.

Productos de trabajo creados

Documento de la reunión del CCB sobre cada cambio propuesto, que reflejará la decisión del comité, el nivel de prioridad asignado y otros datos relevantes surgidos durante la evaluación.

Se almacenará como hoja exportada de Excel en formato *pdf* en la carpeta *Aceptadas* o *Rechazadas,* según la decisión tomada, de la ruta *03\_Evaluacion\_CCB* del repositorio*,* nombrándolo como *[Num]*.pdf siendo *Num* el número de identificación correlativo indicado en la plantilla.

La responsabilidad de completar la plantilla y cargar el documento al repositorio Git recae en el jefe del comité de control de cambios al concluir las subactividades pertinentes. Los cambios se registrarán mediante commits en Git, identificando al jefe según el documento de identificación de la carpeta y proporcionando un mensaje breve que describa la acción realizada.

Criterios de entrada

La actividad comienza con la programación de una reunión del CCB, la cual se llevará a cabo cuando se reciban las solicitudes de cambio evaluadas por los desarrolladores. Durante el proceso de evaluación, se asignará un valor de "urgencia" a cada solicitud en una escala definida. Por ejemplo, se utilizarán los valores de urgencia 1, 3, 5 y 10. Se programará la reunión del CCB cuando el valor total acumulado de urgencia entre todas las solicitudes alcance el umbral predeterminado de 10. Este enfoque asegura que se realice una revisión oportuna por parte del CCB, adaptándose a situaciones donde puede haber una única solicitud con una urgencia significativa que requiere evaluación inmediata.

Criterios de salida

La tarea se considera completada cuando el CCB ha tomado decisiones informadas sobre la aprobación o rechazo de cada solicitud de cambio planteada en la reunión, y completa correctamente la plantilla “Evaluación del CCB”, que es subida al repositorio Git.

Subactividades

#### Revisión del documento de entrada

Durante esta fase, los miembros del CCB llevan a cabo una revisión meticulosa del documento combinado, enfocándose en la coherencia de cada solicitud con los objetivos del proyecto.

#### Análisis de prioridades y riesgos

La priorización de solicitudes se realiza considerando su contribución a los objetivos del proyecto, el nivel de riesgo asociado y la urgencia. Se lleva a cabo un análisis detenido del impacto potencial en los objetivos del proyecto y se identifican los riesgos vinculados a la implementación de cada cambio, garantizando una toma de decisiones fundamentada.

#### Toma y comunicación de decisiones

Se evalúa exhaustivamente el beneficio de cada solicitud para el proyecto, garantizando alineación con la estrategia a largo plazo y considerando la viabilidad técnica y disponibilidad de recursos. La toma de decisiones se comunica de manera transparente, proporcionando información clara sobre las razones de aprobación o rechazo. Se establece un canal efectivo de retroalimentación para comunicar a los solicitantes las razones detrás de las decisiones, fomentando comprensión y aceptación.

### Notificación de rechazo

Descripción y propósito

Esta actividad se enfoca en notificar al cliente sobre el rechazo de la solicitud de cambio propuesta. Su propósito principal es mantener una comunicación transparente y efectiva con el cliente, proporcionando una explicación clara y fundamentada sobre la decisión de no implementar el cambio en el proyecto.

Involucrados

Miembros del equipo de comunicación con el cliente, además del cliente que identificó y propuso el cambio, posteriormente rechazado.

Entradas requeridas

* Documento combinado que incluye la descripción del problema identificado y la solicitud de cambio proveniente de la actividad inicial, etiquetado “En proceso”.
* Documento resultante de la evaluación previa del desarrollador, disponible en la ruta *02\_Evaluacion\_Desarrollador/Rechazadas* del repositorio Git.
* Además, si el rechazo es por parte del CCB, el documento resultante de la evaluación de este equipo, disponible en la ruta *03\_Evaluacion\_CCB/ Rechazadas* del repositorio Git.
* Plantilla "Notificación" en blanco disponible en la ruta *04\_Notificacion\_Rechazo/Plantillas/Rechazo.xlsx* del repositorio Git.

Productos de trabajo

El producto principal de esta actividad es la notificación formal de rechazo al cliente, detallando claramente las razones del rechazo de la solicitud de cambio.

Se almacenará como hoja exportada de Excel en formato *pdf* en la ruta *04\_Notificacion\_Rechazo* del repositorio Git, nombrándolo como *[Num]*.pdf siendo *Num* el número de identificación correlativo indicado en la plantilla y etiquetando como “CCB” o “Desarrollador” según quién hubiese tomado la decisión.

La responsabilidad de completar la plantilla y cargar el documento al repositorio Git recae en el miembro del equipo de comunicación con el cliente encargado, al concluir las subactividades pertinentes. Los cambios se registrarán mediante commits en Git, identificando al responsable según el documento de identificación de la carpeta y proporcionando un mensaje breve que describa la acción realizada.

Criterios de entrada

La actividad inicia cuando se ha completado la evaluación del desarrollador o la revisión del CCB, resultando en la decisión de rechazar la solicitud de cambio propuesta, y por tanto se genera un nuevo documento en una de las carpetas *Rechazadas* de las carpetas del repositorio asociadas a ambas evaluaciones. En este punto, se deben haber identificado claramente las razones para el rechazo, proporcionando la base necesaria para comunicar de manera efectiva esta decisión al cliente.

Criterios de salida

La actividad se considera completada una vez que se ha completado la plantilla, subido al repositorio y enviado al cliente la notificación formal que detalla las razones del rechazo de la solicitud de cambio. La comunicación debe ser clara y comprensible, brindando al cliente información detallada sobre las razones detrás de la decisión. La notificación deberá ser archivada adecuadamente para mantener un registro transparente del proceso de gestión de cambios. Además, la etiqueta asociada a la solicitud de cambio original debe cambiarse de “En proceso” a “Finalizado”, y será el miembro de comunicación quien se encargue de realizarlo.

Subactividades

#### Preparación de la notificación

El responsable de comunicación con el cliente recopilará información detallada sobre las razones del rechazo, indicadas en los productos de trabajo de las actividades previas, y redactará un mensaje claro y conciso para que el cliente comprenda las razones de la decisión.

#### Envío de la notificación

El responsable comunicará de manera efectiva y oportuna al cliente, estableciendo canales para discusiones o aclaraciones si es necesario.

### Planificación

Descripción y propósito

La actividad de planificación tiene como objetivo principal definir de manera detallada todas las fases y aspectos relacionados con la implementación del cambio propuesto. Esto implica la creación de un plan que aborde aspectos técnicos, temporales y de recursos, garantizando una ejecución eficiente y controlada. Durante esta fase, se identifican y resuelven posibles problemas y se establecen las bases para el éxito del cambio, brindando una visión clara y completa de cómo se llevará a cabo la transición.

Involucrados  
Gerente del proyecto de cambio y equipo de desarrollo.

Entradas requeridas

* Documento de la reunión del CCB en el que se refleje la aprobación de los cambios, proveniente de la actividad anterior, disponible en la ruta *03\_Evaluacion\_CCB/Aceptadas* del repositorio Git. O el documento procedente de la actividad de Revisión con resultados no satisfactorios, disponible en la ruta *08\_Revisión/Rechazadas*.
* Documentación de recursos disponibles y posibles restricciones de tiempo.
* Plantilla "Planificación" en blanco disponible en la ruta *05\_Planificación/Plantillas/Plan.xlsx* del repositorio Git.

*[Se asume que la trazabilidad de las plantillas permite acceder a información en actividades previas, en caso de ser necesario.]*

Productos de trabajo

Plan detallado de implementación del cambio, incluyendo cronograma, asignación de recursos, lista de tareas y actividades, identificación de riesgos y medidas de mitigación.

Se almacenará como hoja exportada de Excel en formato *pdf* en la carpeta *05\_Planificación* del repositorio*,* nombrándolo como *[Num]*.pdf siendo *Num* el número de identificación correlativo indicado en la plantilla.

La responsabilidad de completar la plantilla y cargar el documento al repositorio Git recae en el jefe del comité de control de cambios al concluir las subactividades pertinentes. Los cambios se registrarán mediante commits en Git, identificando al jefe según el documento de identificación de la carpeta y proporcionando un mensaje breve que describa la acción realizada.

Criterios de entrada

La actividad de planificación comienza después de que la CCB revise y apruebe los cambios propuestos, asegurándose de contar con todos los detalles técnicos, informes de impacto, y recursos necesarios para proceder con la planificación detallada. La aceptación de la CCB genera un nuevo documento en la carpeta *Aceptadas,* dentro de carpeta del repositorio asociada a la evaluación del CCB, “*03\_Evaluacion\_CCB”.*

Criterios de salida

Plan detallado de implementación del cambio completo y listo para su ejecución. Todos los elementos del plan, incluyendo tareas, plazos y asignación de recursos, deben estar claramente definidos. Por último, se completa correctamente la plantilla “Planificación”, que es subida al repositorio Git.

Subactividades

#### Votación Póker Scrum

En esta fase se lleva a cabo una reunión de planificación para el cambio en el proyecto. Durante esta reunión, el gerente del proyecto y el equipo de desarrollo emplea la técnica de Póker Scrum para hacer una estimación temporal de la duración del proyecto. Se utiliza la baraja de Fibonacci con unidades de 2,5 semanas laborales. Después de desglosar las tareas y discutir el cambio propuesto, cada miembro del equipo, incluido el gerente, estima individualmente el esfuerzo necesario para completar cada tarea. Se relevan simultáneamente sus cartas, facilitando la discusión en caso de discrepancias. Se presta especial atención al esfuerzo de modificación de documentos existentes y a cualquier carga adicional al proyecto. Si es necesario, se repite la votación discutiendo las posiciones extremas hasta llegar a un punto de convergencia, donde se registran las estimaciones finales y se calcula el coste total en las unidades de tiempo acordadas. El gerente del proyecto utiliza estas estimaciones para planificar y priorizar las tareas en el cronograma, mientras que el equipo las considerará durante la ejecución, facilitando un seguimiento efectivo y actualizaciones según sea necesario.

La razón de implementar la actividad de Póker Scrum en la fase de planificación y no en otras previas, es que hasta esta fase no hay garantía de que el cambio se vaya a llevar a cabo. Además, en la fase de planificación, surge la necesidad de hacer una correcta estimación temporal del cambio, para así poder hacer una correcta implementación de este.

#### Definir tareas y actividades

Desglosar el cambio en tareas y actividades específicas para una comprensión más detallada. Además, se ha de determinar la secuencia de las tareas y establecer plazos realistas, identificando dependencias entre ellas.

#### Asignar recursos necesarios para la implementación

Esta subactividad implica un análisis exhaustivo de los recursos esenciales para llevar a cabo la implementación del cambio propuesto. En primer lugar, se identificarán detalladamente los recursos requeridos, abarcando desde las habilidades específicas del personal hasta las herramientas y equipos necesarios. A continuación, se asignarán roles específicos dentro del equipo de desarrollo, considerando las capacidades individuales de cada miembro. Además, se llevará a cabo una evaluación minuciosa de la disponibilidad y eficacia de las herramientas y equipos necesarios, asegurando que estén alineados con los requisitos del cambio planificado.

#### Identificar y planificar medidas de mitigación para posibles riesgos

Esta subactividad se centra en la identificación y evaluación de riesgos potenciales que podrían surgir durante la implementación del cambio, a mayores de aquellos ya identificados inicialmente en la evaluación del desarrollador. Se realizará un análisis detallado de la probabilidad e impacto de cada riesgo identificado. Posteriormente, se priorizarán estos riesgos según su gravedad, destacando aquellos que podrían tener un impacto significativo en el éxito del cambio. La fase final implica el desarrollo de estrategias de mitigación específicas, anticipándose a posibles desafíos y estableciendo medidas preventivas para garantizar una implementación exitosa.

#### Revisar y obtener aprobación para el plan de implementación propuesto

En esta subactividad, se preparará una presentación detallada del plan de implementación, destacando sus componentes clave de manera clara y comprensible. Se llevará a cabo una sesión de revisión con todos los involucrados, proporcionando explicaciones detalladas sobre cada aspecto del plan y respondiendo a preguntas. Cualquier comentario recibido durante esta fase será considerado y, si es necesario, se realizarán ajustes en el plan. La meta final es obtener la aprobación unánime de los participantes antes de avanzar a la siguiente fase del proceso de control de cambio.

### Implementación

Descripción y propósito

La actividad de implementación se centra en la ejecución efectiva del plan detallado definido durante la fase de planificación. El propósito principal es llevar a cabo el cambio propuesto de manera ordenada y controlada, asegurando la transición suave del estado actual al estado deseado. Durante esta fase, se implementan las tareas y actividades planificadas, se asignan los recursos según necesarios y se aplican medidas para mitigar posibles riesgos identificados. Estas actividades deberían de ajustarse a la planificación, pero pueden diferir de esta, por eso es necesario tomar registro de como finalmente fueron implementadas.

Involucrados  
El equipo de desarrollo y el gerente de proyecto.

Entradas requeridas

* Documento con el plan detallado de implementación del cambio generado en la actividad anterior, disponible en la carpeta *05\_Planificación* del repositorio de Git.
* Plantilla "Implementación" en blanco disponible en la ruta *06\_Implementación/Plantillas/Implementación.xlsx* del repositorio Git.

*[Se asume que la trazabilidad de las plantillas permite acceder a información en actividades previas, en caso de ser necesario. Se entiende de forma implícita que la versión anterior del software se encuentra disponible para realizar la implementación de los cambios.]*

Productos de trabajo

La salida de esta actividad será el resultado de la implementación, incluyendo la aplicación efectiva de las tareas y actividades planificadas, la asignación final de recursos, y la ejecución de medidas de mitigación para gestionar los riesgos identificados.

Se almacenará como hoja exportada de Excel en formato *pdf* en la carpeta *06\_Implementación* del repositorio*,* nombrándolo como *[Num]*.pdf siendo *Num* el número de identificación correlativo indicado en la plantilla.

La responsabilidad de completar la plantilla y cargar el documento al repositorio Git recae en el jefe del comité de control de cambios al concluir las subactividades pertinentes. Los cambios se registrarán mediante commits en Git, identificando al jefe según el documento de identificación de la carpeta y proporcionando un mensaje breve que describa la acción realizada.

Criterios de entrada

La implementación comienza en la fecha indicada en el plan detallado de implementación realizado previamente, donde se garantiza la disponibilidad de todos los documentos y recursos necesarios.

Criterios de salida

La salida exitosa de la implementación se verifica mediante la confirmación de que todas las tareas planificadas han sido ejecutadas. Para finalizar exitosamente esta actividad, se completa correctamente la plantilla “Implementación”, que es subida al repositorio Git.

Subactividades

#### Ejecución y monitoreo de tareas

En esta subactividad, se lleva a cabo la ejecución meticulosa de las tareas planificadas, siguiendo el detallado plan de implementación. El enfoque principal reside en el monitoreo continuo, permitiendo la identificación inmediata de cualquier desviación respecto al plan establecido. Se emplea una vigilancia constante para evaluar el progreso, la calidad y la eficiencia de la ejecución. La capacidad de realizar ajustes instantáneos asegura una adaptación ágil a las circunstancias cambiantes, garantizando así la alineación continua con los objetivos de implementación.

#### Comunicación efectiva del progreso

En esta subactividad, se establece y sostiene una comunicación fluida y transparente con el equipo de desarrollo. El objetivo principal es asegurar que cada miembro del equipo tenga una comprensión clara del progreso general de la implementación. La retroalimentación continua y la resolución ágil de problemas contribuyen significativamente al éxito de la implementación.

### Pruebas

Descripción y propósito

La fase de pruebas se lleva a cabo después de la implementación para validar la efectividad del cambio propuesto y garantizar que cumpla con los requisitos y expectativas establecidos durante las fases anteriores del proceso de control de cambios. Durante esta fase, se realizan las pruebas pertinentes según los cambios introducidos, intentando abarcar todos los aspectos necesarios.

Involucrados

El equipo de pruebas, los desarrolladores y el gerente del proyecto.

Entradas requeridas

* Documento con los detalles de implementación generado en la actividad anterior, disponible en la carpeta *06\_Implementación* del repositorio de Git.
* Plantilla "Pruebas" en blanco disponible en la ruta *07\_Pruebas*/*Plantillas/Pruebas.xlsx* del repositorio Git.

*[Se asume que la trazabilidad de las plantillas permite acceder a información en actividades previas, en caso de ser necesario. Se entiende de forma implícita que la nueva versión del software se encuentra disponible para realizar las pruebas.]*

Productos de trabajo

El producto de esta actividad será el resultado de las pruebas, incluyendo tanto la aplicación de las pruebas junto a la documentación de sus resultados, y la detección de los posibles problemas que puedan encontrarse durante las mismas junto a sus respectivos reportes.

Se almacenará como hoja exportada de Excel en formato *pdf* en la carpeta *07\_Pruebas* del repositorio Git, nombrándolo como *[Num].pdf* siendo *Num* el número de identificación correlativo indicado en la plantilla.

El desarrollador responsable de las pruebas será encargado de completar esta plantilla y cargar el documento al repositorio Git al concluir las subactividades correspondientes. Los cambios se registrarán mediante commits en Git, identificando al desarrollador según el documento de identificación de la carpeta y proporcionando un nuevo mensaje breve que describa la acción realizada.

Criterios de entrada

La fase de pruebas comienza tras haber realizado la implementación y haber recibido un informe detallado de este proceso, así como el software resultante de esta etapa.

Criterios de salida

La salida exitosa de esta fase se verifica mediante la confirmación de que se han ejecutado pruebas sobre todas las implementaciones realizadas previamente, cumpliendo con los requisitos del proyecto. Además, en caso de que en los resultados de las pruebas se hayan detectado problemas, se debe confirmar también su identificación y reporte. Finalmente, se considera exitosa una vez se han subido al repositorio Git.

Subactividades

#### Pruebas funcionales y de rendimiento

Esta subactividad se centra en verificar la funcionalidad y rendimiento del sistema implementado. Las pruebas funcionales aseguran que todas las características y funcionalidades del sistema funcionen según lo especificado, mientras que las pruebas de rendimiento evalúan la capacidad del sistema para manejar la carga prevista.

#### Identificación y reporte de problemas

Durante el proceso de implementación, pueden surgir problemas o defectos que requieran ser abordados. Esta subactividad se centra en la identificación, documentación y reporte de cualquier problema encontrado. En caso de identificar problemas durante las pruebas, se documentan detalladamente para su corrección posterior.

### Revisión

Descripción y propósito

La fase de revisión se centra en evaluar críticamente la implementación realizada, revisando aspectos técnicos, arquitectónicos y de calidad del código para garantizar la conformidad con los estándares y requisitos previamente definidos.

Involucrados

Equipo de revisión técnica, desarrolladores y gerente del proyecto.

Entradas requeridas

* Documento con el plan de pruebas y reportes detallado generado en la actividad anterior, disponible en la carpeta *07\_Pruebas* del repositorio de Git.
* Plantilla "Revisión" en blanco disponible en la ruta *08\_Revisión/Plantillas/Revision.xlsx* del repositorio Git.

*[Se asume que la trazabilidad de las plantillas permite acceder a información en actividades previas, en caso de ser necesario. Se entiende de forma implícita que la nueva versión del software se encuentra disponible para realizar la revisión.]*

Productos de trabajo

La salida de esta actividad será el resultado de la revisión, incluyendo el estudio exhaustivo del código para asegurar su calidad y la implementación de las correcciones necesarias en caso de que se detectasen problemas durante la fase de pruebas o se considerase necesario tras la revisión del código.

Se almacenará como hoja exportada de Excel en formato *pdf* en la carpeta *Aceptadas* o *Rechazadas*, según la decisión tomada, en la ruta *08\_Revision*, nombrándolo como *[Num]*.pdf siendo *Num* el número de identificación correlativo indicado en la plantilla.

El desarrollador responsable de la revisión será el encargado de completar la plantilla y cargar el documento al repositorio Git al concluir las subactividades correspondientes. Los cambios se registrarán mediante commits en Git, identificando al desarrollador según el documento de identificación de la carpeta y proporcionando un mensaje breve que describa la acción realizada.

Criterios de entrada

La implementación comienza tras finalizar la ejecución de las pruebas y su documentación, junto al reporte de los posibles problemas encontrados.

Criterios de salida

La salida exitosa de la revisión se verifica mediante la confirmación de que se ha realizado una revisión exhaustiva del código y la realización de una evaluación sobre el mismo (realizando su informe correspondiente), que permite proceder al cierre del proceso de cambios en caso de que sea exitosa, o retornar a la etapa de planificación en caso negativo, así como la subida de la plantilla cubierta con éxito al repositorio Git.

Subactividades

#### Revisión técnica y auditoría

Se realiza una revisión detallada del código y la arquitectura para asegurar la calidad técnica y la conformidad con los estándares.

#### Evaluación general

En base a los resultados obtenidos de la revisión técnica y auditoría, así como de los informes proporcionados a través de las etapas anteriores, se decide aprobar el cierre del proyecto o denegar el mismo, teniendo que retornar a la etapa de planificación para poder solucionar los errores detectados.

### Cierre

Descripción y propósito

La fase de cierre representa la culminación del proyecto, donde se realizan las actividades finales para formalizar la conclusión de todas las tareas y objetivos. El propósito principal de esta etapa es revisar y documentar el resultado final del proyecto, evaluando su éxito.

Involucrados

Gerente del proyecto, el equipo de Desarrollo y los miembros del CCB y *stakeholders* relevantes si procede.

Entradas requeridas

* Documento con el informe de revisión generado en la actividad anterior, disponible en la ruta *08\_Revisión/Aceptadas* del repositorio Git.
* Plantilla "Cierre" en blanco disponible en la ruta *09\_Cierre*/*Plantillas/Cierre.xlsx* del repositorio Git.

*[Se asume que la trazabilidad de las plantillas permite acceder a información en actividades previas, en caso de ser necesario.]*

Productos de trabajo creados

La salida de esta actividad será el resultado del cierre del proceso de control de un cambio, documentando información global del trabajo realizado a lo largo de las etapas previas.

Se almacenará como hoja exportada de Excel en formato *pdf* en la carpeta *09\_Cierre* del repositorio Git, nombrándolo como *[Num]*.pdf siendo *Num* el número de identificación correlativo indicado en la plantilla.

La responsabilidad de completar la plantilla y cargar el documento al repositorio Git recae en el gerente del proyecto, al concluir las subactividades pertinentes. Los cambios se registrarán mediante commits en Git, identificando al responsable según el documento de identificación de la carpeta y proporcionando un mensaje breve que describa la acción realizada.

Criterios de entrada

El cierre comienza tras la creación de un informe de revisión del cambio, donde se garantiza la calidad de este y se indica que el proyecto de cambio está listo para entrar en su última etapa.

Criterios de salida

La salida exitosa del cierre se verifica mediante la evaluación y creación de un informe de cierre, que describe de forma general el proceso de cambios; y su posterior revisión y aprobación, que permite dar por finalizado este proceso. Además, la etiqueta asociada a la solicitud de cambio original debe cambiarse de “En Proceso” a “Finalizado”, y será el encargado del proyecto quien se encargue de realizarlo.

Subactividades

#### Creación del documento de cierre

Esta subactividad tiene como objetivo la elaboración de un documento de cierre integral que documente la finalización exitosa del proyecto, destacando los logros, el alcance alcanzado y la conformidad con los requisitos establecidos.

#### Revisión y Aprobación del CCB

En esta subactividad, se somete el proyecto a una revisión final por parte del Change Control Board (CCB). El objetivo es asegurar que todas las actividades se hayan completado según lo planeado y que el proyecto esté listo para el cierre.

## Plantillas del proceso

### Plantilla de Identificación del problema y Solicitud de cambio

1. **Fecha**: dd/mm/aa (Automática)
2. **Identificador**: IdfSlc + Código correlativo (Automático)
3. **Datos del solicitante**:
   1. *Nombre*: [Nombre completo del solicitante]
   2. *Correo*: [Correo electrónico del solicitante]
4. **Descripción del problema**:
   1. *Situación en la que se detecta*: [Describa el contexto o situación en la que se identificó el problema o se propuso el cambio]
   2. *Descripción detallada*: [Proporcione una descripción detallada del problema identificado, incluyendo cualquier aspecto técnico relevante]
5. **Solicitud de cambio**:
   1. *Descripción del cambio propuesto*: [Proporcione detalles sobre la naturaleza del cambio propuesto]
   2. *Justificación para el cambio*: [Explique la razón detrás de la necesidad de este cambio. ¿Cuál es el impacto si no se implementa?]

#### Modelo



#### Justificación de sus campos

1. **Fecha**: El registro de la fecha de identificación del problema y solicitud de cambio es esencial para evaluar el tiempo transcurrido hasta su resolución. Permite un seguimiento temporal efectivo del proceso.
2. **Identificador**: Asignar un identificador único a cada solicitud evita pérdidas o duplicaciones. Facilita el seguimiento, organización y referencia rápida de cada solicitud.
3. **Datos del solicitante**: Capturar el nombre del solicitante facilita la identificación del origen del problema o la propuesta de cambio. La inclusión del correo electrónico permite una comunicación efectiva y la notificación sobre el progreso de la solicitud.
4. **Descripción del problema**: Proporcionar campos específicos para la situación y descripción asegura una documentación completa y clara del problema identificado. Facilita la comprensión y análisis posterior.
5. **Solicitud de cambio**: Incluir una sección específica para describir el cambio propuesto, junto a su justificación, proporciona información crucial para evaluar la necesidad y el impacto del cambio propuesto.

### Plantilla de Evaluación del desarrollador

1. **Fecha**: dd/mm/aa (Automática)
2. **Identificador**: EvalDes + Código correlativo (Automático)
3. **Identificador del documento base**: [ID de la plantilla Identificación del problema y solicitud de cambio que inicia la evaluación]
4. **Desarrollador asignado**:
   1. *Nombre:* [Nombre completo del desarrollador]
   2. *Correo:* [Correo electrónico del desarrollador]
   3. *ID de empleado:* [Número de empleado asignado al desarrollador]
5. **Evaluación**:
   1. *Aprobación/Rechazo*: [Decisión del CCB sobre la solicitud de cambio]
   2. *Factibilidad técnica:* [Evaluar la factibilidad técnica de la solicitud de cambio, considerando la arquitectura existente, las dependencias y la complejidad técnica]
   3. *Tiempo estimado:* [Proporcionar una estimación del tiempo necesario para implementar el cambio]
   4. *Recursos adicionales:* [Indicar cualquier recurso adicional necesario para la implementación]
   5. *Identificación de riesgos:* [Listar posibles riesgos técnicos asociados con la implementación del cambio y documentarlos para su consideración en etapas posteriores del proceso]

#### Modelo



#### Justificación de sus campos

1. **Fecha**: El registro de la fecha de evaluación del desarrollador permite un seguimiento temporal efectivo del proceso y facilita la organización de las evaluaciones.
2. **Identificador**: Asignar un identificador único a cada evaluación evita pérdidas o duplicaciones, facilitando el seguimiento, organización y referencia rápida de cada evaluación.
3. **Identificador del documento base:** Incluir el identificador del documento base asociado proporciona un vínculo claro entre la evaluación y la solicitud original, facilitando la trazabilidad y referencia.
4. **Datos del desarrollador**: La captura del nombre completo del desarrollador, junto con su correo electrónico y número de empleado, proporciona información detallada y única sobre la persona responsable de la evaluación. Esta información facilita la comunicación, notificación sobre el progreso y un seguimiento eficiente del personal involucrado en el proceso.
5. **Evaluación:** Evaluar la factibilidad técnica de la solicitud de cambio es esencial para determinar su viabilidad y posibles desafíos técnicos. Además, proporcionar una estimación del tiempo y recursos necesarios en la estimación de esfuerzo facilita la planificación y toma de decisiones. La identificación de riesgos técnicos contribuye a una gestión proactiva durante el proceso de control de cambios.

### Plantilla de Evaluación del CCB

1. **Fecha**: dd/mm/aa (Automática)
2. **Identificador**: EvalCCB + Código correlativo (Automático)
3. **Identificador del documento base**: [ID de la plantilla Evaluación del desarrollador, que inicia la presente evaluación]
4. **Miembros del CCB**:
   1. *Nombres:* [Nombre completo del miembro]
   2. *Correos:* [Correo electrónico del miembro]
   3. *Cargos:* [Cargo del miembro]
5. **Decisiones del CCB**:
   1. *Aprobación/Rechazo:* [Decisión del CCB sobre la solicitud de cambio]
   2. *Nivel de prioridad asignado:* [Nivel de prioridad otorgado a la solicitud: *Alto/Medio/Bajo*]
   3. *Justificación de la decisión:* [Descripción detallada del porqué de la aprobación o rechazo, junto al porqué de la prioridad asignada]

#### Modelo



#### Justificación de sus campos

1. **Fecha**: El registro de la fecha de evaluación del CCB permite un seguimiento temporal efectivo del proceso y facilita la organización de las evaluaciones.
2. **Identificador:** Asignar un identificador único a cada evaluación evita pérdidas o duplicaciones, facilitando el seguimiento, organización y referencia rápida de cada evaluación.
3. **Identificador del documento base:** Incluir el identificador del documento base asociado proporciona un vínculo claro entre la evaluación y la solicitud original, facilitando la trazabilidad y referencia.
4. **Miembros del CCB**: La captura del nombre completo y el cargo de los miembros del CCB es esencial para establecer una comunicación efectiva y para identificar de manera clara a las personas responsables de la evaluación. La información de contacto facilita la notificación sobre el progreso del proyecto, asegurando una colaboración eficiente y una participación informada de los miembros del CCB en el proceso de toma de decisiones.
5. **Decisiones del CCB:** Registrar la decisión del CCB, asignando un nivel de prioridad, es fundamental para la trazabilidad y el seguimiento del proyecto. Esta información actúa como una guía esencial para determinar la urgencia y relevancia de cada solicitud en el contexto del proyecto, facilitando una planificación efectiva y una asignación adecuada de recursos. Además, incluir una descripción detallada del porqué de la aprobación o rechazo es esencial para proporcionar una explicación clara y fundamentada. Esta práctica no solo contribuye a la transparencia en el proceso de toma de decisiones, sino que también facilita la comprensión y aceptación por parte de todos los interesados, promoviendo una comunicación abierta y efectiva en el equipo del proyecto.

### Plantilla de Notificación de rechazo

1. **Fecha**: dd/mm/aa (Automática)
2. **Identificador**: Rechazo + Código correlativo (Automático)
3. **Identificador del documento base**: [ID de la plantilla Evaluación del desarrollador o Evaluación del CCB, según quien decidiese rechazar el cambio]
4. **Datos del solicitante**:
   1. *Nombre*: [Nombre completo del solicitante]
   2. *Correo*: [Correo electrónico del solicitante]
5. **Datos del responsable de comunicación**:
   1. *Nombre*: [Nombre completo del solicitante]
   2. *Correo*: [Correo electrónico del solicitante]
   3. *ID de empleado:* [Número de empleado asignado al desarrollador]
6. **Resolución**:
   1. *Fecha de la decisión de rechazo*
   2. *Responsable:* [Datos del desarrollador, o CCB]
   3. *Justificación de la decisión:* [Descripción detallada para el cliente del porqué del rechazo]

#### Modelo



#### Justificación de sus campos

1. **Fecha**: Registrar la fecha de la notificación es esencial para mantener un registro claro y ordenado de la comunicación. Facilita el seguimiento temporal y permite una referencia precisa en el futuro.
2. **Identificador:** Asignar un identificador único a cada notificación de rechazo ayuda en la organización y seguimiento eficiente. Evita duplicaciones y proporciona una referencia rápida a la decisión específica.
3. **Identificador del documento base:** Vincular la notificación al documento base (evaluación del desarrollador o CCB) proporciona una conexión directa entre la decisión de rechazo y las evaluaciones realizadas anteriormente. Asegura trazabilidad y contexto.
4. **Datos del solicitante**: Incluir información sobre el solicitante permite una comunicación directa y facilita la comprensión del contexto. El correo electrónico proporciona un canal de comunicación efectivo.
5. **Datos del responsable de comunicación:** Identificar al responsable de la comunicación asegura que el cliente tenga un punto de contacto claro para discusiones o aclaraciones adicionales. Proporciona transparencia y eficacia en la gestión de la comunicación.
6. **Resolución:** Registrar la fecha de la decisión permite conocer el momento exacto en que se tomó la determinación. Indicar quién fue responsable de la decisión brinda transparencia y claridad sobre el origen de la elección. Proporcionar una descripción detallada de la razón del rechazo es crucial para que el cliente comprenda plenamente las circunstancias y razones detrás de la decisión. Facilita la transparencia y puede guiar al cliente en futuras acciones.

### Plantilla de Planificación

1. **Fecha**: dd/mm/aa (Automática)
2. **Identificador**: Plan+ Código correlativo (Automático)
3. **Identificador del documento base**: [ID de la plantilla Evaluación del CCB]
4. **Gerente del proyecto**:
   1. *Nombre:* [Nombre completo del gerente]
   2. *Correos:* [Correo electrónico del gerente]
   3. *ID Empleado:* [ID del gerente]
5. **Equipo de desarrollo**:
   1. *Nombres:* [Nombres completos de los miembros del equipo]
   2. *Correos:* [Correo electrónico de los miembros]
   3. Cargos: [Cargo de cada miembro del equipo]
   4. *ID Empleados:* [ID de cada miembro]
6. **Planificación**:
   1. *Tiempo estimado:* [Estimación del tiempo estimado de la ejecución de la planificación]
   2. *Recursos requeridos:* [Recursos necesarios para llevar a cabo la planificación]
   3. *Definición de tareas y actividades:* [Descripción detallada del porqué de la aprobación o rechazo, junto al porqué de la prioridad asignada]
   4. *Medidas de mitigación:* [medidas de mitigación para los posibles riesgos identificados]
7. **Póker Scrum:**
8. *Tiempo Mínimo:* [dueño + justificación]
9. *Tiempo Máximo:* [dueño + justificación]
10. *Tiempo más probable:* [número de veces que se repitió]
11. *Tiempo Medio:* [Resultado de aplicar la fórmula (*Tmin* + *4Tprob* + *Tmax*) /6]

#### Modelo



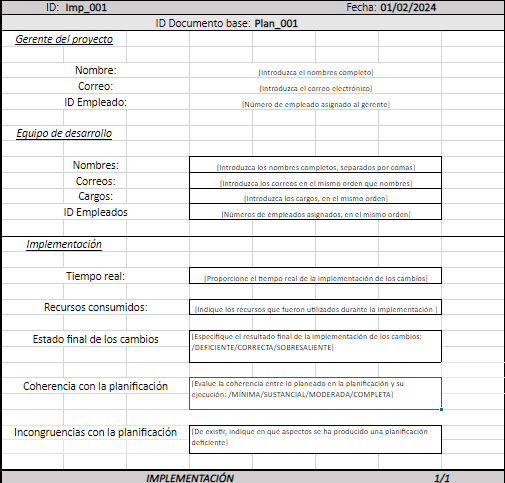
#### Justificación de sus campos

1. **Fecha**: El registro de la fecha de la creación de la planificación permite un seguimiento temporal efectivo del proceso.
2. **Identificador:** Asignar un identificador único a cada planificación evita pérdidas o duplicaciones, facilitando el seguimiento, organización y referencia rápida de cada planificación.
3. **Identificador del documento base:** Incluir el identificador del documento base asociado proporciona un vínculo claro entre la planificación y la evaluación del cambio por parte de la CCB, facilitando la trazabilidad y referencia.
4. **Gerente del proyecto**: La captura del nombre completo, ID y correo electrónico es esencial para establecer una comunicación efectiva y para identificar de manera clara a las personas responsables de la creación de la planificación. La información de contacto facilita la notificación sobre el progreso del proyecto, asegurando una colaboración eficiente.
5. **Equipo de desarrollo**: La inclusión del equipo de desarrollo permite identificar al resto de involucrados en la planificación. Facilita la comunicación interna y proporciona un punto de contacto clave para cualquier consulta o aclaración relacionada con la planificación del plan.
6. **Planificación:** Registrar la planificación desarrollada por el gerente del proyecto es crucial. Necesitamos saber el tiempo y recursos necesarios para la elaboración del proyecto, así como las actividades y tareas a realizar y las medidas de mitigación a desarrollar.
7. **Póker Scrum**: Registrar el proceso de decisión a través de póker scrum incluyendo las estimaciones con los valores trascendentes como son el tiempo máximo, mínimo (los dueños de estos dos), más probable y medio.

### Plantilla de Implementación

1. **Fecha**: dd/mm/aa (Automática)
2. **Identificador**: Plan+ Código correlativo (Automático)
3. **Identificador del documento base**: [ID de la plantilla Planificación]
4. **Gerente del proyecto**:
   1. *Nombre:* [Nombre completo del gerente]
   2. *Correos:* [Correo electrónico del gerente]
   3. *ID Empleado:* [ID del gerente]
5. **Equipo de desarrollo**:
   1. *Nombres:* [Nombres completos de los miembros del equipo]
   2. *Correos:* [Correo electrónico de los miembros]
   3. Cargos: [Cargo de cada miembro del equipo]
   4. *ID Empleados:* [ID de cada miembro]
6. **Implementación**:
   1. *Tiempo real:* [Tiempo real de la implementación de los cambios]
   2. *Recursos consumidos:* [Recursos que fueron utilizados durante la implementación]
   3. *Estado final de los cambios:* [Resultado final de la implementación de los cambios: /DEFICIENTE/CORRECTA/SOBRESALIENTE]
   4. *Coherencia con la planificación:* [Evaluación de la coherencia entre lo planeado en la planificación y su ejecución: /MÍNIMA/SUSTANCIAL/MODERADA/COMPLETA]
   5. *Incongruencias con la planificación:* [Inconsistencias entre la planificación y la implementación]

#### Modelo



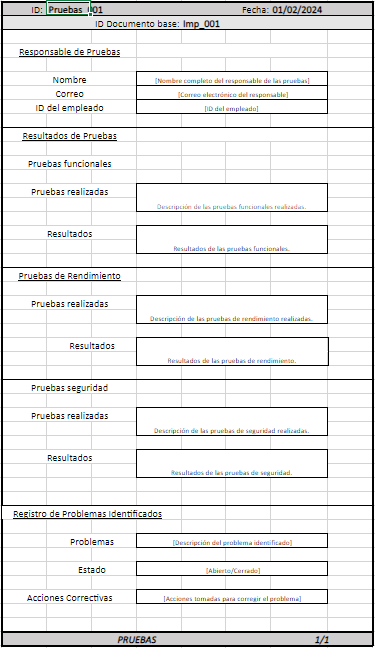
#### Justificación de sus campos

1. **Fecha**: La inclusión de la fecha proporciona un registro temporal automático que facilita la documentación y el seguimiento del progreso. La fecha permite contextualizar la implementación en un momento específico, lo que es esencial para el análisis histórico y la gestión temporal del proyecto.
2. **Identificador**: El identificador único ayuda a organizar y distinguir claramente cada implementación. Facilita la referencia y la recuperación de información, contribuyendo a una gestión eficiente del historial de cambios y permitiendo una identificación rápida y precisa.
3. **Identificador del Documento Base**: La vinculación con el documento base garantiza la trazabilidad y la coherencia entre la planificación y la implementación. Permite rastrear los detalles y decisiones tomadas durante la fase de planificación, brindando un contexto valioso para entender el proceso de implementación.
4. **Gerente del Proyecto**: La inclusión del gerente del proyecto especifica la persona responsable de supervisar y liderar la implementación. Facilita la comunicación interna y proporciona un punto de contacto clave para cualquier consulta o aclaración relacionada con la ejecución del plan.
5. **Equipo de desarrollo**: La inclusión del equipo de desarrollo permite identificar a los responsables de la implementación. Facilita la comunicación interna y proporciona un punto de contacto clave para cualquier consulta o aclaración relacionada con la ejecución del plan.
6. **Implementación**: Esta sección resume aspectos críticos relacionados con la ejecución de los cambios planificados. Incluye información esencial, como el tiempo real dedicado, los recursos utilizados y la evaluación del estado final de los cambios. Esta sección proporciona una visión rápida y estructurada del rendimiento de la implementación y es crucial para la evaluación y toma de decisiones posteriores.

### Plantilla de Pruebas

1. **Fecha:** dd/mm/aa (Automática)
2. **Identificador**: Pruebas + Código correlativo (Automático)
3. **Identificador del documento base**: [ID de la plantilla Implementación]
4. **Responsable de las pruebas**:
   1. *Nombre*: [Nombre completo del responsable de las pruebas]
   2. *Correo*: [Correo electrónico del responsable]
   3. *ID del empleado*: [Número de empleado asignado al desarrollador]
5. **Cambios implementados**: [Lista de cambios implementados previamente]
6. **Resultados de Pruebas**:
   1. *Pruebas funcionales*:
      1. *Pruebas realizadas*: [Descripción de las pruebas funcionales realizadas]
      2. *Resultados*: [Resultados de las pruebas funcionales]
   2. *Pruebas de rendimiento*:
      1. *Pruebas realizadas*: [Descripción de las pruebas de rendimiento realizadas]
      2. *Resultados*: [Resultados de las pruebas de rendimiento]
   3. *Pruebas de seguridad*:
      1. *Pruebas realizadas*: [Descripción de las pruebas de seguridad realizadas]
      2. *Resultados*: [Resultados de las pruebas de seguridad]
   4. *Registro de problemas identificados*:
      1. *Problemas*: [Descripción del problema identificado]
      2. *Estado*: [Abierto/Cerrado]
      3. *Acciones correctivas*: [Acciones tomadas para corregir el problema]

#### Modelo



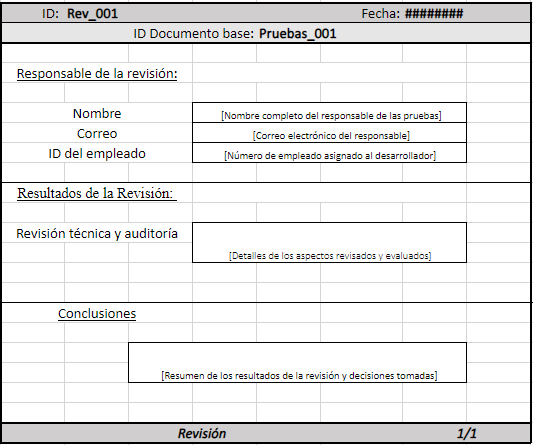
#### Justificación de sus campos

1. **Fecha**: El registro de la fecha de la creación de la planificación permite un seguimiento temporal efectivo del proceso.
2. **Identificador:** Asignar un identificador único a las pruebas evita pérdidas o duplicaciones, facilitando el seguimiento, organización y referencia rápida de las pruebas.
3. **Identificador del documento base**: La vinculación con el documento base garantiza la trazabilidad y la coherencia entre la implementación y las pruebas. Permite rastrear los detalles y decisiones tomadas durante la fase de implementación, brindando un contexto valioso para entender el proceso de pruebas.
4. **Responsable de las pruebas**: La inclusión del responsable de las pruebas especifica la persona responsable de supervisar y liderar la fase de pruebas. Facilita la comunicación interna y proporciona un punto de contacto clave para cualquier consulta o aclaración relacionada con la ejecución del plan.
5. **Cambios implementados**: Este campo especifica los cambios o actualizaciones que se han implementado en el sistema o producto que está siendo probado. Proporciona contexto sobre qué aspectos del sistema están siendo evaluados durante las pruebas.
6. **Resultados de Pruebas**: Este campo registra los resultados de las pruebas realizadas, incluyendo cualquier defecto encontrado, así como la confirmación de que los requisitos y expectativas definidos se han cumplido satisfactoriamente. También registra los problemas detectados y sus acciones correctivas.

### Plantilla de Revisión

1. **Fecha:** dd/mm/aa (Automática)
2. **Identificador**: Rev + Código correlativo (Automático)
3. **Identificador del documento base**: [ID de la plantilla Pruebas]
4. **Responsable de la revisión**:
   1. *Nombre*: [Nombre completo del responsable de la revisión]
   2. *Correo*: [Correo electrónico del responsable]
   3. *ID del empleado*: [Número de empleado asignado al responsable]
5. **Resultados de la Revisión**:
   1. *Revisión técnica y auditoría*: [Detalles de los aspectos revisados y evaluados]
6. **Conclusiones**: [Resumen de los resultados de la revisión y decisiones tomadas]

#### Modelo



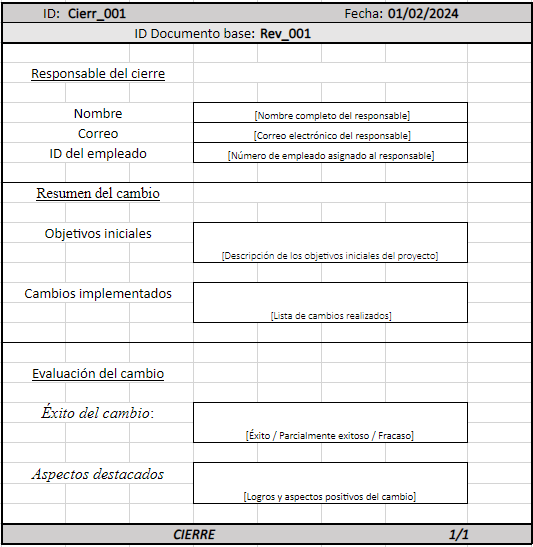
#### Justificación de sus campos

1. **Fecha:** El registro de la fecha de la creación de la revisión permite un seguimiento temporal efectivo del proceso.
2. **Identificador:** Asignar un identificador único a cada revisión evita pérdidas o duplicaciones, facilitando el seguimiento, organización y referencia rápida de cada revisión.
3. **Identificador del documento base**: La vinculación con el documento base garantiza la trazabilidad y la coherencia entre la revisión y la fase de pruebas. Permite rastrear los detalles y decisiones tomadas durante la fase de pruebas, brindando un contexto valioso para entender el proceso de revisión.
4. **Responsable de la revisión:** La inclusión del responsable de la fase de pruebas especifica la persona responsable de supervisar y liderar esta. Facilita la comunicación interna y proporciona un punto de contacto clave para cualquier consulta o aclaración relacionada con la ejecución del plan.
5. **Resultados de la revisión**: Este campo registra los hallazgos, observaciones y evaluaciones realizadas durante la revisión del proceso o documento. Incluye detalles sobre la calidad del trabajo realizado y cualquier otra información relevante identificada durante la revisión.
6. **Conclusiones**: Este campo resume las principales conclusiones y recomendaciones derivadas de la revisión. Proporciona una visión general de los aspectos positivos y áreas de mejora identificadas durante el proceso de revisión. En caso de haber detectado errores se indicaría la necesidad de volver a la actividad de planificación para solucionar las problemáticas.

### Plantilla de Cierre

1. **Fecha:** dd/mm/aa (Automática)
2. **Identificador**: Cierr + Código correlativo (Automático)
3. **Identificador del documento base**: [ID de la plantilla Revisión]
4. **Responsable del cierre**:
   1. *Nombre*: [Nombre completo del responsable]
   2. *Correo*: [Correo electrónico del responsable]
   3. *ID del empleado*: [Número de empleado asignado al responsable]
5. **Resumen del cambio**:
   1. *Objetivos iniciales*: [Descripción de los objetivos iniciales del cambio]
   2. *Cambios implementados*: [Lista de cambios realizados]
6. **Evaluación del cambio**:
   1. *Éxito del cambio*: [Éxito / Parcialmente exitoso / Fracaso]
   2. *Aspectos destacados*: [Logros y aspectos positivos del cambio]

#### Modelo



#### Justificación de sus campos

1. **Fecha:** El registro de la fecha de la creación del cierre permite un seguimiento temporal efectivo del proceso.
2. **Identificador:** Asignar un identificador único al cierre evita pérdidas o duplicaciones, facilitando el seguimiento, organización y referencia rápida del cierre.
3. **Identificador del documento base**: La vinculación con el documento base garantiza la trazabilidad y la coherencia entre el cierre y la revisión. Permite rastrear los detalles y decisiones tomadas durante la fase de revisión, brindando un contexto valioso para entender el proceso de cierre.
4. **Responsable del cierre**: La inclusión del responsable del cierre especifica la persona responsable de supervisar y liderar el cierre. Facilita la comunicación interna y proporciona un punto de contacto clave para cualquier consulta o aclaración relacionada con la ejecución del plan.
5. **Resumen del cambio**: Este campo proporciona un resumen conciso de los aspectos más relevantes del cambio, incluidos los objetivos y las modificaciones realizados. Sirve como referencia rápida para entender el contexto del cambio durante el proceso de cierre.
6. **Evaluación del cambio**: Aquí se realiza una evaluación del cambio, incluyendo puntos como el éxito de este (donde definimos el nivel de éxito conseguido) y los aspectos destacados (que incluye los logros y puntos positivos del cambio).

# ANEXOS

## Anexo 1.- Cambios propuestos por cada miembro del grupo

### Nicolás Barcia Quintela

* *Requisito funcional:* Poder exportar un pdf con todas las operaciones realizadas dentro de un grupo de gasto.
* *Requisito no funcional:* Añadir registro/inicio de sesión con la cuenta de Google de forma externa.
* *Problemas identificados en los documentos:* Eliminar requisitos organizacionales de entrega en el diagrama de requisitos no funcionales.

### Jesús Campos Manjón

* *Requisito funcional:* Implementar una funcionalidad que permita clasificar los gatos por categorías.
* *Requisito no funcional:* Integrar prácticas de desarrollo sostenible, minimizando la huella de carbono de la aplicación y sus servicios asociados
* *Problemas identificados en los documentos:* El requisito de información CRQ-0005 (Frecuencia de envío) está mal descrito, precisa una explicación más detallada del requisito en general y de lo que es un “uso indebido”.

### Guillermo Madriñán González

* *Requisito funcional:* Permitir programar recordatorios personalizados en la configuración de notificaciones.
* *Requisito no funcional:* Añadir un método de configuración de la interfaz para adaptarla al gusto del usuario.
* *Problemas identificados en los documentos:* Reducir el glosario de forma que sea más simple y especifico como guía para el lector.

### Roi Martínez Enríquez

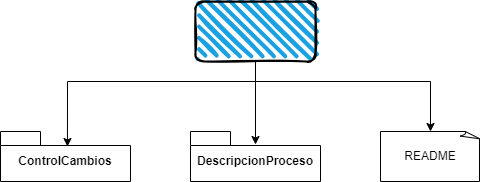
* *Requisito funcional:* Implementar una integración fluida con la base de datos externa de OldTravelsa para garantizar la sincronización de información relevante.
* *Requisito no funcional:* Implementar medidas sólidas de protección de datos personales extraídos de la base de datos externa.
* *Problemas identificados en los documentos:* Se observa la presencia de diagramas de subsistemas con un número reducido de casos de uso (1 o 2). Se propone la consolidación de múltiples subsistemas pequeños en uno único para mejorar la eficiencia y claridad del diseño.

## Anexo 2.- Ejecución del proceso

Después de haber elaborado un proceso de control de cambios junto con las plantillas correspondientes en etapas previas de esta práctica, el equipo se enfrentará esta semana a la ejecución de la mitad de los cambios identificados en el Anexo 1.- Cambios propuestos por cada miembro del grupo.

### Organización del repositorio

La primera tarea consistió en la creación de un nuevo repositorio en GitHub y en la configuración para permitir el acceso a todos los miembros del grupo. Una vez creado, se procedió a organizar la estructura del repositorio de la siguiente manera:



* *ControlCambios*: En esta carpeta se encuentra la estructura desarrollada en la versión 3 del documento, que incluye los archivos y plantillas relacionados con el proceso de control de cambios.
* *DescripcionProceso*: En esta carpeta se incluyen todas las versiones del documento a lo largo del proceso de control de cambios.
* *README*: Se agregó un archivo adicional para proporcionar una descripción sobre el propósito del repositorio y los ID de cada uno de los miembros implicados.

### Distribución del trabajo

Dado que todos los miembros del equipo están reunidos presencialmente al inicio de la tarea, se procede a planificar la distribución del trabajo centrándose en asignar un único rol a cada miembro. Las actividades propuestas para la primera ejecución del proceso, relacionadas con los requisitos funcionales, son las siguientes:

|  |  |
| --- | --- |
| **Integrante** | **Roles** |
| **Nicolás Barcia** | Desarrollador, miembro del CCB y analista |
| **Jesús Campos** | Desarrollador, miembro del CCB y programador |
| **Guillermo Madriñán** | Miembro del CCB y analista |
| **Roi Martínez** | Identificador de solicitudes, miembro del CCB y analista |

Sin embargo, aunque esta distribución sería lógica en un proceso "real", asignando a cada recurso humano uno o varios roles específicos, se observa que no es óptima. Esto impide la paralelización de tareas y obliga, por ejemplo, a que los miembros analistas esperen la identificación de solicitudes para poder realizar su trabajo. Por esta razón, durante la mitad de la primera ejecución, se decide cambiar el enfoque de distribución de trabajo. Cada miembro se hará responsable de la ejecución de un cambio. Esta nueva división facilita que cada miembro pueda avanzar en sus tareas asignadas una vez finalice la reunión. Solo se requerirá una reunión posterior para organizar la preparación y entrega del repositorio. A continuación, se muestra el nuevo reparto:

|  |  |
| --- | --- |
| **Integrante** | **Cambios asignados** |
| **Nicolás Barcia** | 001, 005 y 006 |
| **Jesús Campos** | 003, 008 y 010 |
| **Guillermo Madriñán** | 004, 011 y 012 |
| **Roi Martínez** | 002, 007 y 009 |

### Cambio de planes respecto a la versión 3 del documento

#### Política de gestión de etiquetas

Aunque gran parte del proceso se ejecutó según lo planeado, fue necesario adoptar una nueva política de gestión de etiquetas debido a una malinterpretación de la funcionalidad de estas en versiones anteriores del documento. Así, anteriormente, se interpretaba la utilización de etiquetas para indicar el progreso de cada cambio, como, por ejemplo, "*Cambio\_002: Planificado*" en el commit que añadiría la plantilla completada de planificación del cambio implicado al repositorio.

La primera modificación de la política de etiquetas se centraba en las tres ejecuciones previstas. Por ejemplo, al completar los cuatro primeros cambios, se incluiría una etiqueta en el último de los commits del tipo "*Ejecución\_1*". Sin embargo, al completar el repositorio, se observó que en el enunciado del Campus Virtual se requería incluir etiquetas en el repositorio de las versiones finales después de cada cambio aceptado e implementado. Por lo tanto, se necesitó una segunda modificación que cumpliera con esta condición, cuya ejecución se detallará en el siguiente apartado.

#### Política de identificación de los miembros

Además de la modificación en la política de gestión de etiquetas, se decidió realizar un ajuste en la política de identificación de los miembros implicados en el repositorio. En la versión 3 del documento, se contemplaba la inclusión de un archivo PDF en cada una de las carpetas asociadas a las etapas del proceso. Este archivo contendría información sobre los miembros involucrados en cada etapa, como sus nombres, correos electrónicos e identificadores, con el fin de facilitar la comunicación en caso de incidencias.

Sin embargo, dado que el equipo está compuesto por solo cuatro miembros para toda la ejecución, se optó por incluir estos IDs en el README del repositorio en lugar de crear archivos repetidos en cada carpeta para simplificar la estructura del repositorio y evitar redundancia. No obstante, se reconoce que, en el caso de un aumento considerable en el tamaño del grupo, donde pueda haber personal asignado a tareas específicas de la ejecución, la inclusión de estos archivos identificadores en cada carpeta ganaría importancia.

### Problemas encontrados y soluciones

#### Modificación de etiquetas

Como se mencionó en el apartado anterior, una vez completada la ejecución del proceso, fue necesario revisar y ajustar las etiquetas previamente establecidas. Esta situación requirió explorar las capacidades proporcionadas por Git para observar la evolución de las versiones del proceso y la capacidad de asignar etiquetas a momentos específicos anteriores.

En particular, los comandos ***git log*** (para revisar cada uno de los commits y merges realizados anteriormente) y ***git tag*** (utilizando el hash del merge en la rama principal correspondiente a cada cambio) permitieron a Roi Martínez Enríquez, el miembro encargado de la tarea, crear las nuevas etiquetas requeridas.

#### Commit no deseado

Uno de los miembros del grupo realizó un commit no deseado, que fusionaba varias ramas sin intención. Esto resultó en la necesidad de intentar hacer una serie de modificaciones adicionales en el repositorio, las cuales no solo ensuciaron el historial, sino que tampoco resolvieron el problema. La solución adoptada, aunque eficiente, fue arriesgada: se realizó un ***git reset --hard HEAD~4*** (siendo 4 el número de versiones necesarias para retroceder antes del cambio no deseado), seguido de un ***git push --force***. Este último parámetro fue necesario para superar la restricción que Git impone al intentar subir ramas consideradas "antiguas".

#### Inconsistencia en los títulos de los *merge*

Al abordar el primer problema mencionado, se identificó una inconsistencia en los títulos de cada uno de los merge realizados. Esto se debió a dos razones principales: la falta de una política de títulos de los merge en las versiones del documento y el uso de diferentes métodos por parte de cada miembro para realizar las actualizaciones en la rama principal (GitHub Desktop, la página web de GitHub, VisualStudio, entre otros).

Aunque esta inconsistencia no tuvo un gran impacto en la resolución del problema inicial, una mayor coherencia habría facilitado considerablemente el proceso.

### Gestión de varias solicitudes de cambio en el proceso de análisis

Para gestionar de manera simultánea varias solicitudes en la misma ejecución del proceso de ejecución del cambio, se emplea una estrategia que combina la distribución de trabajo, previamente explicada en el apartado 3.2.2, con la creación de ramas paralelas en Git. Estas ramas paralelas permiten la elaboración de los productos de cada etapa y la ejecución del cambio de manera independiente, sin afectar a la rama principal del repositorio o provocar interferencias entre los diferentes cambios en proceso.

Cada miembro del equipo se responsabiliza así de una rama asociada a un cambio específico. Una vez que el cambio ha sido completamente ejecutado y verificado, se realiza el *merge* de la rama asociada a la rama principal del repositorio y esta es eliminada. Además, se marca la etiqueta correspondiente al final del cambio para mantener un registro claro de la evolución del proceso y de las versiones finales de cada cambio implementado.

### Detalles de la estimación con Scrum Poker

Las estimaciones reflejadas en las plantillas de planificación de los cambios se realizan haciendo uso del Scrum Poker. En este proceso participan, por un lado, el gerente de la planificación, que, en nuestro caso particular, coincide con la persona que propuso el cambio; y el equipo encargado de la implementación, que serán todos los miembros de nuestro grupo, ya que contamos con un número muy limitado.

Los miembros implicados en la implementación del cambio y el gerente de la planificación empiezan exponiendo su punto de vista sobre el desarrollo del cambio. Después se realiza la votación y en caso de convergencia (las votaciones se corresponden con 3 cartas consecutivas), se calcula el tiempo final. De no llegar a converger, se repetirán las votaciones, añadiendo una discusión a posteriori, hasta llegar a un punto de convergencia.

A continuación, se explicará de forma individual el proceso y los resultados de las votaciones realizadas para aquellos cambios que llegaron a la fase de planificación. Se expondrá cualquier imprevisto o situación anómala que haya podido suceder durante la votación. Los cambios que fueron rechazados en alguna fase previa, como cabría esperar, no constan en ninguno de los siguientes subapartados.

#### Cambio 001

Propuesto por Nicolás Barcia. El cambio es el siguiente:  
-Poder exportar un pdf con todas las operaciones realizadas dentro de un grupo de gasto.

Para reflejar fielmente el proceso, explicaremos en este primer cambio el proceso de Scrum Poker al detalle.   
Nicolás Barcia, gerente de la planificación de este cambio, expuso el cambio propuesto, para que a posteriori, el resto del equipo encargado de la implementación, compuesto por Jesús Campos (desarrollador), Guillermo Madriñán (desarrollador) y Roi Martínez (desarrollador) expusiera sus dudas.

A continuación, se llevó a cabo la votación. Los resultados de esta fueron muy dispares, por un lado, Jesús Campos tenía una visión muy optimista, argumentando que el uso de librerías permitiría realizar este cambio de una forma muy sencilla, pudiendo tener el cambio finalizado en apenas un día. Por otro lado, el desarrollado Roi Martínez, quien inicialmente votó por 20 días, exponía que la labor de protección de datos y de asegurar la compatibilidad entre distintos sistemas operativos podría ser tediosa y complicada. Nicolás Barcia, gerente de la planificación destacó que ambos tenían parte de razón, lo que dio lugar a un estado de convergencia en la siguiente votación, aunque ambos desarrolladores continuaron teniendo posturas distintas.  
  
El resultado final de las votaciones fue el siguiente:

Votaciones:  
Jesús Campos: Vota 3 días. Argumenta que el uso de librerías facilitaría la implementación.

Nicolás Barcia: Vota 7 días.

Roi Martínez: Vota 13 días. Argumenta que puede ser complicado asegurar la protección y seguridad de los datos, así como la implementación en distintos sistemas operativos.

Guillermo Madriñán: Vota 7 días.

Al ser cartas consecutivas las obtenidas en la votación, se considera que hay convergencia. Por lo tanto, se aplica la fórmula para obtener el tiempo medio con el que trabajaremos.

Tiempo Mínimo: 3 días.

Tiempo Máximo: 13 días

Tiempo más probable: 1 semana (repetido 2 veces)

Tiempo Medio: 7 días

#### Cambio 002

Propuesto Por Roi Martínez. El cambio es el siguiente:

-Implementar una funcionalidad que permita clasificar los gatos por categorías.

Votaciones:  
Jesús Campos: Vota 3 semanas.

Nicolás Barcia: 8 semanas. Argumenta que pese a que ya existan en el mercado actual, la integración no es sencilla y garantizar la seguridad de los datos y usabilidad para sus usuarios implica una gran carga temporal

Roi Martínez: 3 semanas.

Guillermo Madriñán: 2 semanas. Explica que existen ejemplos similares ya realizados, implicaría mayoritariamente la integración con plataformas externas.

Tras repetir la votación y debatir el caso en repetidas ocasiones, los resultados no varían y la posición de Nicolás sigue siendo muy alejada a la del resto de analistas. Por esto, se desiste de alcanzar un estado de convergencia y se procede a la aplicación de la formula.

Tiempo Mínimo: 2 semanas.  
Tiempo Máximo: 8 semanas:   
Tiempo más probable: 3. Repetido 2 veces

Tiempo Medio: 3.66 semanas

#### Cambio 005

Propuesto por Nicolás Barcia. El cambio es el siguiente:

- Añadir registro/inicio de sesión con la cuenta de Google de forma externa.

En la primera votación los resultados son los siguientes:

Votaciones:  
Jesús Campos: Vota 0.5 semanas. Defiende que la implementación es muy fácil debido a que es un proceso muy común y está muy bien documentado.

Nicolás Barcia: 2 semanas. Expone que pueden tener que realizarse ajustes de compatibilidad con los métodos de registro e inicio de sesión actuales.

Roi Martínez: 1 semana.

Guillermo Madriñán: 1 semana.

Son 3 cartas consecutivas por lo que se considera que existe convergencia y se procede al cálculo pertinente.

Tiempo Mínimo: 0.5 semanas.

Tiempo Máximo: 2 semanas.

Tiempo más probable: 1. Repetido 2 veces

Tiempo Medio: 1.13 semanas

#### Cambio 007

Propuesto por Roi Martínez. El cambio es el siguiente:

-Añadir un método de configuración de la interfaz para adaptarla al gusto del usuario.

Al igual que en el caso anterior los resultados de la primera votación concuerdan con un estado de consenso, por lo que se calcula directamente el tiempo medio.

Votaciones:  
Jesús Campos: Vota 4 días. Asume que el equipo de desarrollo, con experiencia en la arquitectura del sistema y diseño de interfaz, puede abordar la implementación de manera eficiente y sin mayores contratiempos.

Nicolás Barcia: Vota 1 semana.

Roi Martínez: Vota 1 semana.

Guillermo Madriñán: Vota 2 semanas. Defiende la necesidad de pruebas más exhaustivas de lo anticipado y posibles conflictos con otras funcionalidades.

Tiempo Mínimo: 4 días.  
Tiempo Máximo: 2 semanas.  
Tiempo más probable: 1 semana. (2 repetidos)

Tiempo Medio: 1'1 semanas

#### Cambio 010

Propuesto por Jesús Campos. El cambio es el siguiente:

- Se observa la presencia de diagramas de subsistemas con un número reducido de casos de uso (1 o 2). Se propone la consolidación de múltiples subsistemas pequeños en uno único para mejorar la eficiencia y claridad del diseño.

Votaciones:  
Jesús Campos: Vota 5 días.

Nicolás Barcia: Vota 8 días. Esta estimación considera posibles complicaciones como retrasos en la revisión de subsistemas debido a dificultades técnicas imprevistas, necesidades de rediseño extenso, y demoras en las pruebas debido a la identificación de errores críticos.

Roi Martínez: Vota 3 días: La justificación para este tiempo se basa en la experiencia previa con proyectos de consolidación similares. La estimación considera la posibilidad de reutilizar partes del código existente y la disponibilidad completa del equipo asignado al proyecto.

Guillermo Madriñán: Vota 5 días.

Una vez realizadas varias votaciones buscando una convergencia a través de explicaciones de las diferentes posturas, se alcanza ese consenso y resulta en las siguientes estimaciones:

Tiempo Mínimo: 3 días.

Tiempo Máximo: 8 días.

Tiempo más probable: 5 días. (3 repetidos)

Tiempo Medio: 5,16 días

#### Cambio 011

Propuesto por Guillermo Madriñán. El cambio es el siguiente:  
- Reducir el glosario de forma que sea más simple y especifico como guía para el lector.

Después de una votación algo más dispersa, se presentaron las diferentes defensas de los puntos de vista de los participantes, resultando en unos votos que dan lugar a la convergencia.

Votaciones:  
Jesús Campos: Vota 3 días.

Nicolás Barcia: Vota 2 días. Defiende que al existir ya un glosario, aunque que demasiado basto para ser el definitivo, el trabajo sería sencillo.

Roi Martínez: Vota 3 días.

Guillermo Madriñán: Vota 4 días. Expresa la necesidad de especializar el glosario en mayor medida, buscando una comprensión lo mejor posible del sistema.

Tiempo Mínimo: 2 días

Tiempo Máximo: 4 días

Tiempo más probable: 3 días repetido 2 veces

Tiempo Medio: 3 días

## Anexo 3.- Bibliografía y material utilizado

* [*PresGC\_Dia1.pdf*](https://cv.usc.es/pluginfile.php/2387425/mod_folder/content/0/Presentaciones/PresGC_Dia1.pdf?forcedownload=1)
* [*PresGC\_Dia2.pdf*](https://cv.usc.es/pluginfile.php/2387425/mod_folder/content/0/Presentaciones/PresGC_Dia2.pdf?forcedownload=1)
* *[PMBOK 6ªEd](https://cv.usc.es/pluginfile.php/2122501/mod_resource/content/2/PMBOK_6_ES.pdf)*

## Anexo 4.- Relatorio de documentos asociados a éste

|  |  |  |
| --- | --- | --- |
| **Nombre del documento** | **Software de visualización (versión)** | **Descripción del documento** |
| Plantillas.xlsx | Microsoft Excel | Documento con las plantillas de las actividades de nuestro proceso. |
| P6\_Seguimiento\_G1[2]\_v4.pod | Project Libre | Distribución de tareas y trabajo de cada miembro del equipo. |