UNIVERSIDAD MAYOR DE

SAN MARCOS

Universidad del Perú, DECANA DE AMÉRICA.

*FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA*

**PLAN DE LA GESTIÓN DE LA CONFIGURACIÓN**

|  |  |  |  |
| --- | --- | --- | --- |
| Curso | : | Gestión de la Configuración de Software |  |
| Profesor | : | Lenis Rossi Wong Portillo |  |
| Escuela Profesional | : | Ingeniería de Software |  |
| Alumnos | : | Aguilar Salazar, Edwin Ccari | 18200323 |
|  |  | Antúnez Palomino, Kori Xiomara | 18200110 |
|  |  | Gomez Cavero, Mishell | 18200267 |
|  |  | Palomino Gutierrez, Erick Victor | 18200283 |
|  |  | Sandoval Salinas, Anthony Kevin Jerremy | 18200294 |
|  |  | Suarez Herandez, Kenny Joel | 18200102 |
|  |  | Vilca Daza, Diego Percy | 18200128 |

**Lima – Perú**

**Noviembre,2020** 

*Coffee & Lima: Programming Studio*

**Plan de la GCS**

**Versión 1.0**

**Índice**

[**Introducción**](#_vculzdj1o41l)6

[Situación de la empresa (Erick Palomino)](#_6g7xiha4ycwk) 6

[Propósito (Mishell Gomez)](#_sbd8ej1uroj2) 6

[Alcance (Kori Antunez)](#_pqe5g1i42278) 7

[**Gestión de la SCM**](#_x5uo1f7xn1r)7

[Roles o responsabilidades](#_u5xp69v7v23p) 7

[Procedimientos (Edwin Aguilar)](#_vtl97i4ioiak) 8

[Herramientas](#_rdcf7necz5qo) (Diego Vilca y Kenny Suarez) 8

[Calendario (Anthony Sandoval)](#_fse0b2fc9zxs) 12

[**Actividades de la SCM**](#_6snxqp1b4j2r)14

[Identificación de la configuración](#_urvi0793d11g) 14

[Control de la Configuración](#_wi6zqgp1sie5) 15

[Estado de la Configuración](#_8qreciv67nrt) 18

[Auditoría de la Configuración](#_bhqtn24j648k) 22

[Gestión de entrega y Release de Software](#_aln1a558n9k3) 22

# Introducción

## Situación de la empresa (Erick Palomino)

Coffee & Lima: Programming Studio es una empresa nueva en el mercado de desarrollo de software. Actualmente no cuenta con un área encargada de la gestión de la configuración de sus productos y ha habido una propuesta de cierto número de especialistas para poder formalizar este nuevo departamento.

De esta forma como trabajo inicial de este futuro departamento se ha realizado un documento de planificación de la gestión de la configuración de sus productos software, proponiendo así la creación de un área especializada en la gestión de la configuración de software.

Esta propuesta tiene por objetivo disminuir o eliminar los constantes problemas presentados al controlar las versiones de los artefactos construidos dentro de los procesos de desarrollo.

Específicamente aquellos problemas presentados en la contratación de nuevos empleados, estandarización en la codificación del producto, rotación de personal, reglamentar la organización de los artefactos dentro del repositorio y controlar las versiones de los ítems y los releases para los clientes.

## Propósito (Mishell Gomez)

El presente documento tiene como propósito elaborar el plan de gestión de la configuración de los productos software desarrollados por la empresa Coffee & Lima: Programming Studio con el objetivo de llevar un control y registro de las versiones que se realicen durante el ciclo de vida del desarrollo de software, garantizando de esta manera que no se efectúen cambios incontrolados y que todo el equipo de desarrollo disponga de la versión adecuada. Asimismo, la gestión de la configuración facilitará un correcto mantenimiento de los sistemas desarrollados, aportará información para evaluar el impacto de los cambios que se soliciten, permitirá obtener informes acerca del estado de desarrollo en el que se encuentra y reducir la cantidad de errores de adaptación del sistema; lo que se traduce en un aumento de la calidad del producto, satisfacción de los clientes y ,por consiguiente, la mejora de la organización.

## Alcance (Kori Antúnez)

El plan de la Gestión de la Configuración de Software de la empresa Coffee & Lima: Programming Studio presenta como alcance los siguientes supuestos, los cuales, son descritos a continuación:

* El tiempo de duración del proyecto está limitado a 13 semanas; por ello, el equipo de proyecto busca lograr una rápida respuesta a los cambios, optimizando el proceso para reducir el tiempo.
* Se mantiene la integridad de los entregables que se obtienen a lo largo del proceso de desarrollo del proyecto, supervisando que no se realicen cambios incontrolados.
* Se garantiza que todo el equipo participe en el desarrollo de la página web y que dispongan de las versiones adecuadas.
* La gestión de la configuración se realizará en todas las actividades del desarrollo de la página web.
* Se permite que la gestión de la configuración facilite el desarrollo de los cambios solicitados, reduciendo el tiempo de desarrollo de un cambio. De esta manera se tendrá un control del desarrollo de la página web a lo largo de su desarrollo y así reducir el número de errores de la página web, lo que se traduciría en una mejor calidad.

# Gestión de la SCM

## Roles o responsabilidades

|  |  |  |
| --- | --- | --- |
| **Rol** | **Responsabilidades** | **Número** |
| Gestor de la CS | * Se encarga de realizar reportes y gestionar el control de versiones * Responsable de que los cambios estén bien definidos | **2** |
| CCC | * Evaluar el impacto de los cambios | **Variado** |
| Bibliotecario | * Gestionar el repositorio. * Brindar accesos a las bibliotecas | **1** |
| Auditor de la GCS | * Gestión y control del adecuado de los cambios según se haya registrado en los documentos. | **Variado** |
| Equipo de desarrollo | * Definir, trabajar, desarrollar o modificar los elementos del proyecto según el plan de la gestión de la configuración | **Variado** |

## Procedimientos (Edwin Aguilar)

* Cronograma del proyecto
* Estándares internos de la organización
* Documento de Aseguramiento de la Calidad

## Herramientas

### Github (Kenny Suarez)

Características:

* Collaborative coding: Presenta muchas opciones que permite a los colaboradores del repositorio trabajar de una manera más eficiente y cohesiva. tiene diversas sub herramientas como: Codespace, Pull Request, Notificaciones, visualizador de código, multilínea de códigos, etc.
* Automation & CI/CD: Presenta opciones que automatizan los procesos de los usuarios por lo que ahorra mucho tiempo a los colaboradores, además de trabajar con prácticas de integración continua y despliegue continuo.
* Security: Evalúa las vulnerabilidades del código antes de despliegue. Además ofrece los siguientes servicios para mejorar la seguridad de nuestro código: GitHub Security Advisories, GitHub Advisory Database, GPG commit signing verification, Security audit log, entre otros.

Enterprise Security:

Existen algunas características que solo están disponibles con el modo Enterprise, estas son: SMAL, LDAP, IP allow list, Github connect, Audit log API.

* Cliente apps: Github está disponible en diversos dispositivos (computador de escritorio, dispositivos móviles y la clásica consola).
* Project management: Puede dirigir tus proyectos de manera coordinada con las diversas herramientas que ofrecen, como, por ejemplo: Projects, Labels, Issues, gráficos de actividad de organización, gráficos de contribución unificados, repo insights, entre otros.
* Team Administration: Permite administrar el equipo colaborador del repositorio, con lo que puedes agregar o quitar miembros, asignar tareas, conceder permisos o retirarlos. Además se sincroniza con Okta y Azure.

Modos de usuario:

* Free mode: El modo gratuito es el más usado de la plataforma, y tiene servicios muy buenos que permite a un usuario trabajar con sus compañeros sin mucho problema.

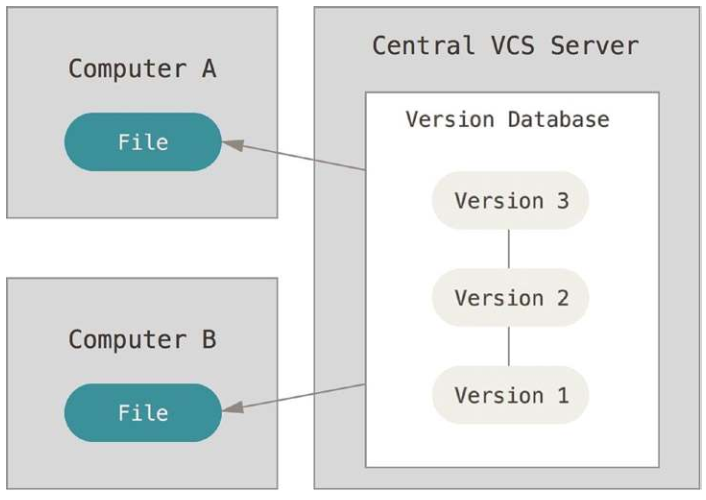
Los servicios que ofrece el modo free son:

* + Ilimitados repositorios públicos y privados.
  + Colaboradores ilimitados.
  + 2,000 Actions minutes/month
  + Repositorios públicos gratuitos.
  + 500MB de almacenamiento para GitHub
  + Soporte de la comunidad
* Team Mode: Ofrece más soporte para los equipos de desarrollo con avanzadas opciones para mejorar la colaboración
  + Ilimitados repositorios públicos y privados.
  + Revisores requeridos
  + 3,000 Actions minutes/month
  + Repositorios públicos gratuitos
  + 2GB de almacenamiento para Github
  + Propietarios de códigos
* Enterprise Mode:
  + Posee todo lo incluido en el Team mode
  + Inicio de sesión SAML
  + 50,000 Actions minutes/month
  + Repositorios públicos gratuitos
  + 50GB de almacenamiento para Github
  + Auditoría avanzada

### Git (Diego Vilca)

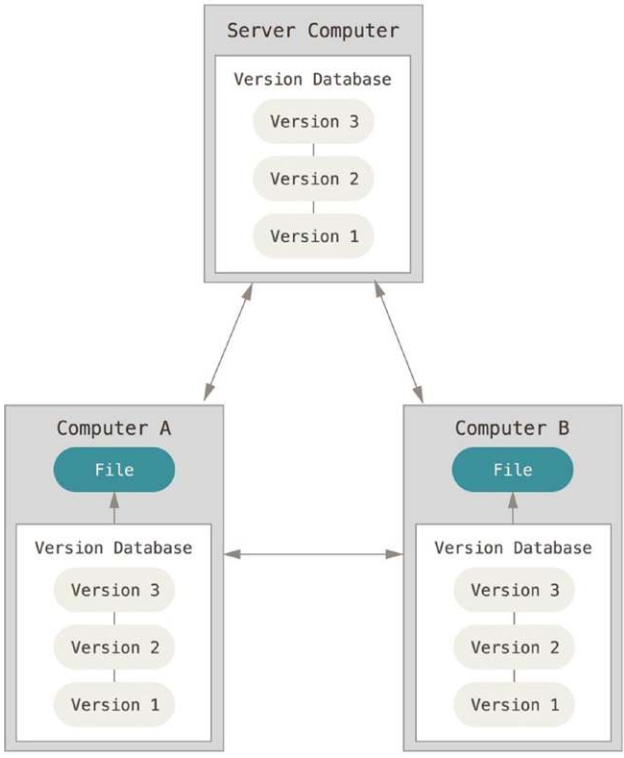
Es un sistema de control de versiones distribuido, es decir, gestiona diversos cambios que se realizaron en un elemento de configuración, esto se logra mediante el registro de los cambios en una base de datos de versiones. Esto le permite al sistema por ejemplo poder dirigirse a una versión específica.

Antes de definir que es un sistema de control de versiones distribuidos, se debe saber que diferencia tiene con un sistema de control de versiones centralizado. La principal diferencia es la forma en cómo trabaja cada sistema en almacenar la información de las versiones. En el caso del sistema centralizado, esta información es guardada en un servidor único donde se les permite a los desarrolladores participantes extraer versiones específicas de la base de datos del servidor.



Como se muestra en la figura, las computadoras A y B extraen la información que requieren de la base de datos del servidor central, en este caso, un archivo. Nótese que no extrae la base de datos de las versiones, solo la versión que requiera en ese momento.

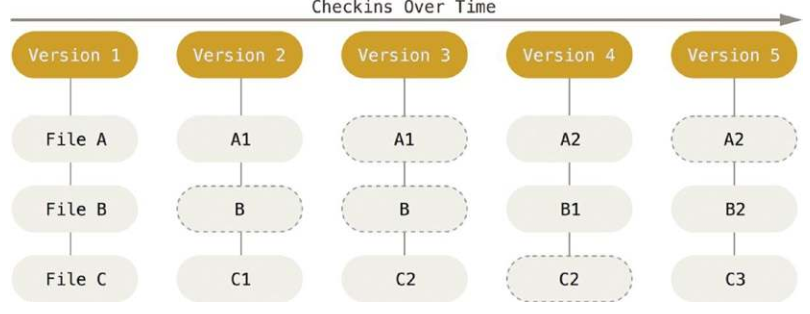
De lo contrario, un sistema distribuido no solo permite a los desarrolladores extraer una versión de un archivo o un conjunto de archivos, sino también permite extraer el historial de versiones de esos archivos.



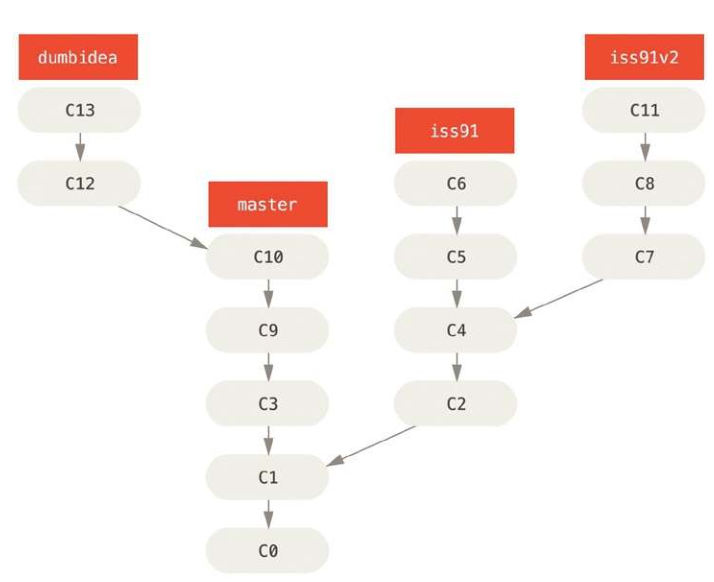
Debido a que como todo el historial se encuentra en un punto central en el sistema centralizado, esto hace que la información de las versiones se puedan perder en caso de un error o caída y no se pueda recuperar.

Mientras que en un sistema distribuido, si se pierde el servidor central, los repositorios. la cual es el conjunto de archivos del proyecto y su historial de versiones, aún se encuentran en las otras computadoras o servidores de los desarrolladores. La cual es una ventaja principal de este tipo de sistemas.

Además Git guarda los datos de un archivo en forma de instantáneas o snapshots, en caso que una versión del proyecto guarde archivos que no fueron modificados, Git guarda un enlace a un snapshot anterior.



Otro punto importante, Git gestiona sus snapshots a partir de commits, las cuales las mismas se pueden gestionar mediante branchs o ramas. Además en Git se puede realizar operaciones entre ramas, las principales son la fusión o merge y rebase.



Por ejemplo, en la figura mostrada se muestra una aplicación de ejemplo sobre la creación de nuevos branches para cada punto o módulo temático del proyecto, esto le permite a los desarrolladores de cada branch realizar su trabajo de forma separada para poder integrarlos cuando terminen su trabajo.

Por último punto a destacar, Git es gratis y open-source, es decir, es un sistema que se puede utilizar sin ninguna licencia comprada y que tiene una amplia comunidad de desarrolladores que brindan nuevas funciones al sistema o apoyo a los nuevos usuarios o novatos del sistema. Con esto, la instalación de Git no sería una preocupación económica fuerte para una organización, lo más importante sería enfocarse en brindar capacitación a los integrantes de la organización para que puedan aprovechar al máximo las funcionalidades que presenta Git, con el objetivo de realizar más efectivo sus tareas correspondientes.

## Calendario (Anthony Sandoval Salinas)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NOMBRE DE LAS ACTIVIDADES** | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **ROL** |
| **Planificación de la SCM** |  |  |  |  |  |  |  |  |
| Identificar la problemática de la empresa. |  |  |  |  |  | **X** |  | Gestor |
| Identificar roles y responsabilidades. |  |  |  |  |  | **X** |  | Gestor |
| Identificar procedimientos. |  |  |  |  |  | **X** |  | Gestor |
| **IDENTIFICACIÓN DE LA SCM** |  |  |  |  |  |  |  |  |
| Definir estructura del repositorio. |  | **X** |  |  |  |  |  | Bibliotecario |
| Definir la nomenclatura de los ítems. |  | **X** |  |  |  |  |  | Gestor |
| Definir herramientas de control de documentos y versionamiento que se utilizarán. |  | **X** |  |  |  |  |  | Gestor |
| Establecer las líneas bases. |  | **X** |  |  |  |  |  | Gestor |
| **CONTROL DE LA SCM** |  |  |  |  |  |  |  |  |
| Definir el formato de solicitud de cambio. |  |  |  |  |  |  | **X** | Gestor |
| Definir el flujo del proceso de control de cambios. |  |  |  |  |  |  | **X** | CCC |
| Crear plan de gestión de cambios. |  |  |  |  |  | **X** |  | Gestor |
| **ESTADO DE LA SCM** |  |  |  |  |  |  |  |  |
| Diseñar consultas y reportes. |  |  |  |  |  |  | **X** | Gestor |
| **AUDITORÍA DE LA GCS** |  |  |  |  |  |  |  |  |
| Diseñar consultas y reportes para la auditoría física. |  |  |  |  |  |  | **X** | Gestor |
| Diseñar consultas y reportes para la auditoría funcional. |  |  |  |  |  |  | **X** | Gestor |
| **GESTIÓN DE ENTREGA Y RELEASES** |  |  |  |  |  |  |  |  |
| Diseñar el proceso de pase a producción. |  |  |  |  |  |  |  | Gestor |
| Diseñar estructura del repositorio para los release de clientes. |  |  |  |  |  |  |  | Gestor |

# Actividades de la SCM

## Identificación de la configuración

* División del repositorio

La disposición del repositorio se encuentra de la siguiente manera:

* + Baseline
    - Proyecto
      * Baseline
  + Client
  + Development
    - Proyectos
      * Analysis
      * Business
      * Deployment
      * Management
      * Requirements
      * Sources
      * Test
  + Documents
    - Guidelines
    - Plans
    - Policity
* Nomenclatura

La nomenclatura para los documentos se mostrará con el siguiente ejemplo:

* + Para documentos de la empresa:

Indie Works plan SQA = IW-PSQA

* + Para documentos del proyecto:

Plan de gestión de la configuración de Software= CLPS-PGCS

* Documentos de la empresa

Cuenta con los siguientes documentos:

|  |  |
| --- | --- |
| Documentos | id |
| Plan de gestión de la Configuración de Software | CLPS-PGCS |
| Plan SQA | CLPS-PSQA |

* Documentos para proyectos

Cada proyecto contará con los siguientes documentos:

|  |  |
| --- | --- |
| Documentos | id |
| Documento del perfil de proyecto | IW-PP |
| Documento del negocio | IW-DN |
| Cronograma de actividades | IW-CP |
| Documento de especificación de casos de uso | IW-ECU |
| Documento de tecnologías y herramientas | IW-TH |
| Documento de diseño de software | IW-DDS |
| Documento de Base de Datos | IW-DBD |
| Código fuente del proyecto | IW-CFP |
| Plan de verificación y validación | IW-PVV |
| Manual de instalación del Sistema | IW-MIS |

## Control de la Configuración

**Control de Solicitud de Cambios:**

* Personal

Los siguientes roles son con los que se interactuará para el control de cambios del sistema:

|  |  |
| --- | --- |
| NombreRol | Descripción |
| Dueño del proceso | El jefe o personal encargado del area en el que se realiza el proceso |
| Autor | Integrante del área del proceso que quiere realizar un cambio en el sistema |
| Asistente 1 | Primer evaluador de formato de la solicitud de cambio y clasificador de este mismo |
| Asistente 2 | Segundo evaluador que tiene la tarea de clasificar las solicitudes de cambio |
| CCC | Evalua el impacto y riesgo de los cambios solicitados. Está conformado por especialistas de diversas áreas |
| Jefe de proyecto | Responsable de guiar el desarrollo al equipo del proyecto |
| Personal Responsable | Integrante del equipo de desarrollo responsable de la implementación correcta del cambio |
| Usuario Verificador | Cualquier usuario dentro del area donde se realiza el proceso |

* Estructura de Solicitud de cambio

|  |  |  |
| --- | --- | --- |
|  | descripción | Formato |
| ID | Código de la solicitud | [PROYECTO]-SC[numero] |
| Fecha | Fecha de creación de la solicitud | dd/mm/aa |
| Sistema | Sistema al cual se le desea realizar el cambio. | Código del documento aceptado |
| Fuente | Dueño del proceso - persona que autoriza el cambio | Nombre- rol |
| Autor | Persona que crea la solicitud. | Nombre- rol |
| Descripción | Breve descripción de la solicitud de cambio | texto |
| Justificación | Explicación de los motivos por el cual se desea realizar el cambio | texto |

* Estado de Solicitud de Cambio

Se establecieron 16 estados para el control de las solicitudes de cambios y son los siguientes:

|  |  |  |
| --- | --- | --- |
| **ID** | **Nombre del estado** | **Descripción** |
| 1 | Creado | El usuario ha creado la solicitud de cambios. |
| 2 | Recibido | El sistema de gestión de cambios ha recibido la solicitud de cambios. Terminando en la bandeja de entrada del asistente 1 |
| 3 | En revisión | El asistente 1 de gestión de cambios revisa la descripción y justificaciones esta ok, cambia el estado a "Revisado", sino cambia a rechazado-etapa 1 |
| 4 | Rechazado - Etapa 1 | La solicitud de cambio no especifica la descripción y justificación |
| 5 | Aprobado - Etapa 1 | La solicitud de cambio especifica correctamente los campos solicitados y es derivado a asistente 2 |
| 6 | Clasificado | La solicitud es clasificada por el asistente 2 según sea el tipo al que pertenezca el cambio y es derivado al CCC |
| 7 | Evaluado | El CCC ha evaluado el impacto y los riesgos del cambio |
| 8 | Rechazado - Etapa 2 | El cambio es rechazado por el dueño del proceso luego de la presentación del impacto y riesgo |
| 9 | Aprobado - Etapa 2 | La solicitud de cambio ha sido aprobada por el dueño del proceso luego de la presentación de impacto y riesgo |
| 10 | Planificado | El jefe de proyecto asignado al cambio ha realizado la planificación de actividades de desarrollo que se describe en la solicitud de cambio. |
| 11 | En construcción | El cambio se encuentra en la fase de construcción realizada por el equipo de desarrollo |
| 12 | Implementado | Se ha implementado la solicitud de cambio. |
| 13 | En verificación | El cambio implementado se encuentra siendo evaluado por el usuario para visualizar su conformidad con el mismo |
| 14 | Verificado | El usuario ha verificado la funcionalidad del cambio. |
| 15 | Rechazado - Etapa 3 | El usuario no está conforme con los cambios implementados en el sistema |
| 16 | Terminado | La solicitud de cambio se da por terminado. |

* Campos de Solicitudes

Una solicitud de cambio contará con los siguientes campos:

|  |  |
| --- | --- |
| ID | Identificador de la solicitud de cambio |
| Fecha | Fecha de creación de la solicitud |
| Sistema | Sistema al que va dirigido la solicitud de cambio. |
| Fuente | Jefe o director del área que esta está solicitando un cambio |
| Autor | Integrante del área que solicita el cambio. |
| Descripción | Una descripción en detalle del cambio que se quiere |
| Justificación | La razón por la cual se quiere realizar el cambio |
| EstadoID | ID del estado en el que se encuentra la solicitud de cambio |
| Fecha de Recepción de la Solicitud Cambio | Fecha en la que la solicitud de cambio es recibida por el asistente 1 para la primera revisión |
| Fecha de Revisión de la Solicitud Cambio | Fecha en la que se realiza la primera revisión por el asistente 1. |
| Fecha de Clasificación de la Solicitud Cambio | Fecha en la que se realiza la clasificación por el asistente 2. |
| CCC(ID) | ID del comité de control de cambios que evaluará la solicitud del cambio |
| Fecha de creación de CCC | Fecha en el que se creó el comité |
| Fecha de inicio de Evaluación (Impacto y riesgo) | Fecha en la que el CCC asignado inicia la evaluación de impacto y riesgo de la solicitud |
| Fecha de Fin de Evaluación (Impacto y riesgo) | Fecha en la que el CCC asignado finaliza la evaluación de impacto y riesgo de la solicitud |
| Tipo de impacto (ID) | Identificador del tipo de impacto que tiene la solicitud de cambio |
| Nivel de riesgo | Magnitud de posibles errores o daños que puede ocurrir luego de implementar los cambios |
| F. de inicio de planificación | Fecha en la que se inicia con la planificación del cambio |
| F. de fin de planificación | Fecha en la que se termina con la planificación del cambio |
| Persona responsable de la atención (Desarrollo) | Integrante del equipo de desarrollo responsable del cambio |
| Fecha inicio de implementación | Fecha en que se inicia la implementación del cambio en el programa |
| Fecha final de implementación | Fecha en la que se culmina la implementación del cambio en el programa |
| F. de verificación | Fecha donde se lleva a cabo la revisión de los cambios por parte del usuario |
| Usuario que verificó | Usuario final que verificó si el cambio se realizó correctamente o no |
| F. de rechazo | Fecha donde sucede el rechazo de la solicitud |
| Motivo de rechazo | Se describe el motivo del rechazo |

## Estado de la Configuración

**Reportes que ayuden Gestor de la Configuración conocer el estado de la GCS**

|  |  |
| --- | --- |
| **ID** | 001 |
| **Alumno:** | Suarez Hernandez, Kenny Joel |
| **Título del reporte** | Lista de solicitudes de cambios del proyecto |
| **Propósito del reporte** | Proporcionar al Gestor de la Configuración un listado de todas las solicitudes de cambios realizados en el proyecto y así pueda llevar un registro y control de ello |
| **Entradas** | -Id del proyecto  -Rango de fecha |
| **Salidas** | -Lista de cambios   * Id de solicitud de cambio * Fecha * Sistema * Fuente * Autor * Descripción * Justificación |

|  |  |
| --- | --- |
| **ID** | 002 |
| **Alumno:** | Antúnez Palomino, Kori Xiomara |
| **Título del reporte** | Lista de cambios del proyecto |
| **Propósito del reporte** | Proporcionar al Gestor de la Configuración un listados de todos los cambios realizados en el proyecto y así pueda llevar un registro y control de ello |
| **Entradas** | -Id del proyecto  -Id autor  -Id Branch  -Rango de fecha |
| **Salidas** | -Tipo de Cambio (mejora, error, adicionar)  -Lista de cambios   * Código de cambio * Título de cambio * Descripción de cambio * Nombre del branch * Fecha de creación |

|  |  |
| --- | --- |
| **ID** | 003 |
| **Alumno:** | Vilca Daza, Diego Percy |
| **Título del reporte** | Lista de autores que realizaron cambios en una rama o Branch |
| **Propósito del reporte** | Proporciona la información respectiva de los autores que aplicaron cambios sobre un Branch con el fin de verificar si estos tenían autorización para modificarlo. |
| **Entradas** | -          ID del proyecto  -          ID del Branch |
| **Salidas** | -          Nombre del Branch  -          Fecha de creación  -          Lista de cambios  -          Nombre del cambio  -          Versión del cambio  -          Fecha de modificación  -          ID del autor  -          Nombre del autor  -          Comentarios |

|  |  |
| --- | --- |
| **ID** | 004 |
| **Alumno:** | Sandoval Salinas, Anthony Kevin Jerremy |
| **Título del reporte** | Lista de autores colaboradores de Primera Línea Base |
| **Propósito del reporte** | Proporcionar una lista con información de los autores de una línea base para poder elaborar un reporte de justificación de modificación de dicha línea base. |
| **Entradas** | * Id del Proyecto * Id del Branch * Id del Commit |
| **Salidas** | * Nombre del Commit * Fecha del Commit * Lista de autores * Nombre del autor * Rol de proyecto * Historial de commits asociados |

* **Reportes que ayuden al equipo de desarrollo de software conocer el  estado de la GCS**

|  |  |
| --- | --- |
| **ID** | 005 |
| **Alumno:** | Gomez Cavero, Mishell |
| **Título del reporte** | Detalle de los entregables de la liberación del proyecto |
| **Propósito del reporte** | Proporcionar al equipo de desarrollo la información sobre el contenido de los entregables que  que se ha dado en la liberación del producto desarrollado, y así poder llevar un registro de ello. |
| **Entradas** | * ID Proyecto * Número de versión del proyecto Release * ID entregable |
| **Salidas** | * Nombre del entregable * Descripción del entregable * Autor del entregable * Fecha de liberación * Ruta de ubicación del entregable * Contenido del entregable |

|  |  |
| --- | --- |
| **ID** | 006 |
| **Alumno:** | Suarez Hernandez, Kenny Joel |
| **Título del reporte** | Reporte de resultados de pruebas de verificación y aceptación |
| **Propósito del reporte** | Proporcionar al equipo de desarrollo la información acerca sobre el estado del producto desarrollado y si ha cumplido con las pruebas de aceptación y verificación establecidas para dicho sprint. |
| **Entradas** | * ID del proyecto * Rango de fechas |
| **Salidas** | * ID de prueba * Fecha * Semana * Resultado |

* **Reportes que ayude jefe de proyecto de desarrollo de software conocer el  estado de la GCS**

|  |  |
| --- | --- |
| **ID** | 007 |
| **Alumno:** | Aguilar Salazar, Edwin Aguilar  Palomino Gutierrez, Erick Victor |
| **Título del reporte** | Lista de Requerimientos a ser modificados |
| **Propósito del reporte** | Dar a conocer al Jefe de proyecto sobre las modificaciones, agregaciones y eliminaciones a los requerimientos del proyecto. |
| **Entradas** | ID Proyecto  ID Solicitud de Cambio  ID Requerimiento |
| **Salidas** | * Requerimientos a modificar:   + Especificación actual del requerimiento.   + Descripción del cambio. * Requerimientos a agregar.   + Descripción del cambio. * Requerimientos eliminados. |

## Auditoría de la Configuración

En este apartado se identifica los elementos del sistema que satisfacen los requerimientos del cliente. Según el SWEBOK existen dos tipos de auditoria: la física y la funcional.

La auditoría funcional verifica si parte del sistema cumple con las especificaciones que rigen el producto. La auditoría física se asegura que la documentación y diseño del software sea igual al sistema construido.

## Gestión de entrega y Release de Software

Construcción del software: Es la combinación entre los elementos y datos de configuración para crear un ejecutable del software y de esta manera entregarlo al cliente.

Gestión del release del software: Es la identificación, empaquetado y entrega del software. Sin embargo, este puede cambiar luego de ser analizado y encontrar las partes del software que deberán ser cambiadas para lanzar un nuevo release.