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) **3**

[Situación de la empresa (Erick Palomino)](#_6g7xiha4ycwk) 3

[Propósito (Mishell Gomez)](#_sbd8ej1uroj2) 3

[Alcance (Kori Antunez)](#_pqe5g1i42278) 3

[**Gestión de la SCM**](#_x5uo1f7xn1r) **4**

[Roles o responsabilidades](#_u5xp69v7v23p) 4

[Procedimientos (Edwin Aguilar)](#_vtl97i4ioiak) 4

[Herramientas](#_rdcf7necz5qo) (Diego Vilca y Kenny Suarez) 4

[Calendario (Anthony Sandoval)](#_fse0b2fc9zxs) 7

[**Actividades de la SCM**](#_6snxqp1b4j2r) **9**

[Identificación de la configuración](#_urvi0793d11g) 9

[Control de la Configuración](#_wi6zqgp1sie5) 9

[Estado de la Configuración](#_8qreciv67nrt) 9

[Auditoría de la Configuración](#_bhqtn24j648k) 9

[Gestión de entrega y Release de Software](#_aln1a558n9k3) 9

# **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.

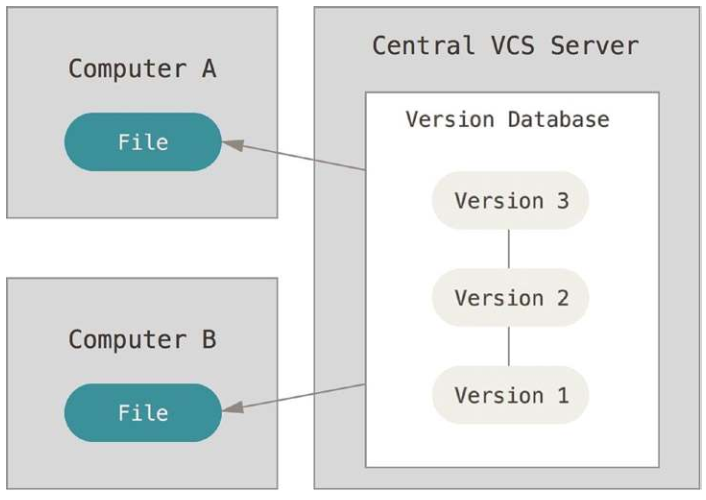
Los servicios que ofrece el modo free son:

* + 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)

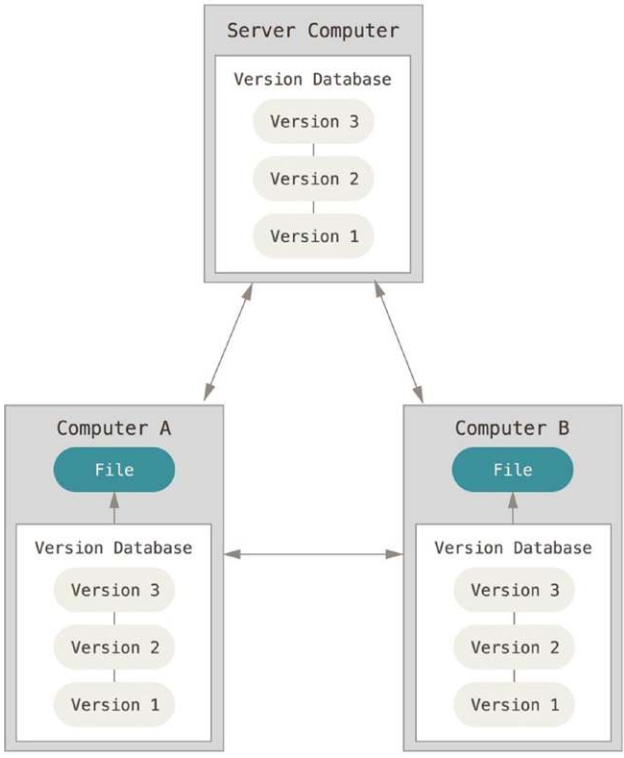
Es un sistema de control de versiones distribuido, es decir, gestiona diversos cambios que se realizaron en un elemento de configuración, esto se logra mediante el registro de los cambios en una base de datos de versiones. Esto le permite al sistema por ejemplo poder dirigirse a una versión específica.

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.

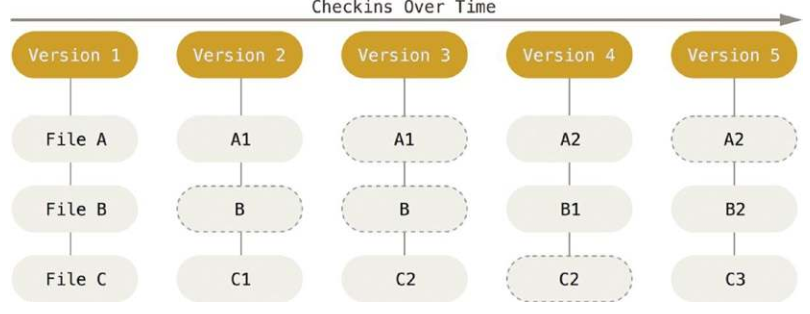
De lo contrario, un sistema distribuido no solo permite a los desarrolladores extraer una versión de un archivo o un conjunto de archivos, sino también permite extraer el historial de versiones de esos archivos.



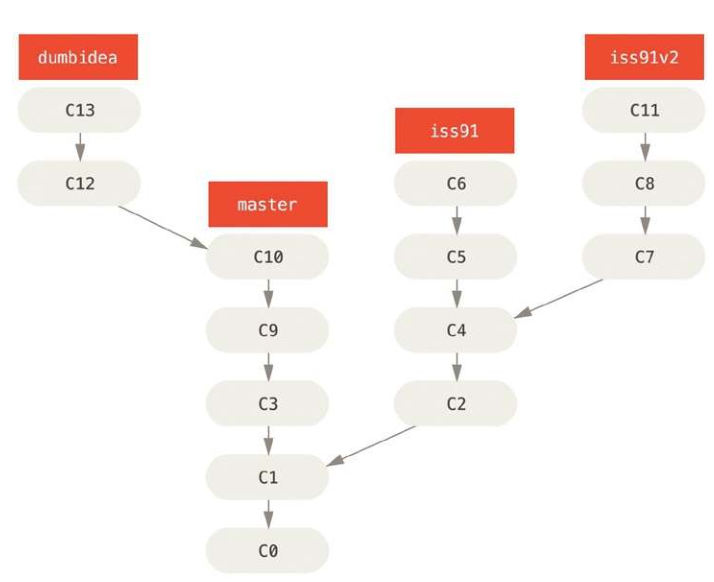
Debido a que como todo el historial se encuentra en un punto central en el sistema centralizado, esto hace que la información de las versiones se puedan perder en caso de un error o caída y no se pueda recuperar.

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**

## Control de la Configuración

## Estado de la Configuración

## Auditoría de la Configuración

## Gestión de entrega y Release de Software