**Gestión de la configuración de software**

**Empresa Tesla Tech**

**ÍNDICE**

[**1. Introducción 3**](#_heading=h.f9lx5fz8p2zk)

[1.1. Situación de la empresa 3](#_heading=h.xpczr856ofqp)

[1.2. Problemática 3](#_heading=h.bmmq3ogjlytd)

[1.3. Objetivo 3](#_heading=h.5415xz7ejlto)

[**2. Gestión de la SCM 4**](#_heading=h.d1c5fj8bp4zg)

[2.1. Roles de la GCS 4](#_heading=h.1mb1x2pcdoy)

[2.2. Herramientas 4](#_heading=h.gd9063b6wxtt)

[2.3. Diagrama de arquitectura de la herramienta elegida 6](#_heading=h.68c29lnio1s9)

[**3. Actividades de la SCM 7**](#_heading=h.j4pwzvx6g8vm)

[3.1. Clasificación de ítems 7](#_heading=h.b9nozzd2yku7)

[3.2. Definición de nomenclatura 7](#_heading=h.pc2abo3nmamd)

[3.3. Diseño del repositorio 9](#_heading=h.650flzvmd4p2)

[3.4. Definición de la Línea Base 10](#_heading=h.49y19tury5dv)

# **Introducción**

## Situación de la empresa

TeslaTech es una empresa con 5 años de experiencia especializada en proyectos educativos para instituciones educativas, universidades, colegios, academias y bibliotecas. Nos dedicamos a proporcionar soluciones tecnológicas innovadoras para mejorar los procesos educativos y de gestión en entornos académicos.

Actualmente contamos con tres proyectos significativos a diversos clientes, como por ejemplo: la Universidad Nacional Mayor de San Marcos, la Universidad Tecnológica del Perú y el colegio Pamer.

## Problemática

Actualmente, nos enfrentamos a desafíos relacionados con el control de versiones en nuestros proyectos. La falta de una gestión adecuada de versiones ha resultado en problemas como la sobreescritura de código, la pérdida de cambios importantes. la entrega incorrecta de una versión errónea a uno de nuestros clientes y la falta de seguimiento de las modificaciones realizadas por los miembros del equipo.

## Objetivo

El objetivo principal de este plan es aplicar las buenas prácticas de Gestión de Configuración de Software (GCS) para abordar las problemáticas identificadas en el control de versiones y la gestión de cambios.

Al implementar un proceso sólido de GCS, buscamos mejorar la calidad, la confiabilidad y la eficiencia de nuestros proyectos educativos, garantizando la integridad y la trazabilidad de los artefactos de software y documentación asociados. Además, buscamos establecer un marco de trabajo que promueva la colaboración, la comunicación y la coordinación entre los miembros del equipo, facilitando así el desarrollo y la entrega exitosa de nuestros productos y servicios.

# Gestión de la SCM

## Roles de la GCS

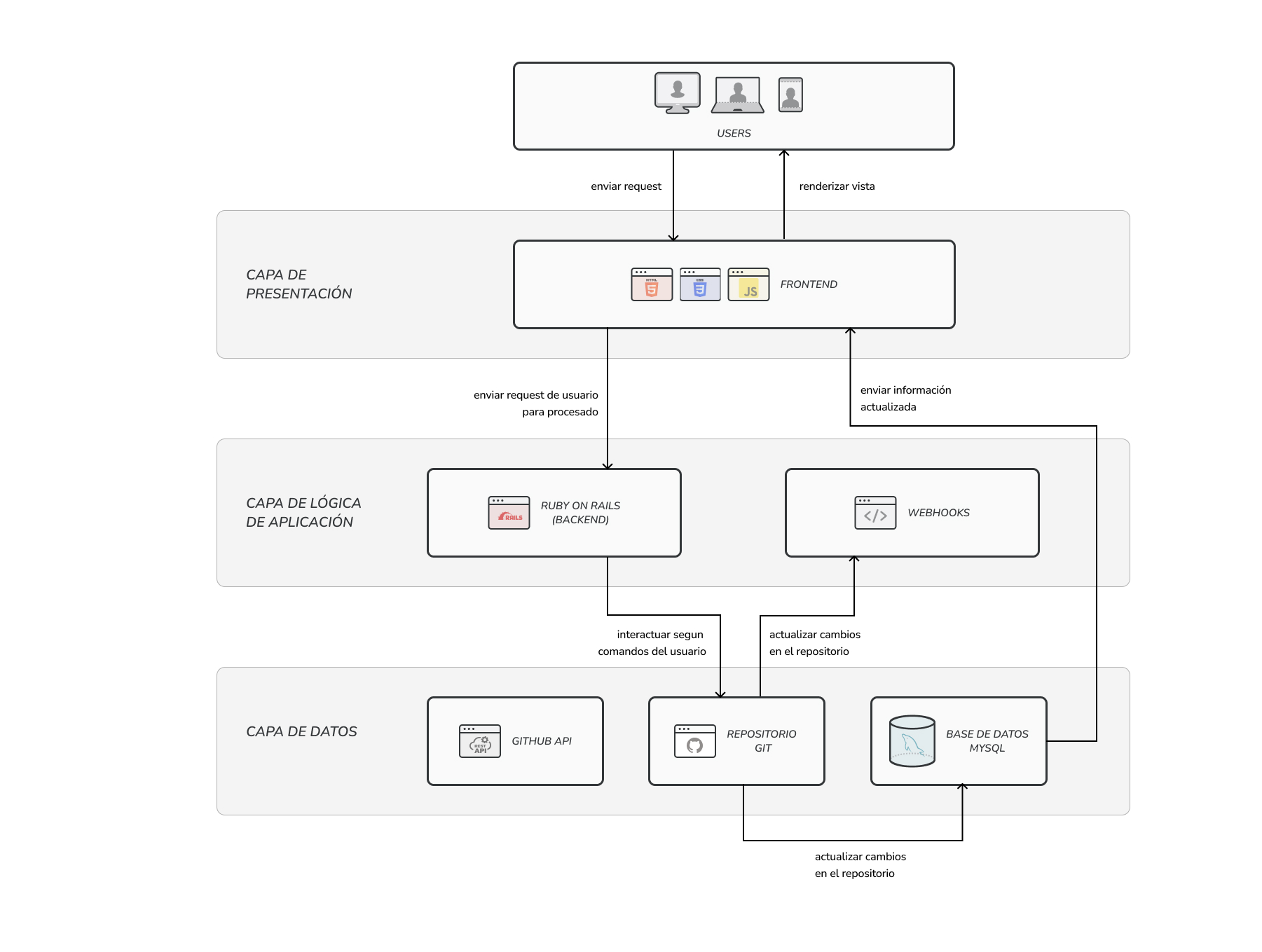
| Nombre de rol | Persona asignada | Responsabilidades | Niveles de autoridad |
| --- | --- | --- | --- |
| Project  Manager | Josué Carbajal | Supervisar el funcionamiento de la Gestión de la Configuración. | Toda autoridad sobre el proyecto y sus funciones. |
| Gestor de configuración | Sebastian Ayala | Ejecutar todas las tareas de gestión de la configuración. | Autoridad para operar las funciones de Gestión de la configuración. |
| Inspector de Aseguramiento de Calidad | Braulio Saldaña | Auditar la Gestión de la Configuración. | Auditar la Gestión de la Configuración según indique el Project Manager. |
| Control de cambios | Jean Lavaud | Evaluar y revisar las propuestas de cambios en el proyecto. | Autoridad para aprobar o rechazar solicitudes de cambio en el proyecto. |
| Comité de GCS | Denilson Morales  Leslie  Yazid Arroyo  Bruno Chochoca | Establecer el procedimiento de la gestión de la configuración en la organización. Revisar y aprobar cambios en las líneas base. | Autoridad para tomar decisiones respecto a la gestión de configuración de software y establecer procedimientos para los proyectos. |

## Herramientas

| Herramientas | Logo | Fuente | Características |
| --- | --- | --- | --- |
| GitHub |  | <https://github.com/> | - Plataforma de desarrollo colaborativo basada en Git.  - Gestión eficiente de versiones de código. |
| GitLab |  | <https://about.gitlab.com/> | - Plataforma completa de ciclo de vida de desarrollo de software (SDLC) basada en Git.  - Flexibilidad para ser implementado en la nube o en entornos locales. |
| GitKraken |  | <https://www.gitkraken.com/> | - Herramienta visual para Git que simplifica operaciones de control de versiones.  - Interfaz gráfica intuitiva para representar ramas, fusiones y confirmaciones. |
| Mantis Bug Tracker |  | <https://mantisbt.org/> | -Sistema de seguimiento de errores basado en web de código abierto.  -Gestión y colaboración ágiles desde el inicio con su interfaz intuitiva. |
| Jira |  | <https://www.atlassian.com/es/software/jira> | -Jira proporciona tableros ágiles como Scrum y Kanban para gestionar proyectos de software de manera eficiente.  -Centraliza información, automatiza flujos de trabajo. |
| Bugzilla |  | <https://www.bugzilla.org/> | -Permite planificar, organizar y lanzar software según el cronograma de tu propio equipo.  -Interfaz web intuitiva, altamente personalizable, segura y se integra con otras herramientas de desarrollo. |

## Diagrama de arquitectura de la herramienta elegida

A continuación se muestra el diagrama de arquitectura de la herramienta de control de versiones elegida por el equipo: Github.



Elegimos GitHub como nuestra herramienta de control de versiones debido a su facilidad de uso, amplia adopción en la industria y su robusto sistema de control de versiones basado en Git.

La plataforma nos ofrece no solo características de colaboración efectivas, sino también integraciones con otras herramientas clave, una comunidad activa de desarrolladores y sólidas medidas de seguridad, demostrando por qué es una de las herramientas de versionamiento más usadas en el mercado actual.

Como equipo confiamos en que GitHub facilitará la colaboración entre las distintas áreas de la empresa, mejorará la gestión de nuestro código y contribuirá al éxito de nuestros proyectos de desarrollo, permitiéndonos mantener nuestros estándares de calidad con los clientes que depositaron su confianza en nuestra empresa.

# Actividades de la SCM

## Clasificación de ítems

| Tipo | Ítem | Extensión | Proyecto |
| --- | --- | --- | --- |
| E | Project Charter | .docx | BF |
| E | Cronograma de proyecto | .xlsx | BF |
| E | Documento de negocio | .docx | BF |
| E | Requisitos funcionales | .xlsx | BF |
| E | Requisitos no funcionales | .xlsx | BF |
| E | Especificación de cada caso de uso | .docx | BF |
| E | Planificación | .docx | BF |
| F | Diseño web | .html | BF |
| F | Estilos del diseño web | .css | BF |
| F | Parte principal del backend del proyecto (main) | .py | BF |

## Definición de nomenclatura

Regla N°1:

La nomenclatura para cada ítem de cada proyecto es el siguiente:

| *“Acrónimo del proyecto”+”-”+”Acrónimo de item” + ext* |
| --- |

Cada acrónimo deberá estar en mayúscula

Ejemplo:

Proyecto: QR Cakes

Ítem: Project Charter

Extensión: *.docx*

*“QRC-PC.docx”*

Regla N°2:

En el acrónimo de cada ítem, la nomenclatura no tendrá en cuenta si el nombre del ítem tiene alguna conjunción o preposición.

Ejemplo:

Ítem: Cronograma de proyecto

Extensión: .xlsx

Proyecto: Tuku Fermix

*“TF-CP*.xlsx*”*

Regla N°3:

En el caso de que dos o más ítems, tengan el mismo acrónimo y se encuentren en la misma carpeta, estos modificarán su acrónimo de item, aumentando, la segunda consonante de la última palabra.

Ejemplo:

Proyecto: Biblio F-easy

Ítem 1: Project Charter

Extensión: .xlsx

*“BF-PCH*.xlsx*”*

Ejemplo:

Ítem 2: Planificación de Calidad

Extensión: .xlsx

Proyecto: Biblio F-easy

*“BF-PCL*.xlsx*”*

Regla N°4:

En el caso de que un ítems, contenga varios archivos, se tendrá que enumerar.

Ejemplo:

Proyecto: Biblio F-easy

Ítem 1: Documento de especificación de caso de uso

Extensión: *.docx*

*“BF-DECU-1.docx”*

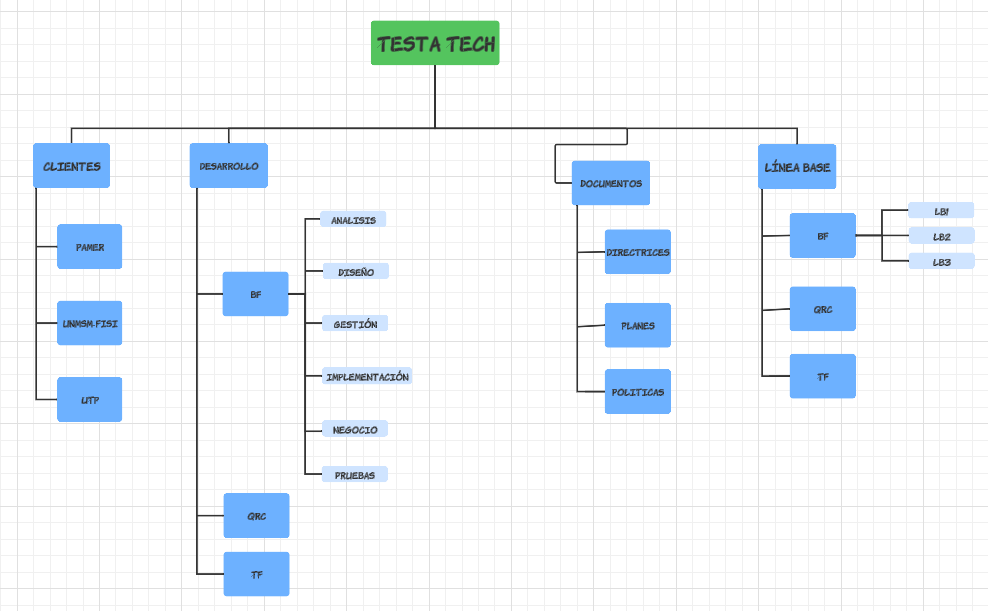
*“BF-DECU-2.docx”*

*“BF-DECU-3.docx”*

*“BF-DECU-4.docx”*

*…*

## Diseño del repositorio



## Definición de la Línea Base

Para mostrar la organización de las líneas base se tomará como ejemplo nuestro proyecto en desarrollo Biblio F-easy (BF), con fecha de inicio el 12 de abril de 2024 y con fecha de finalización el 7 de julio de 2024.

| Línea Base | Hito | Artefactos |
| --- | --- | --- |
| LB1 | 1 (19/05/2024) | Project Charter  Cronograma del proyecto  Lista de requisitos funcionales  Lista de requisitos no funcionales  Documento de negocio  Doc. Esp. Caso de uso 01  Doc. Esp. Caso de uso 02  Doc. Esp. Caso de uso 03  Doc. Esp. Caso de uso 04  Doc. Esp. Caso de uso 05  Doc. Esp. Caso de uso 06  Doc. Esp. Caso de uso 07  Doc. Esp. Caso de uso 08  Doc. Esp. de interfaz de usuario  Doc. de guía de estilos  Doc. Esp. Base de datos  Doc. de arquitectura de software  Reporte del desarrollo del software  Reporte del primer sprint |
| LB2 | 2 (07/06/2024) | Doc. Esp. Caso de uso 01  Doc. Esp. Caso de uso 02  Doc. Esp. Caso de uso 03  Doc. Esp. Caso de uso 04  Doc. Esp. Caso de uso 05  Doc. Esp. Caso de uso 06  Doc. Esp. Caso de uso 07  Doc. Esp. Caso de uso 08  Doc. Esp. de interfaz de usuario  Doc. Esp. Base de datos  Reporte del desarrollo del software  Reporte del segundo sprint |
| LB3 | 3 (06/07/2024) | Doc. Esp. Caso de uso 01  Doc. Esp. Caso de uso 02  Doc. Esp. Caso de uso 03  Doc. Esp. Caso de uso 04  Doc. Esp. Caso de uso 05  Doc. Esp. Caso de uso 06  Doc. Esp. Caso de uso 07  Doc. Esp. Caso de uso 08  Doc. Esp. UI  Doc. Esp. Base de datos  Doc. Esp. Arquitectura de software  Doc. Esp. Diseño del software  Doc. de guía de estilos  Manual de usuario  Doc. de pruebas del software  Reporte del desarrollo del software  Reporte del tercer sprint |