**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**“Universidad del Perú. Decana de América”**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**E.P. INGENIERÍA DE SOFTWARE**



**“Plan de Gestión de la Configuración”**

**Empresa: AETHER TECH**

**Grupo 2**

**Docente:**

Wong Portillo, Lenis Rossi

**Integrantes:**

* Davalos Benito Rodrigo
* Limachi Sarmiento, Jose Luis
* Meza Torres, Joseph Omar
* Panta Labán, Leonardo Justo
* Uscamayta Sanchez, Gabriel Omar
* Villanueva Aguirre, Cesar Alexander

**2025 - I**

# Introducción

## Situación de la empresa

AETHER\_TECH es una empresa innovadora que ofrece soluciones tecnológicas de alto nivel, ayudando a empresas y emprendedores a llevar sus proyectos al siguiente nivel. Fundada en 2019, nuestra empresa actualmente cuenta con tres proyectos en producción: un Sistema de Comercio Electrónico potenciado con IA para tiendas Falabella, un Sistema de Gestión de Distribuidores y Ventas para Lindcorp, y un Sistema de Gestión de Citas Médicas para EsSalud. Asimismo, contamos con un proyecto en desarrollo: un Sistema de Gestión de Almacenes para tiendas Cencosud. Algunos problemas que hemos detectado acerca del control de versiones son la falta de documentación adecuada de las versiones, los cambios realizados y las razones detrás de ellos. Esto puede generar dificultades para comprender la forma en la que ha ido evolucionando el proyecto.

## Objetivo

Queremos implementar el plan de Gestión de Configuración para llevar un control claro y ordenado de los cambios que se realicen en el proyecto. Esto nos permitirá saber quién hizo cada modificación, en qué momento y con qué propósito. Además, ayuda a evitar errores por cambios mal gestionados, mejora el trabajo en equipo y nos asegura tener versiones estables del sistema a las que podamos volver si es necesario. En definitiva, es una forma de mantener el proyecto bien organizado y bajo control.

# Gestión

## Roles y Responsabilidades

| Roles | Responsabilidades |
| --- | --- |
| Project Manager | * Supervisa la aplicación del Plan de la Gestión de la Configuración en el proyecto. * Coordinar los recursos y tiempos para las actualizaciones aprobadas. |
| Gestor de Configuración | * Definir y aplicar políticas de control de versiones, líneas base y auditoría. * Garantizar el seguimiento de todos los ítems de configuración. |
| Comité de Control de Cambios | * Evaluar el impacto técnico, económico y operativo de cada cambio. * Revisar y aprobar/rechazar las solicitudes de cambio. |
| Desarrollador Backend | * Implementar y mantener el código fuente y la infraestructura del servidor. * Registrar y documentar los cambios realizados en el repositorio. |
| Desarrollador Frontend | * Desarrollar la interfaz gráfica de usuario (UI) y asegurar su integración con el backend. * Asegurar que los cambios estén versionados correctamente. |
| Administrador de Base de Datos | * Administrar y mantener la integridad, seguridad y rendimiento de la base de datos. * Controla y actualiza los scripts de estructura de datos. |
| Analista QA | * Realizar pruebas para verificar la calidad de los cambios antes de su liberación. * Reporta desviaciones o defectos encontrados tras los cambios. |

## Benchmarking de Herramientas, Gráfica de la infraestructura (de la herramienta elegida)

| Criterios | Git | Perforce | Mercurial | SVN |
| --- | --- | --- | --- | --- |
| Gratuito | Sí | No (licencia) | Sí | Sí |
| Curva de aprendizaje | Alta | Alta | Media | Baja |
| Centralizado | No | Si | No | Sí |
| Distribuido | Sí | Parcialmente | Sí | No |
| Enfocado a instantaneas | Sí | Sí | Sí | No |
| Enfocado a deltas | No | Sí | No | Sí |
| Plataformas | Multiplataforma | Multiplataforma | Multiplataforma | Multiplataforma |
| Comunidad | Alto | Media | Bajo | Media |

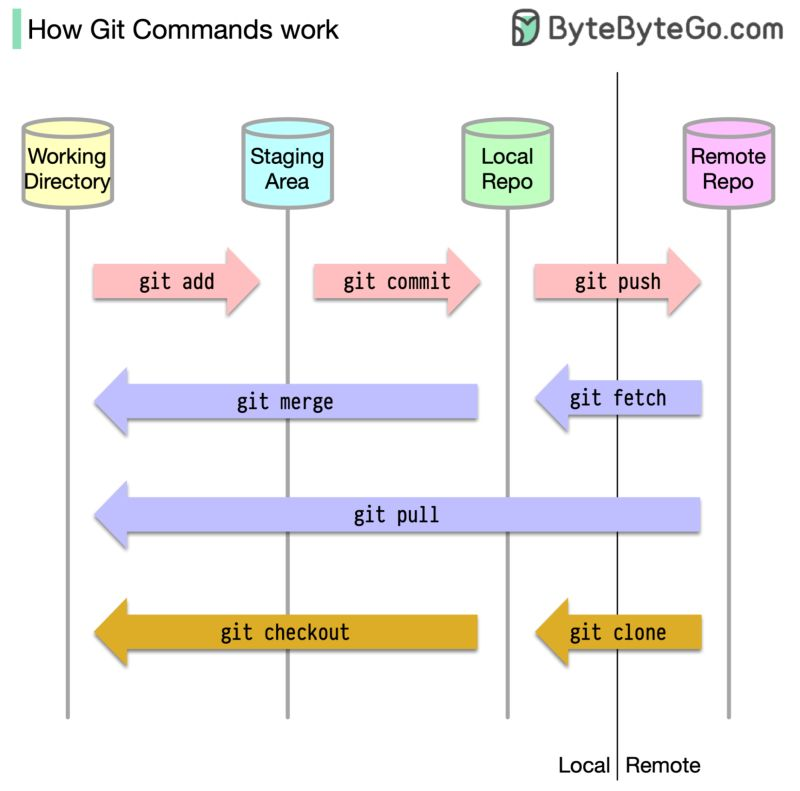
* **Justificación de la elección:**

Considerando la disponibilidad y las funcionalidades necesarias para el desarrollo de este proyecto, se ha optado por utilizar Git, ya que su enfoque distribuido facilita un control más detallado del versionado del software.

* **Herramienta seleccionada:** Git
* **Descripción:**

Git es un sistema de control de versiones distribuido que permite gestionar los cambios mediante el uso de instantáneas. Gracias a su arquitectura, cada colaborador cuenta con una copia completa del repositorio, lo que mejora la colaboración, permite trabajar sin conexión y ofrece mayor flexibilidad durante el desarrollo.

* **Arquitectura de Git:**



# Actividades de la GCS

### Identificación de la GCS

### Definir estructura de la librerías del repositorio

### Definición de Nomenclatura

ACRÓNIMO DEL PROYECTO + “-” + ACRÓNIMO DEL ELEMENTO + EXTENSIÓN

ACRÓNIMO DEL PROYECTO + “-” + SIGLAS DEL ELEMENTO + EXTENSIÓN

ACRÓNIMO DEL PROYECTO + “-” + ACRÓNIMO DEL ELEMENTO + N° + EXTENSIÓN

### Lista de Ítems

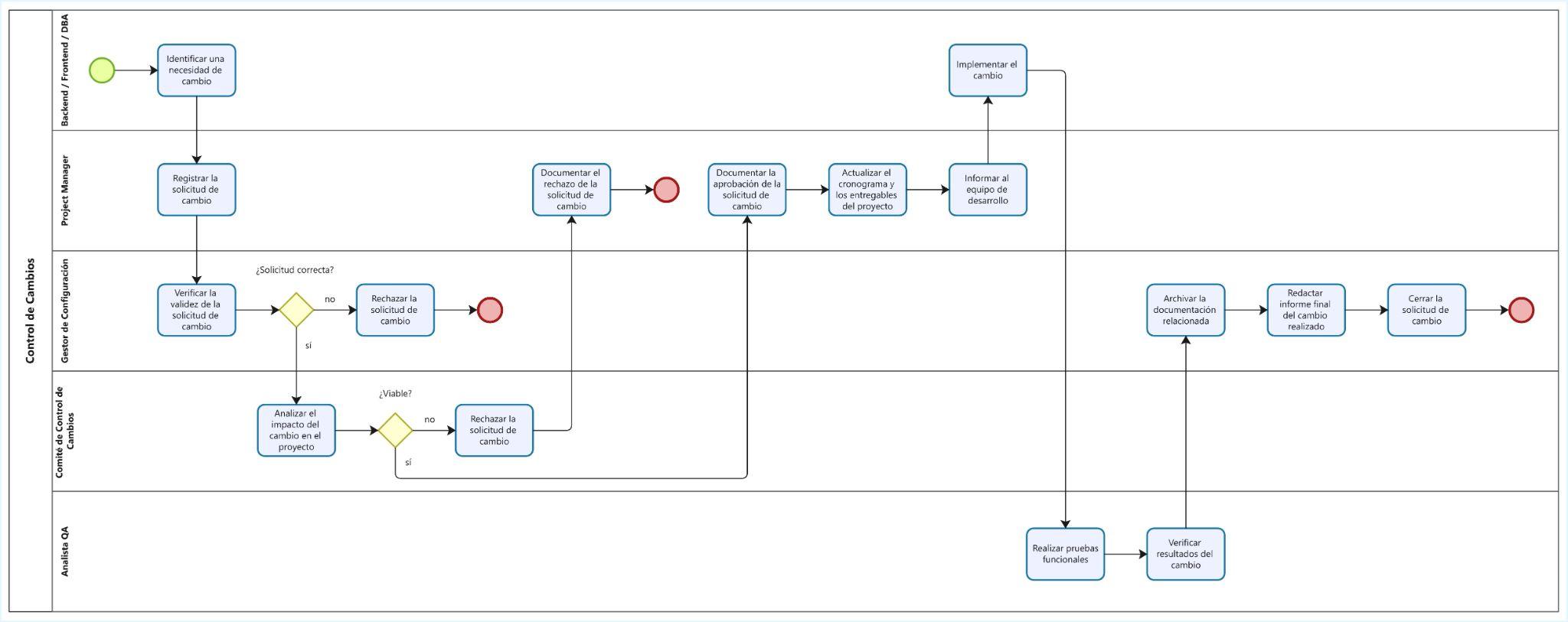
| Tipo ( E= Evolución F= Fuente S= Soporte) | Ítem | Nomenclatura | Proyecto |
| --- | --- | --- | --- |
| E | Documento de Negocio | SGATC-DN.docx | SGATC |
| E | Cronograma del Proyecto | SGATC-CP.xlsx | SGATC |
| E | Project Charter | SGATC-PC.docx | SGATC |
| E | Documento de Especificación de Requisitos | SGATC-DER.docx | SGATC |
| E | Requisito de Usuario N 01 | SGATC-RU01.docx | SGATC |
| E | Requisito de Usuario N 02 | SGATC-RU02.docx | SGATC |
| E | Requisito de Usuario N 03 | SGATC-RU03.docx | SGATC |
| E | Requisito de Usuario N 04 | SGATC-RU04.docx | SGATC |
| E | Requisito de Usuario N 05 | SGATC-RU05.docx | SGATC |
| E | Requisito de Usuario N 06 | SGATC-RU06.docx | SGATC |
| S | Git-2.32.0-64-bit.exe |  |  |
| S | Visual Studio Code | SGATC-VSC.rar | SGATC |

### 

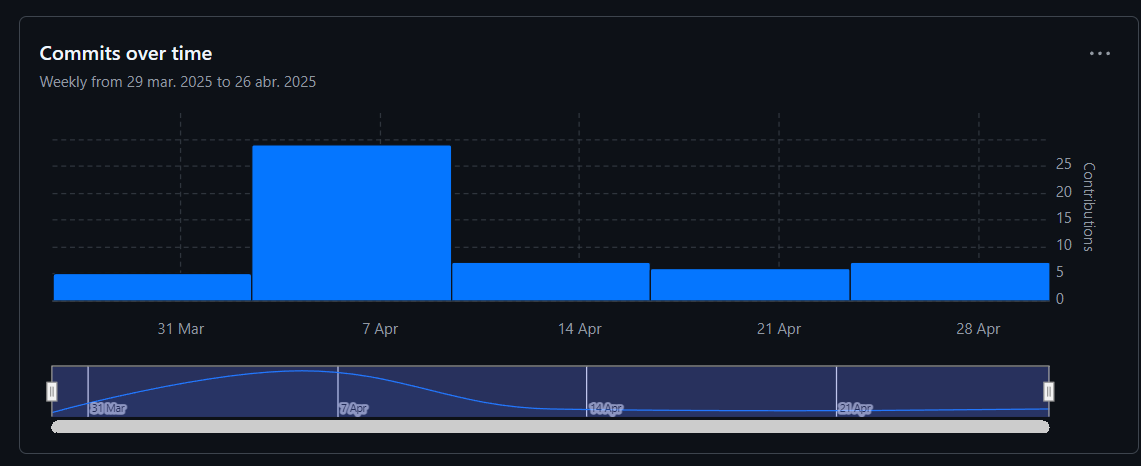
### Definicion de Linea Base

| Hito | Fecha | N° de Linea Base | Items |
| --- | --- | --- | --- |
| 1 | 01/05/25 | Linea Base 1 | Project Charter (SGATC-PC.xlsx)  Cronograma del Proyecto (SGATC-CP.docx)  Documento de Negocio (SGATC-DN.docx)  Lista de requisitos (SGATC-LR.docx)  Requisitos de usuario 01 (SGATC-RU01.docx)  Requisitos de usuario 02 (SGATC-RU02.docx)  Requisitos de usuario 03 (SGATC-RU03.docx)  Requisitos de usuario 04 (SGATC-RU04.docx)  Requisitos de usuario 05 (SGATC-RU05.docx)  Requisitos de usuario 06 (SGATC-RU06.docx) |
| 2 | 15/05/25 | Linea Base 2 |  |
| 3 | 19/06/25 | Linea Base 3 |  |

### Control de la GCS

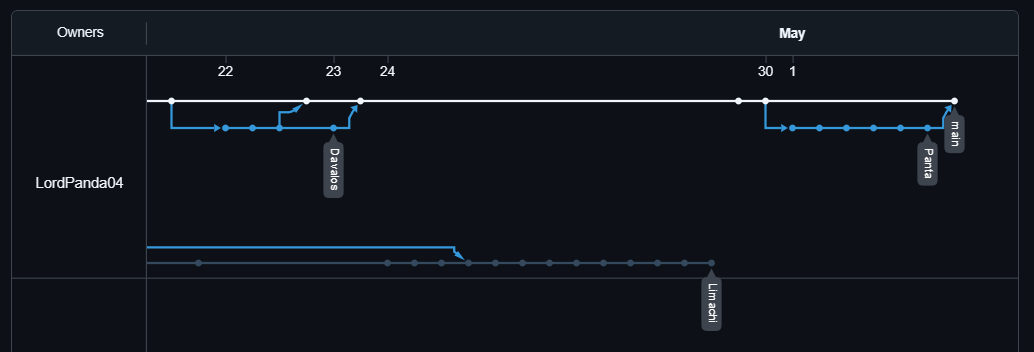
1. Ejemplos de Solicitudes de cambio (1 por alumno)
2. Proceso de Control de Cambios (De la Empresa SGATC)
3. Gráfica del proceso
4. Tipos de estados de la Solicitud de cambio

| Estado | Descripción |
| --- | --- |
| Propuesta | La solicitud ha sido registrada y presentada formalmente por un miembro del equipo. En este estado, aún no ha sido revisada ni evaluada. |
| En evaluación | La solicitud está siendo analizada por el Comité de Control de Cambios y/o el Project Manager para determinar su viabilidad técnica y su impacto en el proyecto |
| Aprobada | La solicitud ha sido revisada y aceptada, y puede pasar a la etapa de implementación según las prioridades del proyecto. |
| Rechazada | La solicitud ha sido evaluada y no será implementada, debido a falta de viabilidad o conflicto con los objetivos del proyecto. |
| En desarrollo | El cambio aprobado está siendo implementado por el equipo técnico. Se registran las modificaciones realizadas en el sistema de versiones. |
| En pruebas | El cambio desarrollado está siendo validado por el equipo de QA. Se ejecutan pruebas necesarias para asegurar que el cambio cumple con lo solicitado sin generar errores adicionales. |
| Desplegada | El cambio ha pasado exitosamente las pruebas y ha pasado a producción. Además, se actualiza la documentación correspondiente. |
| Cerrada | El ciclo de la solicitud de cambio ha finalizado formalmente. Todo el seguimiento ha sido documentado, y se archiva para fines de auditoría o revisión futura. |

1. Ejemplos de reportes (3)
2. Estados de la GCS

*Figura: Gráficos de contribución por integrante de commits del repositorio*

*Figura: Gráficos de contribución por integrante de commits del repositorio*

*Figura : Gráfico de flujo de ramas en el repositorio*

1. Auditoria de la GCS
2. Auditoria Fisica

| Nomenclatura | Ítem | Fecha de auditoria | Auditor | Estado:  Cumple (**✓**)  No cumple (✘) |
| --- | --- | --- | --- | --- |
| SGATC-LR | Lista de Requisitos | 03/05/25 | Wong Portillo, Lenis Rossi | **✓** |
| SGATC-DLN | Documento de Lógica de Negocio | 17/05/25 | Wong Portillo, Lenis Rossi | **✓** |
| SGATC-DAS | Documento de la Arquitectura de Software | 17/05/25 | Wong Portillo, Lenis Rossi | ✘ |
| SGATC-DMBD | Documento del Modelado de Base de Datos | 17/05/25 | Wong Portillo, Lenis Rossi | **✓** |
| SGATC-DUI | Documento de la User Interface | 17/05/25 | Wong Portillo, Lenis Rossi | **✓** |
| SGATC-BFR01 | Código backend y frontend | 21/06/25 | Wong Portillo, Lenis Rossi | **✓** |
| SGATC-BFR02 | Código backend y frontend | 21/06/25 | Wong Portillo, Lenis Rossi | ✘ |
| SGATC-BFR03 | Código backend y frontend | 21/06/25 | Wong Portillo, Lenis Rossi | **✓** |
| SGATC-DPS | Documento de Pruebas de Software | 21/06/25 | Wong Portillo, Lenis Rossi | **✓** |
| SGATC-MU | Manual de Usuario | 21/06/25 | Wong Portillo, Lenis Rossi | ✘ |

1. Auditoria Funcional

| Nomenclatura | Requisitos de Usuario | Fecha de auditoria | Auditor | Estado:  Cumple (**✓**)  No cumple (✘) |
| --- | --- | --- | --- | --- |
| SGATC-RU01 | Procesar pedidos de reabastecimiento | 03/05/25 | Wong Portillo, Lenis Rossi | **✓** |
| SGATC-RU02 | Gestionar usuarios y roles | 03/05/25 | Wong Portillo, Lenis Rossi | ✘ |
| SGATC-RU03 | Controlar productos perecibles | 03/05/25 | Wong Portillo, Lenis Rossi | **✓** |
| SGATC-RU04 | Registrar y auditar movimientos de inventario | 03/05/25 | Wong Portillo, Lenis Rossi | **✓** |
| SGATC-RU05 | Registrar, actualizar y eliminar productos en el inventario | 03/05/25 | Wong Portillo, Lenis Rossi | **✓** |
| SGATC-RU06 | Generar de reportes y dashboards | 03/05/25 | Wong Portillo, Lenis Rossi | ✘ |

1. Gestión de Release y Entrega del Software
2. Gestión de Release (Cómo voy a hacer la entrega de los release de los clientes)

La Gestión de Release establece un proceso que garantiza que la entrega de cada nueva versión del software sea estable y conforme a los requisitos pactados con el cliente, garantizando el cumplimiento de los estándares de calidad y trazabilidad.

| Fases del proceso | Descripción |
| --- | --- |
| Planificación del Release | Se definen fechas de entrega, alcance funcional, recursos asignados y criterios de aceptación. |
| Gestión de versiones | Cada entrega corresponde a una versión controlada del software y un registro de los cambios implementados. |
| Validación Previa | Todo release debe pasar las pruebas correspondientes antes de ser aprobado por el equipo QA. |
| Revisión en Entorno de Staging | El cliente valida el release en un entorno staging que replica el ambiente productivo, permitiendo verificar funcionalidad y comportamiento, antes del pase a producción. |
| Aprobación del Cliente | Una vez validado el release en staging, se obtiene la aprobación del cliente, autorizando el despliegue en producción. |
| Documentación del Release | Cada release incluye un paquete con el ejecutable, documentación técnica, manual de usuario y registro de cambios. |

1. Entrega del Software (Pase a producción)

El pase a producción es un proceso que permite implementar una versión validada del software al entorno operativo del cliente, garantizando la estabilidad del sistema y su seguridad durante la transición.

| Actividades del Proceso | Descripción |
| --- | --- |
| Verificación de requisitos | Se confirma que todos los ítems asociados al release se encuentran completos, validados y correctamente versionados |
| Revisión técnica integral | Se hace una revisión técnica final para asegurar que el paquete cumple con los estándares de implementación y coincide con la línea base aprobada. |
| Backup del entorno actual | Se realiza un backup completo de la infraestructura productiva (código, datos, configuraciones) como medida de contingencia ante posibles fallos. |
| Despliegue del Release | Utilizando herramientas de automatización, se ejecuta el pipeline de CI/CD que despliega la versión en el entorno productivo. |
| Pruebas post-despliegue | Se aplican pruebas rápidas en producción para confirmar el correcto funcionamiento del sistema. En esta etapa, cualquier incidente es reportado y gestionado de inmediato. |
| Monitoreo activo | Durante las primeras horas tras el despliegue, se establece un monitoreo continuo para detectar anomalías, caídas o comportamientos inesperados. |
| Cierre formal | Se elabora un informe del pase a producción. Además, el cliente recibe una notificación de cierre con observaciones finales. |

| Herramientas utilizadas | |
| --- | --- |
| Control de versiones | Git, GitHub |
| Automatización de despliegue | Jenkins, GitHub Actions |
| Pruebas y aseguramiento | Selenium, JUnit |
| Gestión de tareas | Jira, Trello |
| Monitoreo | Grafana, Prometheus |