**WILSON DEVS**

****

PLAN DE GESTIÓN DE LA CONFIGURACIÓN

**Docente: Wong Portillo, Lenis Rossi**

**Curso: Gestión de la configuración del software**

**Sección: 1**

**Grupo: 4**

**Integrantes: Blas Juarez, Marx Johan Cristopher**

**Blas Muñoa, Joao Lestter**

**Hurtado Santos, Estiven Salvador**

**Llactahuaman Muguerza, Anthony Joel**

**López Terrones, Xiomy Ximena**

**Martinez Santos, Luz Cristina**

**Mondragón Zúñiga, Rubén Alberto**

**Morales Robladillo, Nicole Maria**

**Semestre 2022 – 2**

# ÍNDICE

[**ÍNDICE**](#_heading=h.30j0zll) **1**

[**1. INTRODUCCIÓN**](#_heading=h.2s8eyo1) **4**

[**1.1. Situación de la empresa y problemática**](#_heading=h.17dp8vu) **4**

[**1.2. El objetivo del plan**](#_heading=h.3rdcrjn) **4**

[**2. GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE**](#_heading=h.26in1rg) **5**

[**2.1. Roles y responsabilidades**](#_heading=h.lnxbz9) **5**

[2.2. Herramientas, entorno e Infraestructura](#_heading=h.hqptl07t06jo) 5

[**3. ACTIVIDAD**](#_heading=h.1ksv4uv) **8**

[**3.1. Identificación**](#_heading=h.44sinio) **8**

[**3.1.1. Definición de la estructura de las librerías**](#_heading=h.2jxsxqh) **8**

[**3.1.2. Definición de Líneas Base**](#_heading=h.z337ya) **9**

[**3.1.3. Lista de clasificación de CI**](#_heading=h.3j2qqm3) **10**

[3.1.4. Definición de la Nomenclatura de ítem](#_heading=h.6zpcyj507s5g) 12

[3.1.4.1. Subcomponentes de la Nomenclatura de ítem](#_heading=h.4a3b2cedi3aq) 12

[3.1.4.2. Casos de Nomenclatura de ítem](#_heading=h.8lhv4xxyfoqq) 13

[**3.1.5. Lista de ítem con la nomenclatura**](#_heading=h.4i7ojhp) **14**

[3.2. Control de la GCS](#_heading=h.vh6qzdd0osir) 15

[**3.2.1. Formatos de solicitudes de Cambio (1c/u)**](#_heading=h.1ci93xb) **15**

[**3.2.4. Proceso de Gestión de Cambios**](#_heading=h.3whwml4) **16**

[**3.3. Estado**](#_heading=h.2bn6wsx) **16**

[**3.3.1. Definición de Reportes para el Estado (Gestor - 6)**](#_heading=h.qsh70q) **16**

[**3.3.2. Definición de Reportes para el Estado (Jefe de PY - 4)**](#_heading=h.3as4poj) **16**

[**3.3.3. Definición de Reportes para el Estado (Desarrollador- 3)**](#_heading=h.1pxezwc) **16**

[**3.4. Auditoría**](#_heading=h.49x2ik5) **16**

[**3.4.1. Definición de Reportes de Auditorías (8)**](#_heading=h.2p2csry) **16**

[**3.5. Entrega y gestión de release**](#_heading=h.147n2zr) **16**

[**3.5.1. Proceso de pase a producción**](#_heading=h.3o7alnk) **16**

[**3.5.2. Gestión de release**](#_heading=h.23ckvvd) **16**

[**4. REFERENCIAS**](#_heading=h.z8avthfk4by7) **16**

**CONTROL DE VERSIÓN**

**Historial de versiones**

| **Fecha** | **Versión** | **Autor** | **Descripción** |
| --- | --- | --- | --- |
| 03/10/2022 | 1.0 | Wilson Devs | Creación del documento “Plan - PGC” |
| 05/11/2022 | 2.0 | Wilson Devs | Finalización de líneas base y casos de nomenclatura |

# 1. INTRODUCCIÓN

## 1.1. Situación de la empresa y problemática

En la actualidad, las empresas de software están en auge, lo cual genera que gestionen constantemente una gran cantidad de proyectos. Sin embargo, la mayoría de ellas no cuenta con la planificación y la gestión necesaria para tener un control sobre las muchas versiones de los muchos proyectos que trabajan, pues, además de que el proyecto evoluciona con el paso del tiempo, cada desarrollador maneja una versión independiente. Es aquí donde resulta necesario tener un área especializada de control de versiones en las empresas de software. Es por ello que, “Wilson Devs”, una empresa de desarrollo de software en Lima (Perú) para empresas pequeñas y medianas, ha decidido que, a pesar de tener un año en la industria del software, debe implementar esta área con el fin de prevenir problemas de control de versiones en los proyectos futuros. Actualmente, la empresa se encuentra desarrollando una plataforma web para el alquiler de perfiles de cuentas de servicios de streaming, la cual se llamará “Zorrito PLUS” y está asociado al cliente FOX streaming, además de contar con otro cliente estratégico que es Gimnasio Smartfit con el cuál también se está realizando un proyecto de software.

## 1.2. El objetivo del plan

El objetivo es establecer las pautas iniciales para la administración y rastreo de los productos de software emergentes junto con sus versiones. Además del control de la configuración identificada del software, las herramientas y el hardware a lo largo del ciclo de vida. En base a ello, se busca:

* Administrar remotamente el sistema.
* Reducir el tiempo de inactividad del usuario.
* Hacer copias de seguridad confiables del software.
* Configurar la estación de trabajo.
* Dar soporte a múltiples usuarios.

# 2. GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE

## 2.1. Roles y responsabilidades

Los roles involucrados en la gestión de los cambios dependen de diversos factores, como el tamaño y el tipo de organización de TI. En la Tabla 1, se muestran los roles presentes para cada proyecto, la cantidad de personas que toman dicho rol, y cuáles son las acciones que realizan.

**Tabla 1**

*Roles y responsabilidades de la gestión de la configuración*

| **Cantidad Necesaria de personal** | **Rol en la Gestión de la Configuración** | **Especificación de Rol** |
| --- | --- | --- |
| 3 | Gestor de la Gestión de la Configuración | Encargado de implementar, mantener y mejorar la gestión de la configuración. |
| 2 | Bibliotecario | Define y da mantenimiento a las bibliotecas que son usadas durante la gestión de configuración. |
| 3 | Comité de Control de Cambios | Evaluar el registro de cambios, las solicitudes de cambio y seguimiento de registro de los eventos, las solicitudes de cambios a través de sus respectivos ciclos de vida. |

*Nota*. Elaboración propia.

## 2.2. Herramientas, entorno e Infraestructura

Los sistemas de control de versiones son herramientas de software que ayudan a los equipos de software a gestionar los cambios en el código fuente a lo largo del tiempo. A medida que los entornos de desarrollo se aceleran, los sistemas de control de versiones ayudan a los equipos de software a trabajar de forma más rápida e inteligente. El software de control de versiones realiza un seguimiento de todas las modificaciones en el código en un tipo especial de base de datos. Si se comete un error, los desarrolladores pueden ir hacia atrás en el tiempo y comparar las versiones anteriores del código para ayudar a resolver el error, al tiempo que se minimizan las interrupciones para todos los miembros del equipo. Uno de los sistemas de control de versiones más utilizados es Git. Un asombroso número de proyectos de software dependen de este para el control de versiones, incluidos proyectos comerciales y de código abierto. Los desarrolladores que han trabajado con Git cuentan con una buena representación en la base de talentos disponibles para el desarrollo de software, y este sistema funciona a la perfección en una amplia variedad de sistemas operativos e IDE (entornos de desarrollo integrados). Para este versionamiento, se suelen emplear repositorios en la nube que almacenan las versiones del código del proyecto, a los que se puede acceder cuando sea necesario. Ejemplos de estos repositorios son Github, GitLab y Mercurial, los cuales se comparan entre sí en la Tabla 2 (Atlassian, s.f.).

**Tabla 2**

*Tabla comparativa entre Github, GitLab y Mercurial*

| **Criterios** | **GitHub** | **GitLab** | **Mercurial** |
| --- | --- | --- | --- |
| Gratuito | Sí | Sí | Sí |
| Sistema de control de Versiones | Git | Git | Mercurial |
| Curva de aprendizaje | Sencillo de aprender | Sencillo de aprender | Mucho más sencillo que git |
| Control de trabajo colaborativo | Sí | Sí | No |
| Centralizado | Trabajo local - servidor | Trabajo local - servidor | Trabajo directo con el servidor |
| Distribuido | Microsoft | GitLab Inc | Mercurial |
| Integración Continua | Sí, pero se debe trabajar con herramientas de terceros | Sí | No |
| Flujo de trabajo | Enfocado en velocidad | Enfocado en fiabilidad | Enfocado en la velocidad |
| Rastreador de problemas | Sí | Sí | No |
| Revisión de código | Sí | Sí | No |
| Pruebas de rendimiento de carga | Sí, en la aplicación | Sí, en plan de pago | Sí, pero no es confiable |

*Nota*. Elaboración propia. Información basada en Diferenciario (2019) y IONOS (s.f.).

En los proyectos de construcción de software, “Wilson Devs” emplea el sistema de control de versiones Git y el repositorio GitHub para el control de las versiones del producto. Se ha elegido esta herramienta por todos los beneficios que brinda y porque gran parte del equipo cuenta con experiencia utilizándola. Además, debido a que todos los integrantes del equipo son estudiantes, Github resulta conveniente ya que brinda una serie de funciones y servicios para que los estudiantes puedan desarrollar proyectos de software y controlar su versionamiento de forma gratuita. Se adjunta el [link](https://platzi.com/tutoriales/1557-git-github/5899-como-obtener-github-student-developer-pack/) de la página web que describe los beneficios que brinda y el procedimiento para poder activar el beneficio.

# 3. ACTIVIDAD

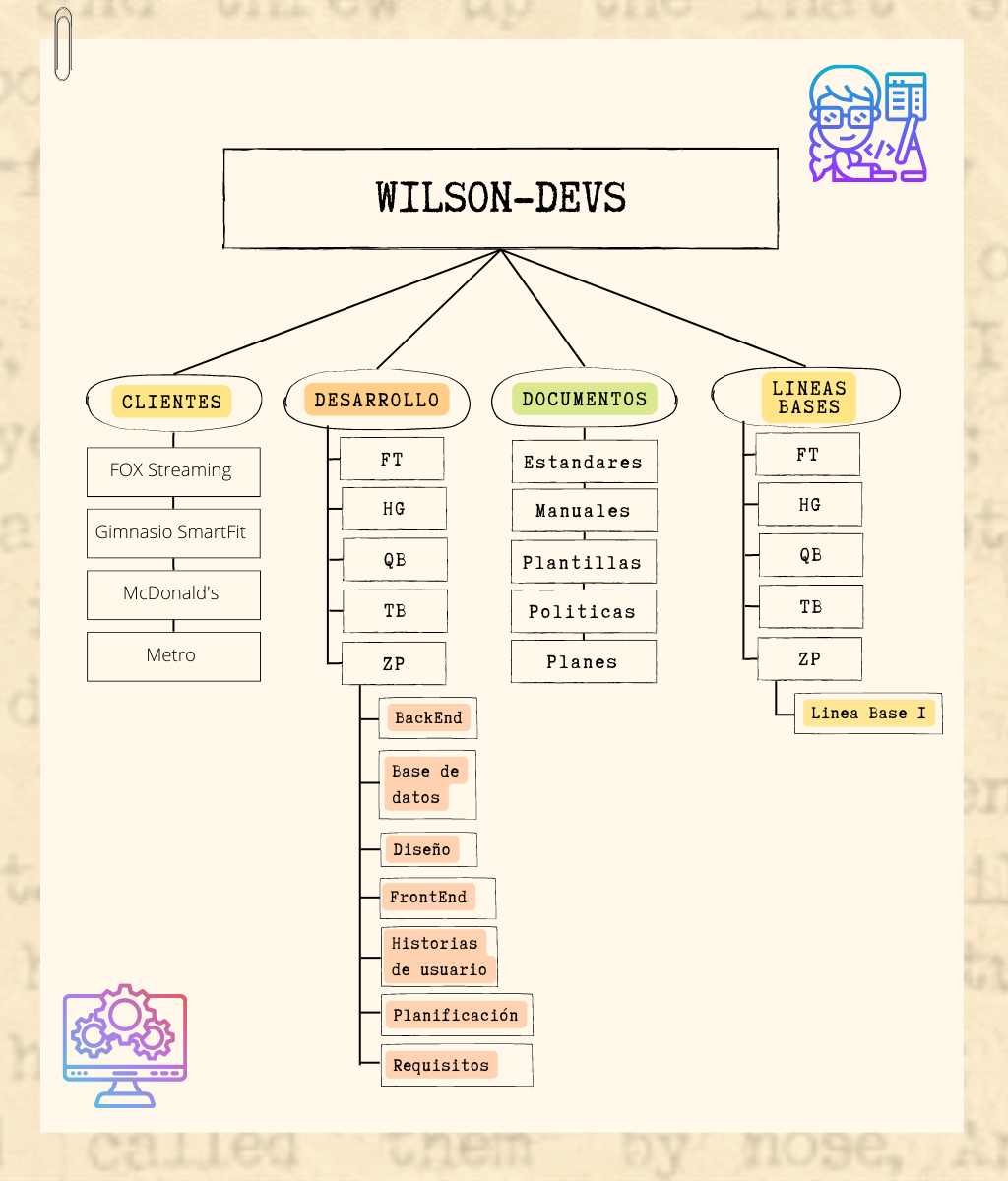
## 3.1. Identificación

### 3.1.1. Definición de la estructura de las librerías

Wilson Devs cuenta con un repositorio en Github, en donde se almacenan los documentos de la empresa sobre la gestión de la configuración del software. En la Figura 1 se muestra la estructura del mismo.

**Figura 1**

*Estructura del repositorio de Wilson Devs*



*Nota*. Elaboración propia.

### 3.1.2. Definición de Líneas Base

La línea base de un proyecto es la última versión del cronograma que ha sido aprobada formalmente por el sponsor el proyecto, o el comité de dirección del proyecto, y que define los objetivos de este en relación a los plazos. Esto no implica que cada vez que modifiquemos el cronograma se genere una nueva línea base, de hecho la mayoría de las veces no ocurrirá así (idealmente nunca debería). En la Tabla 3 se muestra la estructura general de las líneas base de los proyectos (Recursos en Project Management, 2014).

**Tabla 3**

*Estructura básica de las líneas base de los proyectos*

| **LÍNEA BASE 1** | *Plan del Proyecto (PROJECT CHARTER)* |
| --- | --- |
| *Cronograma del Proyecto* |
| *Repositorio GitHub* |
| *Lista de Historias de Usuario* |
| *Documentos de Especificación de Requisitos* |
| *Documento de Especificación de UI* |
| *Documento de Guía de Estilos* |
| *Documento de Especificación de la BD* |
| *Documento de Arquitectura del Software* |
| *Reporte del Primer Sprint* |
| **LÍNEA BASE 2** | *Módulo de Página de Inicio* |
| *Módulo de Registro* |
| *Módulo de Cliente* |
| *Módulo de Administrador* |
| *Reporte del Segundo Sprint* |
| **LÍNEA BASE 3** | *Módulo de Administrador* |
| *Módulo de Cliente* |
| *Módulo de Administrador* |
| *Manual de usuario* |
| *Documento de Pruebas del Software* |
| *Acta de cierre del proyecto* |

*Nota*. Elaboración propia.

### 3.1.3. Lista de clasificación de CI

Todos los proyectos de construcción de software producen ítems durante el proceso, los cuales vendrían a ser el resultado de cada actividad realizada dentro del cronograma del mismo. En la Tabla 3, se listan los ítems producidos en todos los proyectos realizados por Wilson Devs, indicando su tipo, descripción y proyecto al que pertenecen.

**Tabla 4**

*Lista de clasificación de ítems*

| **Tipo de ítem**  **(E= Evolución F=Fuente S=Soporte)** | **Descripción de ítem** | **Proyecto** |
| --- | --- | --- |
| E | Project Charter | ZP |
| E | Cronograma del Proyecto | ZP |
| E | Lista de Historias de Usuario | ZP |
| E | Documento de Especificación de Requisitos | ZP |
| E | Documento de Especificación UI | ZP |
| E | Documento de Guía de Estilos | ZP |
| E | Documento de Especificación de Base de Datos | ZP |
| E | Documento de Arquitectura de Software | ZP |
| E | Manual de Usuario | ZP |
| E | Documento de Pruebas del software | ZP |
| E | Acta de cierre de proyecto | ZP |
| F | Pantalla de página de inicio | ZP |
| F | Pantalla de Registro de Usuario | ZP |
| F | Pantalla de Inicio de Sesión | ZP |
| F | Pantalla de control de Cuentas para Administradores | ZP |
| F | Pantalla de Control de Perfiles para Administradores | ZP |
| F | Pantalla de solicitud de perfiles | ZP |
| E | Project Charter | FT |
| E | Cronograma del Proyecto | FT |
| E | Lista de Historias de Usuario | FT |
| E | Documento de Especificación de Requisitos | FT |
| E | Documento de Especificación UI | FT |
| S | JDK 8 | FT |
| E | Project Charter | HG |
| E | Cronograma del Proyecto | HG |
| E | Lista de Historias de Usuario | HG |
| E | Documento de Especificación de Requisitos | HG |
| E | Documento de Especificación UI | HG |
| F | Pantalla de Administradores | HG |
| F | Pantalla de solicitud | HG |
| E | Project Charter | TB |
| E | Cronograma del Proyecto | TB |
| E | Lista de Historias de Usuario | TB |
| E | Documento de Especificación de Requisitos | TB |
| E | Documento de Especificación UI | TB |
| F | Pantalla de Administradores | TB |
| F | Pantalla de solicitud | TB |
| E | Project Charter | QB |
| E | Cronograma del Proyecto | QB |
| E | Lista de Historias de Usuario | QB |
| E | Documento de Especificación de Requisitos | QB |
| E | Documento de Especificación UI | QB |
| F | Pantalla de Administradores | QB |
| F | Pantalla de solicitud | QB |

*Nota*. Elaboración propia.

### 3.1.4. Definición de la Nomenclatura de ítem

#### 3.1.4.1. Subcomponentes de la Nomenclatura de ítem

* **Para identificar los Proyectos:**

Fórmula: [Acrónimo de Nombre del Proyecto]

Ejemplo: Zorrito PLUS → ZP

* **Para la identificación de Archivos:**

Fórmula: [Acrónimo de Nombre del Archivo]

Ejemplo: Project Charter → PC

* **Extensión:**

Fórmula: [Extensión abreviada]

Ejemplo:

Archivo de Microsoft Excel → xlsx

Archivo de Microsoft Word → docx

Archivo comprimido RAR → rar

#### 3.1.4.2. Casos de Nomenclatura de ítem

* **Caso 1: Para la identificación de archivos de Wilson Devs**

Fórmula: [Acrónimo de Nombre del Archivo] + “.” + [Extensión]

Ejemplo:

Plan de Gestión de la Configuración → PGC.docx

* **Caso 2: Para la identificación general de archivos de un proyecto específico**

Fórmula: [Acrónimo de Nombre del Proyecto] + “-” + [Acrónimo de Nombre del Archivo] + “.” + [Extensión]

Ejemplo:

Project Charter del proyecto Zorrito PLUS (Archivo de Microsoft Word) → ZP-PC.docx

* **Caso 3: Para la identificación de Documentos que se hacen en base a una Historia de Usuario de un proyecto específico**

Fórmula: [Acrónimo de Nombre del Proyecto] + “-” + [Acrónimo de Nombre del Archivo] + “-” + “0” + [Número de Historia de Usuario] + ”.” + [Extensión]

Ejemplo:

Documento de Especificación de Requisitos de la Historia de Usuario 1 del proyecto Zorrito PLUS → ZP-DER-01.docx

Código fuente de la Historia de Usuario 1 del proyecto Zorrito PLUS → ZP-CÓDIGO-01.rar

* **Caso 4: Ítems con la misma nomenclatura**

Fórmula: [Nomenclatura expuesta en los anteriores casos] + [Letra que sigue a la primer letra del Acrónimo del Nombre del Archivo]

Ejemplo:

Se tienen dos archivos: Plan del Cronograma y Project Charter del proyecto Zorrito PLUS

Nomenclatura del **Pl**an del Cronograma del proyecto Zorrito PLUS → ZP-PLC.xlsx

Nomenclatura del **Pr**oject Charter del proyecto Zorrito PLUS → ZP-PRC.docx

### 3.1.5. Lista de ítem con la nomenclatura

Los ítems producidos en los proyectos de Wilson Devs, son identificados bajo una nomenclatura, cuyo formato y casos se exponen en la sección 3.1.4. En la Tabla 4, se listan los ítems producidos en todos los proyectos realizados por Wilson Devs, indicando su nomenclatura, extensión y a que proyecto al cual pertenecen. Esto con el objetivo de tener constancia de la cantidad de ítems que maneja la empresa.

**Tabla 5**

*Lista de ítems con nomenclatura*

| **Descripción del ítem** | **Nomenclatura** | **Extensión** | **Proyecto** |
| --- | --- | --- | --- |
| Project Charter | ZP - PC | .DOCX | ZP |
| Cronograma del Proyecto | ZP - CP | .XLSX | ZP |
| Lista de Historias de Usuario | ZP - LHU | .XLSX | ZP |
| Documento de Especificación de Requisitos | ZP - DER | .DOCX | ZP |
| Documento de Especificación UI | ZP - DEUI | .PDF | ZP |
| Documento de Guía de Estilos | ZP - DGE | .PDF | ZP |
| Documento de Especificación de Base de Datos | ZP - DEBD | .DOCX | ZP |
| Documento de Arquitectura de Software | ZP - DAS | .DOCX | ZP |
| Manual de Usuario | ZP - MU | .DOCX | ZP |
| Documento de Pruebas del software | ZP - DPS | .DOCX | ZP |
| Acta de cierre de proyecto | ZP - ACP | .DOCX | ZP |
| Project Charter | FT - PC | .DOCX | FT |
| Cronograma del Proyecto | FT - CP | .XLSX | FT |
| Lista de Historias de Usuario | FT - LHU | .XLSX | FT |
| Documento de Especificación de Requisitos | FT - DER | .DOCX | FT |
| Documento de Especificación UI | FT - DEUI | .PDF | FT |

*Nota*. Elaboración propia.

## 3.2. Control de la GCS

### 3.2.1. Formatos de solicitudes de Cambio (1c/u)

Se deben brindar los siguientes datos generales:

* Nombre del Proyecto
* Cliente

El formato de tabla de solicitud de cambio se muestra en la Tabla 6.

**Tabla 6**

*Formato de solicitud [número] - [Título]*

| **Campo** | **Descripción** |
| --- | --- |
| **Fecha de creación** | <Fecha de petición del cambio> |
| **Sistema** | <Nombre del sistema> |
| **Fuente (Persona que ha identificado la necesidad del cambio - Stakeholder)** | <Persona que ha identificado la necesidad de cambio> |
| **Autor (Autoriza el cambio)** | <Persona que tiene a cargo el sistema, y debe formalizar la petición de cambio> |
| **Descripción** | <Descripción del cambio>  Esta descripción debe contener las siguientes pautas: El motivo, el propósito, los ítems de la configuración, los recursos necesarios para la implementación y otros detalles. |
| **Justificación** | El cambio se considera necesario debido a que <Justificación del cambio solicitado> |

*Nota*. Elaboración propia.

### 3.2.4. Proceso de Gestión de Cambios

## 3.3. Estado

### 3.3.1. Definición de Reportes para el Estado (Gestor - 6)

### 3.3.2. Definición de Reportes para el Estado (Jefe de PY - 4)

### 3.3.3. Definición de Reportes para el Estado (Desarrollador- 3)

## 3.4. Auditoría

### 3.4.1. Definición de Reportes de Auditorías (8)

## 3.5. Entrega y gestión de release

### 3.5.1. Proceso de pase a producción

### 3.5.2. Gestión de release

# 4. REFERENCIAS

Atlassian. (s.f.). *Qué es el control de versiones*. Atlassian. https://www.atlassian.com/es/git/tutorials/what-is-version-control

Atlassian. (s.f.). *Qué es Git: conviértete en todo un experto en Git con esta guía*. Atlassian. https://www.atlassian.com/es/git/tutorials/what-is-git

Diferenciario. (2019, mayo 8). Diferencia entre Git y Mercurial. Diferenciario. https://diferenciario.com/git-y-mercurial/

IONOS. (s.f.). GitLab vs. GitHub: comparativa de ambos sistemas de control de versiones. IONOS Digital Guide. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/gitlab-vs-github/

Recursos en Project Management. (2014, junio 14). *Línea base del proyecto. Definición y ejemplos para usarla eficazmente 📈*. Recursos en Project Management. https://www.recursosenprojectmanagement.com/linea-base-del-cronograma/