***“Año del Fortalecimiento de la Soberanía Nacional”***

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**



**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**Consultora G2**

**GRUPO 2**

***PLAN DE GESTIÓN DE LA CONFIGURACIÓN***

***(PGC)***

**Curso:** Gestión de configuración y mantenimiento

**Profesora:** Lenis Rossi Wong Portillo

**Integrantes:**

* Rodríguez Callata Elton John
* Pino Benavente Cesar Augusto
* Vásquez Enrique Jhonatan David
* Rupay Huamaní Jorge Luis
* Calderón Bermudo Paulo Cesar
* Pizarro Sánchez Frank Sebastián
* Laos Carrasco Rafael Alonso

**LIMA – PERÚ**

**2022**

**CONTROL DE VERSIONES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Autor(es)** | **Descripción** | **Fecha** |
| 1.0 | * Rodríguez Callata Elton John * Pino Benavente Cesar Augusto * Vásquez Enrique Jhonatan David * Rupay Huamaní Jorge Luis * Calderón Bermudo Paulo Cesar * Pizarro Sánchez Frank Sebastián * Laos Carrasco Rafael Alonso | * Introducción (Situación de la empresa – Propósito del plan) * Gestión de la SCM (Roles o responsabilidades – Herramientas – Diagrama de la arquitectura) | 07/06/2022 |

**Plan de Gestión de la Configuración**

**Consultora G2**

**Versión 1.0**

1. **Introducción**
   1. **Situación de la empresa**

La empresa g2 es una consultora que realiza actividades de desarrollo y mantenimiento de proyectos de software para diversos clientes, teniendo mayor experiencia y especializándose en el desarrollo de sistemas de gestión de inventario para empresas comercializadoras, sin embargo, se han presentado problemas en el desarrollo de ciertos proyectos. Una de las problemáticas más comunes que tenemos en nuestra consultora g2 se generan cuando desarrollamos un proyecto de software, pues lo miembros del equipo de desarrollo hacen uso de diferentes repositorios, esto genera que el jefe de proyecto no haga un control eficiente de las versiones de los elementos, además de generar retrasos en las entregas y errores que no se tenían planificados.

Estos problemas se generan por una falta de comunicación entre los miembros del equipo y el jefe de proyecto, así como también la falta de capacitación en el uso de repositorios como GitHub, los miembros del equipo no saben lo que deben hacer cada vez que quieren cambiar algo, es por eso que cada uno trabaja por su propia cuenta.

* 1. **Propósito del plan**

El objetivo primordial de este plan es mantener una correcta integridad de los productos que se logran a través del ciclo de vida del desarrollo del software, en la cual, es tener un correcto seguimiento de que no se realizan cambios incontrolados, como resultado es que todos las personas tengan una correcta versión del sistema, como también las fuentes de código, modelo de datos, requisitos, entre otros.

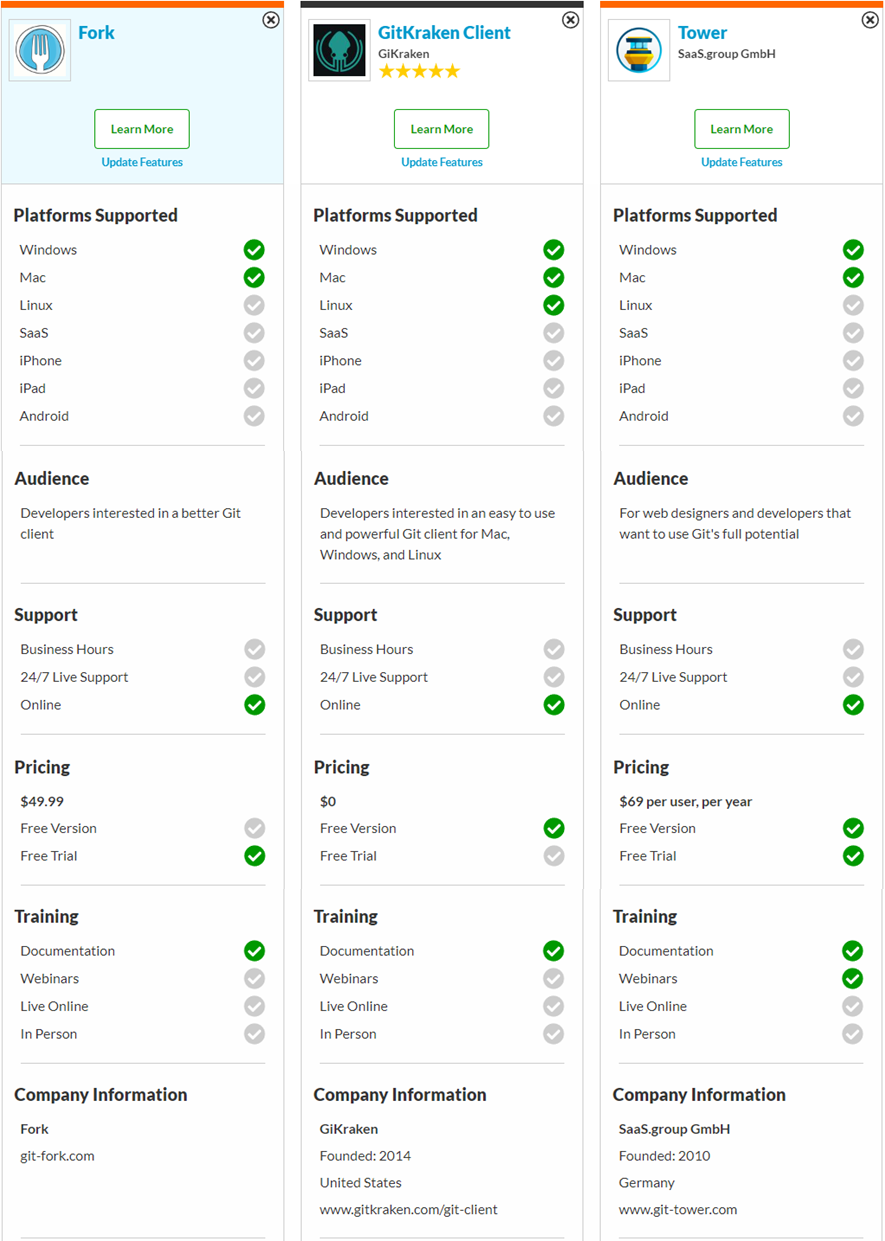
1. **Gestión de la SCM**
   1. **Roles o responsabilidades**

**Tabla 1:** Roles, responsabilidades y cantidad

|  |  |  |  |
| --- | --- | --- | --- |
| Roles | Responsabilidades | Nivel de autoridad | Cantidad |
| Comité de Control de cambios | Establecer un control de cambio de los proyectos de la empresa, conjuntamente con el documento de gestión de la configuración. |  | 1 |
| Gestor de la configuración | Gestionar la infraestructura global de la gestión de la configuración y el entorno del equipo de desarrollo de los productos. | Autoridad para operar las funciones de Gestión de la configuración. | 2 |
| Project Manager | Supervisar el funcionamiento del plan de la gestión de la configuración. | Toda autoridad sobre el proyecto y sus funciones. | 1 |
| Bibliotecario | Mantener los archivos documentales y de desarrollo controlados, trabajar en conjunto con el gestor de la configuración. |  | 1 |
| Inspector del aseguramiento de la calidad | Auditar la gestión de la configuración, velar por la calidad de la gestión de configuración. |  | 2 |
| Miembros del equipo | Formar parte del equipo operativo de los proyectos. | Específica para cada artefacto y cada CI, dependiendo de la tarea asignada a cada miembro. | 7 |

**Nota.** Descripción de las responsabilidades y niveles de autoridad por rol. Elaboración propia.

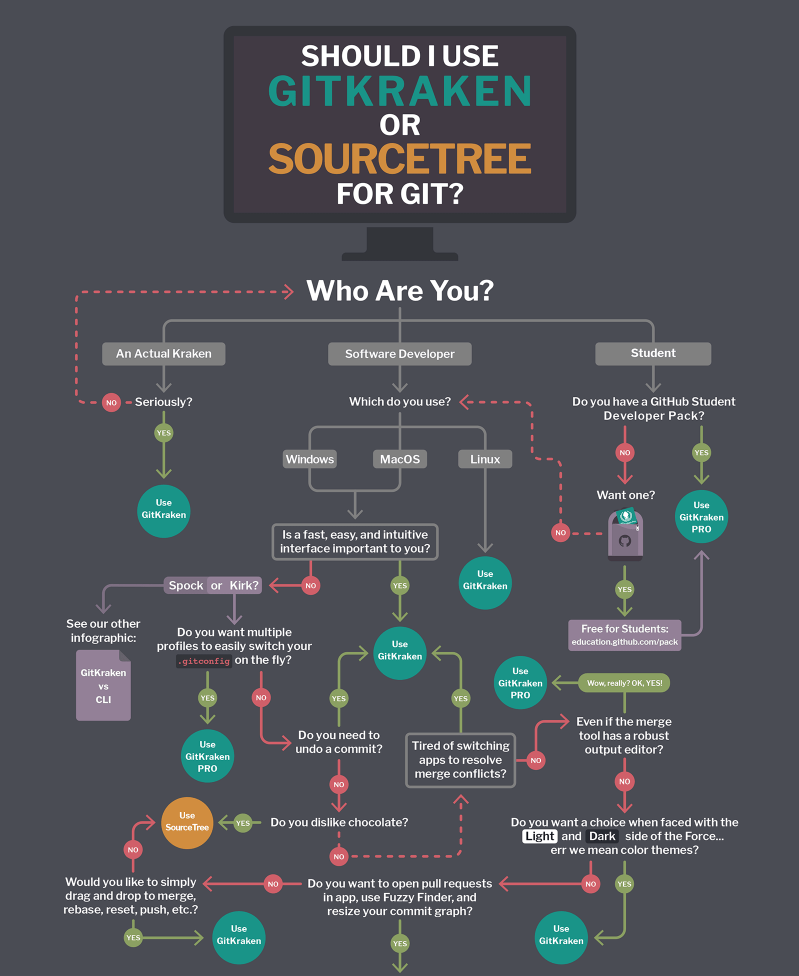
* 1. **Herramientas (Benchmarking)**

**Tabla 2:** Benchmarking Fork, GitKraken Client y Tower

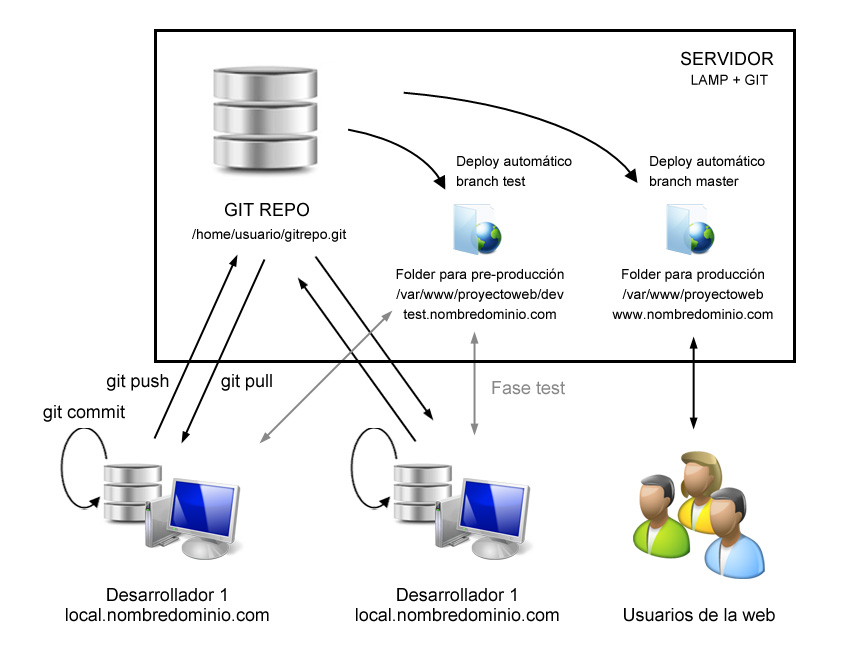
**Nota.** Comparación elaborada por SourceForge. https://sourceforge.net/software/compare/Fork-vs-GitKraken-vs-Tower/

El benchmarking realizado entre las herramientas de visualización de control de versiones Git, toma en consideración criterios de plataformas soportadas, público dirigido, soporte ofrecido, precios, capacitación e información de la compañía. Se eligió la herramienta GitKraken Client debido a la versatilidad de sistemas operativos soportados, la existencia de documentación y soporte online y principalmente el hecho de ser freeware, no representar ningún costo.

* 1. **Diagrama de arquitectura de la herramienta elegida**

**Figura 1:** Razones para usar GitKraken

**Nota.** Infografía elaborada por GitKraken. https://www.gitkraken.com/compare/gitkraken-vs-sourcetree

**Figura 2:** Arquitectura de repositorio Git

**Nota.** Figura creada por Daniel Navarro. http://www.danielnavarroymas.com/gestion-de-proyectos-web-con-git/

La figura 2 describe una arquitectura base de cualquier proyecto web trabajado en un repositorio Git, el cual permite el trabajo de más de un desarrollador en paralelo, gestionando control de versiones por medio de el uso de commits e integrando el desarrollo personal con todo el proyecto en el repositorio por medio de push y pulls. En el repositorio es posible desplegar las versiones funcionales del proyecto mediante deploys de las ramas correspondientes tanto en entornos de pruebas pre-producción, como en el despliegue final para producción, de acceso a los usuarios.

1. **Referencias**

SourceForge. (2022). Fork vs. GitKraken Client vs. Tower Comparison Chart. SourceForge. https://sourceforge.net/software/compare/Fork-vs-GitKraken-vs-Tower/

Navarro, D. (s.f.). Gestión de proyectos Web con GIT. Daniel Navarro y más. http://www.danielnavarroymas.com/gestion-de-proyectos-web-con-git/#comments

GitKraken. (2022). GitKraken vs Sourcetree. GitKraken. https://www.gitkraken.com/compare/gitkraken-vs-sourcetree