**Universidad de las Américas**

**Facultad de Ingeniería y Ciencias Aplicadas**

**Ingeniería en Sistemas de Computación e Informática**

**Implementación de una guía de buenas prácticas de Integración Continua en proyectos de desarrollo en el área de tecnología de una Empresa Privada.**

Edison P. Mosquera León, Carlos X. Ruiz Águila, David F. Martínez Bravo

**GIT & GITFLOW**

1. **Historia.**

La comunidad de desarrolladores del sistema operativo Linux utilizaba BitKeeper, el cual era un sistema de versiones distribuido y comercial. BitKeeper inspiro muchos de los conceptos en los que hoy en día se basó git. Linus Torvalds tras probar diferentes soluciones existentes descubrió que ninguna cumplía con sus expectativas por lo que procedió a inclinarse por la búsqueda de un nuevo sistema de control de versiones por lo que decisión diseñar e implementar su propio sistema de control de versiones. Días después ya se había desarrollado las bases de git para la autogestión del propio código de git dentro de git. Los primeros en adoptarlo fueron Torvalds y el workflow de sus desarrolladores debido a su modo de funcionamiento distribuido, su eficiencia a la hora de trabajar con múltiples ramas y otras operaciones comunes como diffs, git fue diseñado con la intención de gestionar proyectos grandes como el kernel del propio Linux sin problema alguno.

1. **Descripción - Git.**

La herramienta Git es un controlador de versiones diseñado por Linus Torvalds, el cual es un reconocido ingeniero de software creador del kernel del sistema operativo Linux. Esta herramienta permite la facilidad de guardar código fuente, compara distintas versiones del código y navegar de manera fácil entre las distintas versiones de un mismo proyecto.

Dentro de un proyecto en Git tenemos una estructura dividida en tres partes

* **Working Directory:** donde se encuentran todos nuestros ficheros y donde se trabaja constantemente.
* **Staging Area:** en esta área se encuentran los archivos que se modifican y aceptan para una futura revisión.
* **Git Directory:** dentro de esta área de almacenan las revisiones completas del código fuente de un proyecto.

1. **Descripción - GitFlow.**

Gitflow es una idea abstracta del flujo de trabajo que se maneja dentro de lo que es git, lo que nos permite ordenar, configurar y fusionar los distintos tipos de ramas que tenemos dentro de nuestro proyecto. Con gitflow los desarrolladores en vez de trabajar con una única rama maestra, el flujo de trabajo utiliza dos ramas para registrar el historial del proyecto las cuales vendrían a ser, la rama maestra que permite el almacenamiento de todo el historial de publicación oficial y la rama de desarrollo que sirve como una rama de integración de funciones.

Esto permite al equipo después de haber acumulado funciones del proyecto tras haber comprobado su correcto funcionamiento, fusionar la rama maestra con la de desarrollo atribuyendo la preparación para la publicación haciendo posible que el equipo de desarrollo perfeccione la publicación actual mientras otro equipo sigue trabajando en las funciones para la siguiente publicación.

1. **Características.**

* Cada versión registrada en el tiempo es una revisión.
* Cada operación se realiza en el repositorio local.
* Permite enviar cambio a repositorios remotos.
* Su funcionalidad de ramas permite hacer experimentos, nuevos desarrollos o pruebas de diferentes funcionalidades de un proyecto principal.

1. **Beneficios.**

Los puntos fuertes de git que nos pueden ofrecer para su utilización en todo proyecto de software, son:

* Es software libre y de código abierto.
* Permite el trabajo en equipo, permitiendo que varios desarrolladores trabajen simultáneamente sobre un proyecto compartido, sin estar necesariamente cerca el uno del otro.
* Permite mejor y mayor autonomía dentro del desarrollo del proyecto dado a que cada uno de los desarrolladores que trabajan sobre el proyecto poseen una copia local de todo el proyecto y cambios generados permitiendo trabajar de forma individual y a un ritmo propio.
* Permite comparar, fusionar o restaurar versiones de una aplicación y contar con una copia del código fuentes para volver atrás ante cualquier fallo o imprevisto.

1. **Conclusiones.**

* Git permite que dentro de un equipo de trabajo exista mayor agilidad y sencillez, consiguiendo una detección temprana de fallos, disminución de errores humanos y seguimiento de etapas del proyecto.
* Permitir el alojamiento de proyectos dentro de repositorios como GitHub o GitLab es una de las decisiones profesionales más inteligentes dado que se dispondrá de un robusto sistema de gestión de proyectos.