**Universidad**

Universidad tecnológica del centro de Veracruz

**Carrera**

Ingeniería desarrollo y gestión de software

**Cuatrimestre**

9ª

**Datos de alumno**

Baez Andrade Alfonso

**Datos del docente**

ITI Francisco Trujillo Romero

**Fecha**

4/junio/2022

Introducción

La importancia de la implementación de sistemas de versiona miento para la administración de un proyecto es vital a mi punto de vista ya que nos ayuda a organizar y llevar un control referente a un proyecto donde podemos dar seguimiento y gestionar mejor nuestro proyecto donde podemos tener la facilidad de ver las versiones y procedimiento de nuestro proyecto de esta manera con el versiona miento podemos localizar fácilmente si se presenta un error o en algunos casos llevar el control de actividades de cada miembro.

De esta manera usamos el versiona miento como herramienta de prevención y gestión a mi parecer ya que, si llegáramos a tener un margen de error durante el proyecto, podremos visualizar y localizar rápidamente el origen de dicho error.

No solo localizar errores si no también definir actividades repartir roles y trabajar un solo proyecto cómodo y de manera remota.

plataformas más utilizadas para el versiona miento

* 1. GITHUB

GitHub es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, y que fue comprada por Microsoft [en junio del 2018](https://www.xataka.com/aplicaciones/oficial-microsoft-compra-github-7-500-millones-dolares). La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargarte la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.

* 1. GITLAB

Es una plataforma Git y DevOps basada en la nube que ayuda a los desarrolladores a supervisar, probar y desplegar su código.

Gitlab ofrece un servicio web de control de versiones y desarrollo de software colaborativo basado en Git. Además de ser un gestor de repositorios, ofrece generación de wikis y un sistema de seguimiento de errores.

GitLab es una suite completa que permite gestionar, administrar, crear y conectar los repositorios con diferentes aplicaciones y hacer todo tipo de integraciones, ofreciendo un ambiente y una plataforma para realizar varias etapas de su SDLC/ADLC y DevOps.

* 1. BITBUCKET

Es una herramienta de alojamiento de código y colaboración basada en Git diseñada para equipos. Las inmejorables integraciones de Bitbucket con Jira y Trello están concebidas para unir a todo el equipo de software con el fin de poner en práctica un proyecto. Ofrecemos un lugar en el que tu equipo pueda colaborar con código desde el concepto hasta la nube, crear código de calidad mediante pruebas automatizadas e implementar código con total seguridad.

flujos de trabajo para los controladores de versiones

1. GIT FLOW

Es un modelo alternativo de creación de ramas en Git en el que se utilizan ramas de función y varias ramas principales. Fue [Vincent Driessen en nvie](http://nvie.com/posts/a-successful-git-branching-model/) quien lo publicó por primera vez y quien lo popularizó. En comparación con el desarrollo basado en troncos, Gitflow tiene diversas ramas de más duración y mayores confirmaciones.

1. GITHUB FLOW

Es un modelo alternativo de creación de ramas en Git en el que se utilizan ramas de función y varias ramas principales. Fue [Vincent Driessen en nvie](http://nvie.com/posts/a-successful-git-branching-model/) quien lo publicó por primera vez y quien lo popularizó.

1. GITLAB FLOW

Es una alternativa más simple a  [GitFlow](https://about.gitlab.com/blog/2020/03/05/what-is-gitlab-flow/)  y combina desarrollo basado en funciones y ramas de funciones con seguimiento de problemas. Con GitLab Flow, todas las funciones y correcciones van a  main rama mientras se habilitan  producción y  stable se ramifican. GitLab Flow incluye un conjunto de [mejores prácticas](https://about.gitlab.com/topics/version-control/what-are-gitlab-flow-best-practices/) y pautas para garantizar que los equipos de desarrollo de software sigan un proceso fluido para enviar funciones de forma colaborativa.

1. ONE FLOW

OneFlow fue concebido como una alternativa más simple a GitFlow. Sin embargo, “más simple” no quiere decir que permita hacer menos.

Con OneFlow es necesario tener un master Branch que tendrá una vida infinita en el repositorio de código, además, cada nueva versión de producción debe basarse en la versión anterior, una condición que la mayoría de proyectos de software cumple. La mayor diferencia entre One Flow y Git Flow es que en OneFlow no existe develop.

Conclusiones

Para concluir cabe destacar que Utilizar un software de control de versiones es una práctica recomendada para los equipos de software y de DevOps de alto rendimiento. El control de versiones también ayuda a los desarrolladores a moverse más rápido y permite que los equipos de software mantengan la eficiencia y la agilidad a medida que el equipo se escala para incluir más desarrolladores.

Bibliografía

<https://bitbucket.org/product/es/guides/getting-started/overview>

<https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>

<https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow#:~:text=Gitflow%20es%20un%20modelo%20alternativo,vez%20y%20quien%20lo%20populariz%C3%B3>.

<https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow>

<https://about.gitlab.com/topics/version-control/what-is-gitlab-flow/>

<https://openwebinars.net/blog/estrategias-de-branching-gitflow-gitlab-flow-oneflow-github-flow/>