

Universidad Tecnológica del Centro de Veracruz.

Programa Educativo: Ingeniería en Desarrollo y Gestión de Software.

9°

Desarrollo Web Integral

ITI. Francisco Trujillo Romero

Lectura Técnica

Flores Lucas Itzel Teresa

20183L301037

Cuitláhuac, Ver., a 04 de Junio de 2022

# **INTRODUCCIÓN**

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.

Principalmente cuando 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.

ÍNDICE

[**INTRODUCCIÓN** 2](#_Toc105276897)

[**CAPITULO I** 4](#_Toc105276898)

[GITHUB 4](#_Toc105276899)

[¿Qué es GitHub? 4](#_Toc105276900)

[¿Para qué sirve? 4](#_Toc105276901)

[Principales características 4](#_Toc105276902)

[GITLAB 5](#_Toc105276903)

[¿Qué es GitLab? 5](#_Toc105276904)

[Funcionamiento 5](#_Toc105276905)

[Principales características 5](#_Toc105276906)

[BITBUCKET 6](#_Toc105276907)

[¿Qué es Bitbucket? 6](#_Toc105276908)

[Principales características 6](#_Toc105276909)

[**CAPITULO II** 7](#_Toc105276910)

[GIT-FLOW 7](#_Toc105276911)

[¿Qué es git-flow? 7](#_Toc105276912)

[Ramas principales 7](#_Toc105276913)

[GITHUB FLOW 8](#_Toc105276914)

[¿Qué es GitHub Flow? 8](#_Toc105276915)

[Tipos de Branches 8](#_Toc105276916)

[GITLAB FLOW 8](#_Toc105276917)

[¿Qué es GitLab Flow? 8](#_Toc105276918)

[ONEFLOW 9](#_Toc105276919)

[¿Qué es OneFlow? 9](#_Toc105276920)

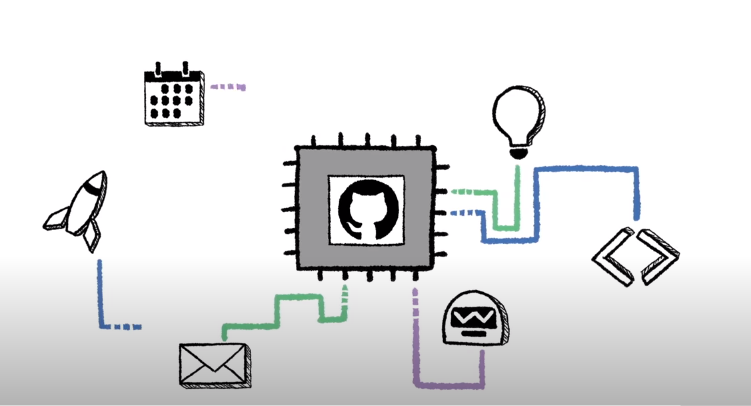
[**CONCLUSIÓN** 10](#_Toc105276921)

[**BIBLIOGRAFIA** 10](#_Toc105276922)

# **CAPITULO I**

## GITHUB

### ¿Qué es GitHub?



GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git(Castillo, 2017).

### ¿Para qué sirve?

GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.

### Principales características

Entre las principales características que nos ofrece GitHub podemos destacar:

* Una Wiki que opera con Git para el mantenimiento de las distintas versiones de las páginas.
* Sistema de seguimiento de problemas. Se trata de un sistema muy parecido al tradicional ticket, donde cualquier miembro del equipo o persona (si nuestro repositorio es público) puede abrir una consulta o sugerencia que se quiera hacer.
* Herramienta de revisión de código. Permite añadir anotaciones en cualquier punto de un fichero.
* Visor de ramas que permite comparar los progresos realizados en las distintas ramas de nuestro repositorio(Gómez Bachiller, n.d.).

## GITLAB

### Pipelines en Gitlab, el camino hacia el CI/CD (parte I) | Develatio's Blog¿Qué es GitLab?

GitLab es un software de control de versiones basado en la web basado en git y agrega funciones adicionales como el seguimiento de errores de administración de sucursales y la integración continua. Está desarrollado en rubí.

### Funcionamiento

En GitLab podemos gestionar principalmente proyectos, Grupos y Sinppets. Los proyectos son los protagonistas del sistema, básicamente repositorios de software gestionados por GitLab y todo el ecosistema GitLab. Los grupos son básicamente empresas y usuarios. Los snippets por su parte son como pedazos de código que puedes dejar para hacer cualquier cosa.

### Principales características

* Velocidad
* Diseño sencillo
* Completamente distribuido
* Se adapta para todo tipo de proyectos.
* Desarrollo no lineal (branching)(MIGUEL ANGEL MARTINEZ, n.d.).

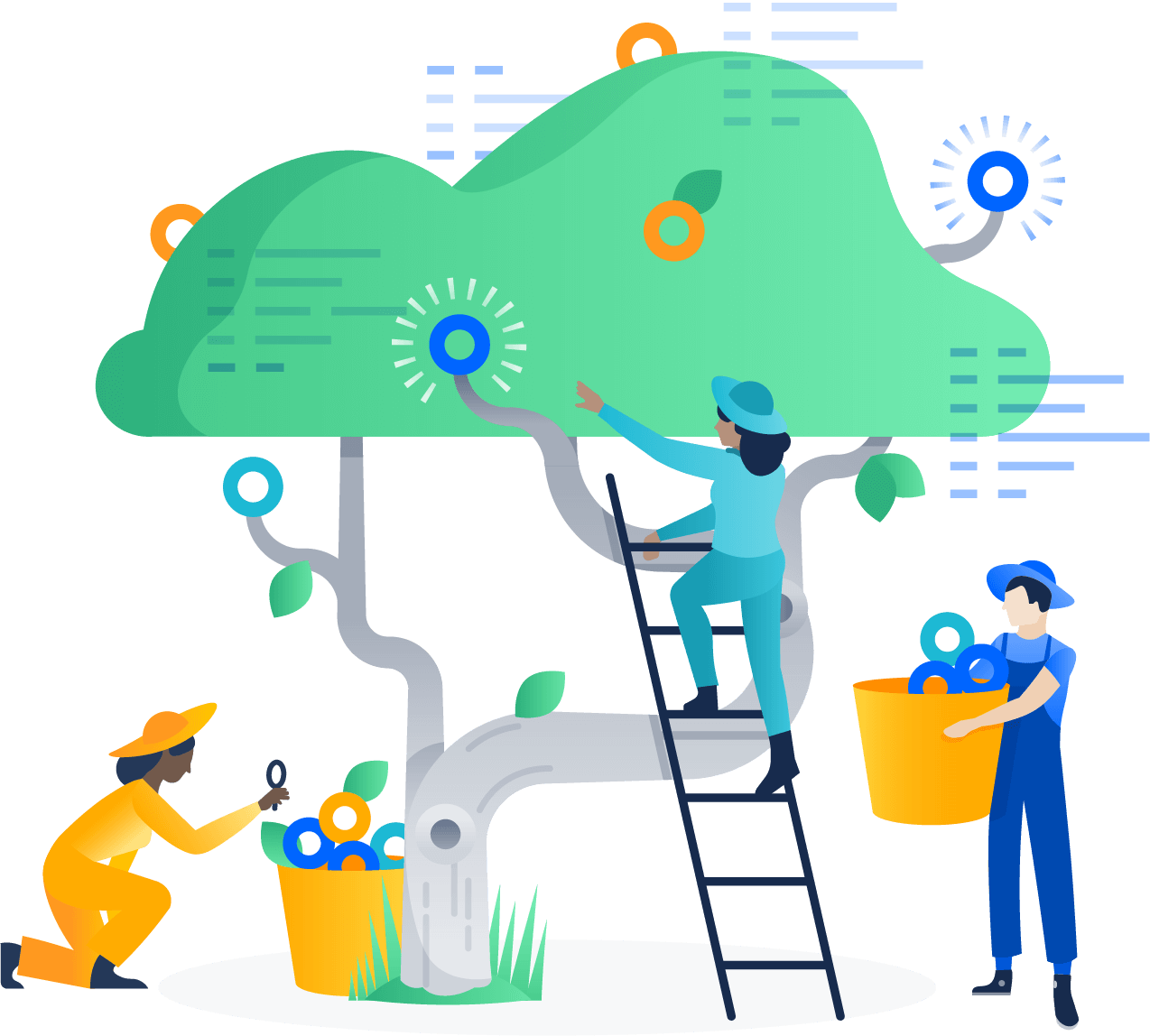
## BITBUCKET

### ¿Qué es Bitbucket?

Es una herramienta para colaboración de coding y con alojamiento en [Git](https://www.hostgator.mx/blog/git-sistema-de-control/?itm_source=blog&itm_medium=como-funciona-bitbucket&itm_campaign=git), Bitbucket es una excelente opción para equipos profesionales de gestión de proyectos e ingeniería de software(*Bitbucket vs GitHub: ¿Qué Repositorio de Código Es Mejor Para Sus Proyectos de Desarrollo?*, n.d.) .

### Principales características

* Integración directa con Jira, Bamboo, Crucible y Jenkins.
* La capacidad de importar repositorios desde Git, Codeplex, GoogleCode y SVN.
* Soporte de autenticación externa para GitHub, Google, [Facebook](https://kinsta.com/es/blog/estadisticas-facebook/) y [Twitter](https://kinsta.com/es/blog/twitter-marketing/).
* Integración masiva con [Trello](https://kinsta.com/es/blog/trello-vs-asana/)(*Bitbucket: Descubre Qué Es, Para Qué Sirve y Cómo Se Usa*, n.d.).
* Un cliente para Mac y Windows ([Sourcetree](https://www.sourcetreeapp.com/" \t "_blank)) y una aplicación para Android (Bitbeaker).
* Cifrado de datos.



# **CAPITULO II**

## GIT-FLOW

### ¿Qué es git-flow?

git-flow es un conjunto de extensiones para Git que proporcionan una abstracción de alto nivel. Están basadas en un post de Vincent Driessen, un ingeniero de software holandés, que describe el modelo de desarrollo basado en Git que lleva usando en varios proyectos. Posteriormente desarrolló un software, publicado en GitHub, en que implementaba la funcionalidad descrita en el post (Becerra, n.d.).

### Ramas principales

Esta metodología define que deben existir dos ramas principales:

* master
* develop

Rama master

Contiene cada una de las versiones estables del proyecto. Cualquier commit que subamos en esta rama debe estar preparado para que se pueda incluir en producción.

Rama develop

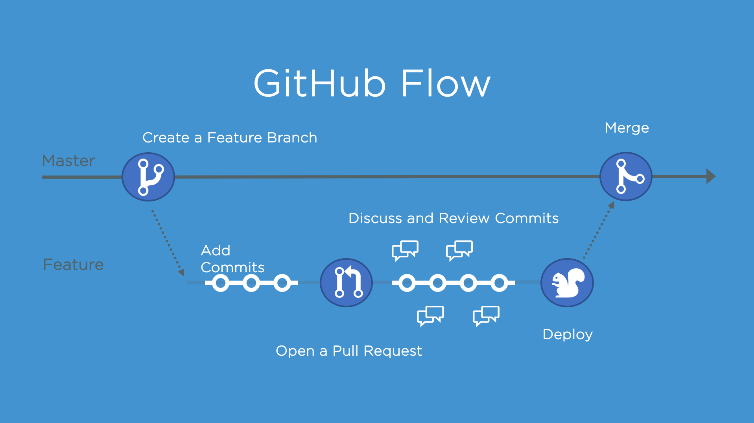
Contiene el código de desarrollo de la siguiente versión planificada del proyecto. En ella se incluirán cada una de las nuevas características que se desarrollen. Esta rama puede incorporarse tanto en una rama release (que veremos más adelante) como en la rama master, para su despliegue en producción

## GITHUB FLOW

### ¿Qué es GitHub Flow?

GitHub Flow se basa en un flujo de trabajo basado en branches que permite a equipos de desarrollo enfocarse principalmente en la entrega continua.

### Tipos de Branches



En esta estrategia de branching, en el repositorio tenemos dos tipos de branches:

* main (o master): el branch de código principal, es el que contiene el código que está listo para producción.
* features: los branches de funcionalidades que permiten el desarrollo en paralelo.

## GITLAB FLOW

### ¿Qué es GitLab Flow?

GitLab Flow se caracteriza por un flujo de trabajo bastante similar a GitHub Flow. La principal diferencia es que no solo permite el uso de release branches, sino que adiciona el uso de environment branches, por ejemplo, QA, Pre-Producción y Producción. Esto debido a que considera los casos donde una nueva funcionalidad no siempre puede implementarse en entornos de producción.

Al igual que GitHub Flow, GitLab Flow propone utilizar un master branch y feature branches. Una vez que terminamos el desarrollo en un feature branch, hacemos merge de nuevo con master.

* Con respecto a los branches por entornos, estos se crean a partir de *master* cuando estamos listos para implementar nuestra aplicación. Tener diferentes ramas por entorno nos permite configurar herramientas de CI y CD para que se despliegue automáticamente los commits realizados en cada una de esas ramas. Si detectamos un problema crítico, lo solucionamos en el *feature* o en *master*, una vez corregido, hacemos merge con las ramas de los entornos.
* Con respecto a los *release* branches, seguimos haciendo todo el trabajo en *feature* branches que luego hacemos merge a *master* al finalizar. Cuando nos aseguramos que *master* es lo suficientemente estable, es decir, ya hemos realizado todas las pruebas y corregido todos los errores, creamos un *release* branch que va a contener nuestra versión de código para desplegar. Si hay un problema crítico, primero lo solucionamos en master, realizamos cherry picking para hacer merge en release, es decir, seleccionamos las porciones de código que deseamos que contenga nuestro release.

## ONEFLOW

### ¿Qué es OneFlow?

One Flow 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.

# **CONCLUSIÓN**

Git y GitHub son herramientas fundamentales para el proceso de desarrollo de un proyecto o producto. Como hemos visto podemos usar otras alternativas, pero no debemos de suprimir estas herramientas que nos ayudarán a una mejor gestión del código. También GitHub podrás compartir tus habilidades técnicas, exhibiendo tus conocimientos y haciéndolos públicos en tu cuenta de perfil y además puedes aprender de otros desarrolladores más experimentados o compartir información y seguir progresando como programador.

# **BIBLIOGRAFIA**

Becerra, J. A. (n.d.). *git-flow*.

*Bitbucket: descubre qué es, para qué sirve y cómo se usa*. (n.d.). Retrieved June 4, 2022, from https://www.hostgator.mx/blog/como-funciona-bitbucket/?gclid=Cj0KCQjwheyUBhD-ARIsAHJNM-OF6YD-wFRQDkEAPI\_HQl\_MjOacRCtO-6OtEWCvV\_PQHldFgepV0FQaAs7eEALw\_wcB

*Bitbucket vs GitHub: ¿Qué repositorio de código es mejor para sus proyectos de desarrollo?* (n.d.). Retrieved June 4, 2022, from https://kinsta.com/es/blog/bitbucket-vs-github/

Castillo, L. (2017). *Conociendo GitHub Documentation Release 0.1*.

Gómez Bachiller, S. (n.d.). *INTRODUCCIÓN A GIT Y GITHUB-DÍA 1*.

MIGUEL ANGEL MARTINEZ. (n.d.). *El uso y herramientas de Git y GitLab*. Retrieved June 4, 2022, from https://www.redisybd.unam.mx/redisybd/pluginfile.php/1331/mod\_resource/content/1/GitGitlab.pdf

https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/tutorial-de-gitlab/