

SISTEMA DE CONTROL DE VERSIONES

Git, es un software de control de versiones diseñado por Linus Torvalds. La pregunta es ¿qué es control de versiones? Pues bien, se define como control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo es decir a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración, y para los que aún no les queda claro del todo, control de versiones es lo que se hace al momento de estar desarrollando un software o una página web. Exactamente es eso que haces cuando subes y actualizas tu código en la nube, o le añades alguna parte o simplemente le editas cosas que no funcionan como deberían o al menos no como tú esperarías.

Y, entonces ¿a qué le llamamos sistema de control de versiones? Muy sencillo, son todas las herramientas que nos permiten hacer todas esas modificaciones antes mencionadas en nuestro código y hacen que sea más fácil la administración de las distintas versiones de cada producto desarrollado; es decir Git.

GIT

Git fue creado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente, es decir Git nos proporciona las herramientas para desarrollar un trabajo en equipo de manera inteligente y rápida y por trabajo nos referimos a algún software o página que implique código el cual necesitemos hacerlo con un grupo de personas.

Algunas de las características más importantes de Git son:

- Rapidez en la gestión de ramas, debido a que Git nos dice que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente.
- Gestión distribuida; Los cambios se importan como ramas adicionales y pueden ser fusionados de la misma manera como se hace en la rama local.
- Gestión eficiente de proyectos grandes.
- Realmacenamiento periódico en paquetes.

ORDENES BÁSICAS

Iniciar un repositorio vacío en una carpeta específica.

```
git init
```

Añadir un archivo específico.

```
git add "nombre_de_archivo"
```

Añadir todos los archivos del directorio

```
git add .
```

Confirmar los cambios realizados. El "mensaje" generalmente se usa para asociar al commit una breve descripción de los cambios realizados.

```
git commit -am "mensaje"
```

Revertir el commit identificado por "hash_commit"

```
git revert "hash_commit"
```

Subir la rama(branch) "nombre_rama" al servidor remoto.

```
git push origin "nombre_rama"
```

Mostrar el estado actual de la rama(branch), como los cambios que hay sin hacer commit.

```
git status
```