

Sistema de Control de Versiones

Centro ADM

Barragán Jimenez Jonathan

7 de julio de 2018

Índice

1 Sistema de control de versiones

2 Git

- Comandos
- Ramas
- Mezclando Ramas

Índice

1 Sistema de control de versiones

2 Git

- Comandos
- Ramas
- Mezclando Ramas

Sistema de Control de Versiones

Nos ayuda a dar seguimiento a todos los cambios sobre un proyecto.

Repositorio

Es la colección de todas las versiones del proyecto.

Ademas de brindarnos información util:

- Descripción de cambios
- Fecha de los cambios
- Quien realizó los cambios

Commit

Un commit es la forma de integrar los nuevos cambios al repositorio.

Índice

1 Sistema de control de versiones

2 Git

- Comandos
- Ramas
- Mezclando Ramas

Git

Git es un sistema de control de versiones Distribuido.

- No existe un repositorio central
- Todos tienen una copia del repositorio
- Las contribuciones al proyecto no requieren de conexión a internet

Índice

1 Sistema de control de versiones

2 Git

- Comandos
- Ramas
- Mezclando Ramas

Crear un repositorio

Hay dos formas para crear un repositorio.

- `git init < nombre>`
Crea una carpeta donde estara el repositorio.
- `git init`
Crea el repositorio en el directorio actual.

Seguimiento de los cambios

Con el comando **git add** informamos de los nuevos cambios.
Con el comando **git commit** realizamos integramos los cambios al repositorio.

Configurar quien hace los cambios

```
git config --global user.name "nombre"  
git config --global user.email "correo"
```

Stagin area

Para saber el estado actual del proyecto utilizamos.
git status

Listar los commit

- Para listar todos los commit utilizamos el comando. **git log**
- Para saltar a un commit atras utilizamos. **git checkout <id commit >**
- Con el comando **git diff** podemos ver las diferencias entre commits.

Índice

1 Sistema de control de versiones

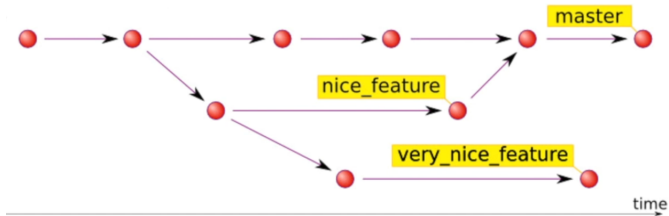
2 Git

- Comandos
- **Ramas**
- Mezclando Ramas

Rama

La idea de las ramas es poder trabajar en características del proyecto de forma independiente. La rama **master** es donde la versión estable del proyecto debe encontrarse (versión en producción).

Ramas



Comandos de ramas

- **git branch <nombreRama>** Con este comando creamos una rama nueva.
El nombre de las ramas debe hacer referencia a la nueva característica.
- **git branch** Con este comando listamos todas las ramas
- Con **git checkout <nombreRama>** Podemos saltar a otras ramas.
- Con **git checkout -b <nuevaRama>** Podemos crear y saltar a una nueva rama.
- Con **git branch -d <nombreRama>** Podemos borrar una rama.

Mezclar ramas

Una vez terminada la nueva característica o funcionalidad se debe mezclar a la rama **master**.

Para Mezclar una rama siempre se debe estar sobre la rama padre. Utilizamos el comando **git merge** <ramaHija> para mezclar las ramas.

Índice

1 Sistema de control de versiones

2 Git

- Comandos
- Ramas
- Mezclando Ramas

Merge

Existen 3 casos que pueden ocurrir a la hora de mezclar ramas.

Caso 1

Haces un merge de una rama en otra sin haber tocado los mismos archivos.

Los cambios se integran bien.

Caso 2

Has modificado el mismo archivo en ambas ramas pero editado en diferentes lineas.

Git detecta que no has modificado el mismo renglon y suele integrar bien.

Caso 3

Has modificado el mismo archivo en ambas ramas y has modificado la misma región de código.
Git genera un conflicto porque no sabe con qué modificación quedarse.