

# Capacitación PHP/SYMFONY

Módulo: GIT – GITLAB





# GIT

- Software de control de versiones
- Open Source
- Rastreador de contenido, principalmente utilizado para el almacenamiento de código
- Desarrollo en paralelo, cada colaborador posee una copia exacta del proyecto (repositorio)

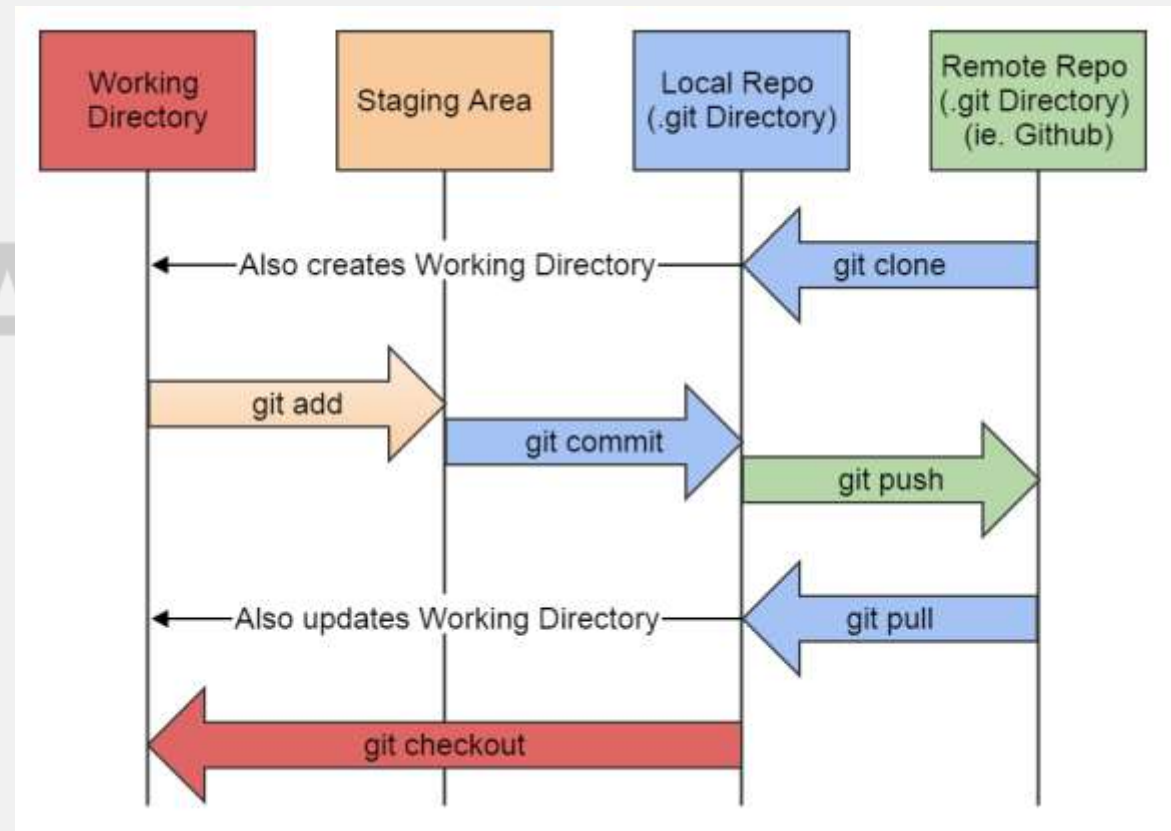
# Conceptos

- **Repositorio:** Donde se almacena todo el proyecto (local o remoto). Lleva un histórico de las versiones para comparación.
- **Branch (rama):** Se trata de una bifurcación o versión paralela de un repositorio.
- **Master/Main:** Por lo general, es la rama principal (estable).

- **Merge:** Fusión donde se toman los cambios de una rama (del mismo repositorio o de una bifurcación) y se aplican en otra.
- **Conflictos:** Diferencia que se produce al fusionar ramas y se encuentren modificaciones sobre la misma línea de código.
- **HEAD:** Puntero que nos indica en que lugar o instancia se encuentra nos repositorio.
- **Merge/Pull Request:** Solicitud para la incorporación de una rama a otra. Generalmente se realiza sobre master/main.
- **Origin:** Es el nombre por defecto que Git le da a un repositorio remoto cuando ejecutas git clone.

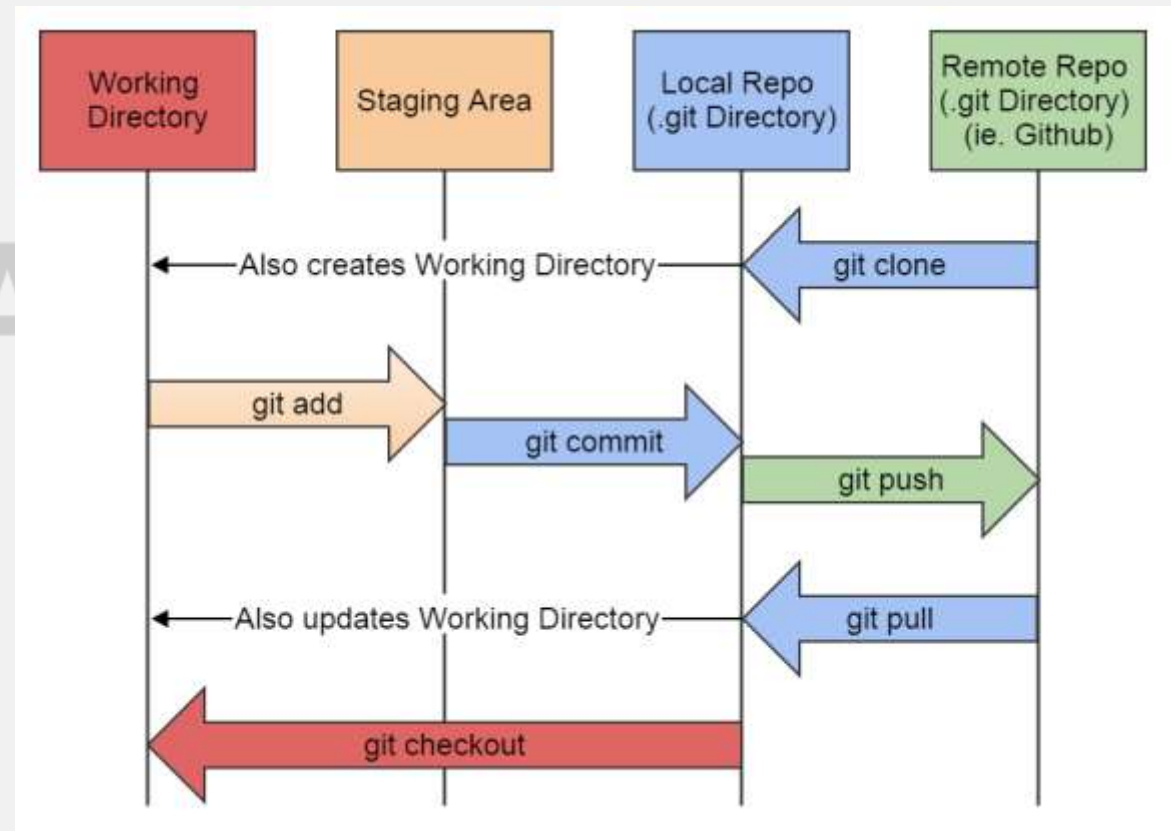
# Áreas de Trabajo

- **Working Directory:** Se encontrarán los archivos que han sido modificados vs su estado inicial.
- **Staging área:** Se encuentran los archivos preparados para el siguiente commit (índice).
- **Local repository:** Los archivos dentro del área de preparación (staging) pasan a formar parte del repositorio local luego del commit.
- **Remote repository:** Los cambios confirmados en el repositorio local se los sube al repositorio remoto.



# Estados de los archivos

- **Modified (Modificado):** El fichero o los ficheros han sido modificados y aún no están almacenados.
- **Staged (Preparado):** Los datos se encuentran a la espera de ser almacenados (committed).
- **Committed (Confirmado):** Los datos se encuentran almacenados en el repositorio.



# Archivo .gitignore

- El archivo .gitignore es un archivo de texto que le dice a Git qué otros archivos y carpetas de un proyecto debe ignorar.

```
# Ignorar archivos del sistema de Mac  
.DS_store
```

```
# Ignorar carpeta node_modules  
node_modules
```

```
# Ignorar todos los archivos de texto  
*.txt
```

```
# Ignorar archivos relacionados con claves de una API  
.env
```

```
# Ignorar archivos de configuración SASS  
.sass-cache
```

# Comandos básicos de Git

```
//Inicializar repositorio
git init

//Inspección del repositorio
git status

//Agregar cambios al área de staging
git add . (todos)
git add archivo.txt (solo el archivo)

//Confirmar cambios en staging
git commit -m "Mensaje commit"
git commit --amend (modifica el ultimo commit
realizado)
git commit -am "Mensaje commit" (agrega todos
los archivos en staging y genera el commit)

//Ver historial de commits
git log
git log --oneline
```

```
//Visualizar ramas (branches) existentes
git branch

//Crear nueva rama branch
git branch nombreRama

//Moverse a otra rama
git checkout nombreRama
git checkout -b nombreRama (crea y se mueve a
la rama)

//Mostrar las modificaciones en los archivos
git diff (muestra las diferencias de todos
los archivos con respecto al último commit)
git diff ./ruta/archivo.txt (cambios de un
archivo con respecto al último commit)
git diff hash1 hash2 (diferencias entre dos
commits)
```

# Comandos básicos de Git

```
//Clonar repositorio
git clone url-repositorio

//Asociar repositorio remoto a repositorio local
git remote add origin url-repositorio

//Configurar Nombre y Usuario
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
(la opción --global aplicará para todos los repositorios)

//Descargar cambios del repositorio remoto (commits, histórico, referencias) al repositorio local
git fetch
git fetch repoRemoto
Git fetch repoRemoto ramaRemota

//Fusionar cambios entre ramas
git merge ramaFusionar
git merge --squash ramaFusionar (combina varias confirmaciones en una sola)
```

```
//Recuperar cambios de repositorio remoto y fusionarlo en rama local
git pull

//Pasar cambios de repositorio local al remoto
git push repoRemoto ramaLocal:ramaRemota

//Resetear a rama diferente o commit anterior sin preservar los cambios realizados
git reset --hard nombreRama
Git checkout hash-commit

//Almacenar temporalmente los cambios
git stash save "Texto descriptivo"
git stash list (lista los stash)
git stash pop stash@{numero-stash} (aplicar cambios del stash, se elimina el stash)
git stash apply {numero-stash} (aplicar cambios del stash, mantiene el stash)
git stash show (ver cambios del stash)
git stash drop stash@{numero-stash} (elimina el stash)
```



# GITLAB

- Al igual que GitHub, GitLab es un gestor de repositorios Git que permite que los equipos colaboren en códigos informáticos.
- Es una plataforma completamente de código abierto.
- Las funciones nativas de integración continua y entrega continua (Gitlab CI/CD) permiten el desarrollo, las pruebas y el despliegue continuos de una aplicación

