



# Git y Github

Contenidos

**CÓNQUER** **BLOCKS**

## Bloque de contenidos

Programación didáctica correspondiente al bloque de Git y Github de Conquer Blocks. Este módulo será común a todos los másters de desarrollo de la escuela ya que es una utilidad imprescindible para llegar a ser un desarrollador profesional.

### Módulo 0: La importancia de Git y Github por Mouredev

#### Módulo 1: Introducción a Git y Control de Versiones

- ¿Qué es el control de versiones?
- Beneficios del control de versiones.
- Introducción a Git y su historia.
- Conceptos básicos: repositorio, commit, rama, clonar, fusionar y bifurcar.
- Instalación y configuración inicial de Git.

#### Módulo 2: Uso Básico de Git

- Inicialización de un repositorio local.
- Realizar y deshacer commits.
- Ver historial de commits y diferencias.
- Ignorar archivos y directorios con .gitignore.
- Trabajar con ramas (branches).

#### Módulo 3: Trabajo Colaborativo Local

- Fusionar ramas.
- Resolver conflictos de fusión.
- Revertir cambios y utilizar git reset.
- Trabajar con etiquetas (tags).
- Stash: guardar y aplicar cambios temporales.

#### Módulo 4: Introducción a GitHub

- ¿Qué es GitHub?
- Creación de una cuenta y configuración del perfil.
- Creación de un repositorio en GitHub.
- Clonar un repositorio remoto.
- Sincronizar cambios con un repositorio remoto.

#### Módulo 5: Trabajo Colaborativo en GitHub

- Contribuciones mediante pull requests.
- Comentar y revisar pull requests.
- Resolver conflictos en pull requests.
- Aprobar y fusionar pull requests.
- Uso de issues y proyectos.

### Módulo 6: Buenas Prácticas y Consejos

- Escribir mensajes de commit descriptivos.
- Mantener commits atómicos y coherentes.
- Evitar cambios directos en la rama principal.
- Uso de ramas de características.
- Mantener un historial limpio y comprensible.

### Módulo 7: Integración Continua y Despliegue

- Integración continua con GitHub Actions.
- Automatización de pruebas y flujos de trabajo.
- Despliegue automatizado a través de Git/GitHub.

### Módulo 8: Casos de Uso Específicos

- Versionado semántico.
- Gestión de proyectos con Git/GitHub.
- Uso de Git/GitHub en proyectos de código abierto.
- Gestión de archivos grandes con Git LFS.