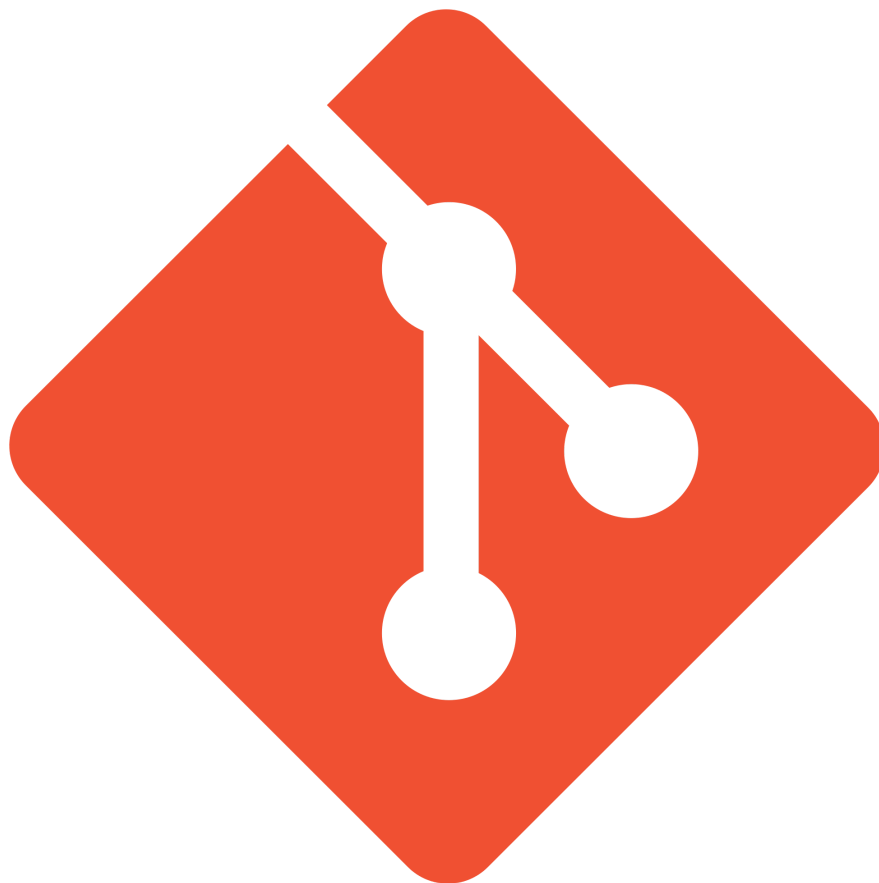


# **Git y Github**

## **Guia de comandos**

**Prácticas profesionalizantes 1**

**Profesor Galo Olguin**



<b>Configuración.....</b>	<b>3</b>
<b>Control de versiones.....</b>	<b>4</b>
<b>Ramas / Branch.....</b>	<b>5</b>
<b>Gestión del historial de cambios.....</b>	<b>6</b>

## Configuración

Para la configuración utilizar los datos de la cuenta de github en el caso de trabajar con repositorios remotos.

En caso de querer tener una configuración diferente para cada repositorio evitar la opción **--global**.

**git --version**

Versión de git instalada

**git config --global user.name nombre**

Agrega el nombre de usuario de forma global (todos los repositorios).

**git config --global user.email**

Agrega el email de forma global (todos los repositorios).

**git config --global user.password**

Agrega la contraseña de forma global (todos los repositorios), actualmente se utilizan los personal access tokens.

**git config --global core.editor "code -wait"**

Establece visual studio code como editor predeterminado.

**git config --global -e**

Ver toda la configuración global en el editor predeterminado.

**git config --global core.autocrlf true**

Establecer caracteres especiales.

TRUE para windows

INPUT linux y mac

**git config --global init.defaultbranch nombre**

Establece el nombre de las ramas a iniciar un nuevo repositorio

**git config --global --unset**

Elimina todas las credenciales. Al trabajar en windows revisar el "administrador de credenciales" o "credential manager", luego "credenciales de windows" y buscar lo que diga git o github y click en quitar o eliminar.

## Control de versiones

**git init**

Inicia un repositorio.

**git add archivo**

Agrega las modificaciones realizadas en el **archivo** al stage.

**git commit -m "mensaje"**

Confirma las modificaciones realizadas de los archivos que se encuentren en el stage agregando un **mensaje** de referencia.

**git merge rama**

Mezcla las modificaciones realizadas en una rama secundaria en la rama en la que se esté posicionado.

**git remote add nombreRepositorio linkRepositorioRemoto**

Agrega una conexión entre el repositorio local y remoto. En **nombreRepositorio** se especifica el **alias** con el que se va a hacer referencia al repositorio remoto y en **linkRepositorioRemoto** se especifica el link del repositorio.

**git remote -v**

Muestra los repositorios remotos agregados al repositorio local

**git remote rm aliasRepositorioRemoto**

Elimina la conexión entre el repositorio local y remoto establecidas según el alias del repositorio remoto especificado en **aliasRepositorioRemoto**.

**git pull aliasRepositorioRemoto ramaRepositorioRemoto**

Descarga los cambios existentes del repositorio remoto especificado en **aliasRepositorioRemoto** según la rama especificada en **ramaRepositorioRemoto**.

**git push aliasRepositorioRemoto ramaRemota**

Sube las modificaciones confirmadas al repositorio remoto especificado en **aliasRepositorioRemoto** y a la rama especificada en **ramaRemota** ingresados.

**git tag**

Muestra los tags generados

**git tag -a nombreTag -m "descripción"**

Genera un nuevo tag según el **nombreTag** y **descripción** ingresada.

**git push repositorioRemoto nombreTag**

Sube al **repositorioRemoto** el tag indicado en **nombreTag**.

## Ramas / Branch

**git branch**

Muestra las ramas creadas.

**git branch nombreRama**

Crea una nueva rama con el nombre indicado en **nombreRama**.

**git branch -m nombreRama nuevoNombreRama**

Modifica el nombre de una rama especificada en **nombreRama** según el nuevo nombre que se le ingrese en **nuevoNombreRama**.

**git branch -d nombreRama**

Elimina la rama que se le especifique en **nombreRama**.

### **git checkout**

Es un comando que según lo que se le ingrese nos permitirá desplazarnos a una versión que se le especifique.

### **git checkout -b nombreRama**

Crea una nueva rama con los archivos que contenga la rama en la que se esté alojado.

### **git checkout nombreRama**

Actualiza los archivos a la versión especificada según la rama que se especifique en **nombreRama**.

### **git checkout hash**

Actualiza los archivos a la versión especificada según el **hash** ingresado.

### **git switch -**

Actualiza los archivos a la versión de la rama anterior a utilizar **git checkout hash**.

### **git push -u nombreRepositorioRemoto nombreRama**

Sube al repositorio remoto indicado en **nombreRepositorioRemoto** la rama especificada en **nombreRama**.

## Gestión del historial de cambios

### **git log**

Muestra el historial completo de commits realizados.

### **git log --oneline**

Muestra el historial en una línea de cada uno de los commits realizados.

### **git reset**

Deshace los cambios confirmados en un repositorio. Se comporta de forma distinta dependiendo de las opciones con los que se lo utilice.

```
git reset --soft HEAD~
```

Deshace el último commit realizado, dejando los archivos en el stage.

```
git reset --hard HEAD~
```

Deshace el último commit y también **borra los cambios realizados dejando el stage vacío.**