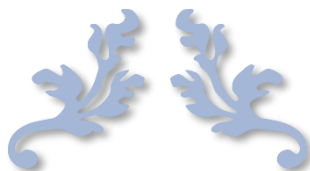




COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI



PRÁCTICA 1: PRIMERA PARTE

Programación de Aplicaciones Telemáticas



ALEJANDRA BOTÍN LEHM

MARIO VIDAL DOMÍNGUEZ

3ºB GITT-BA

26 DE ENERO DE 2025

ÍNDICE

INTRODUCCIÓN	2
Git Clone	2
Git Status	2
Git Add.....	2
Git Commit.....	3
Git Push.....	3
Git Checkout.....	4
CONCLUSIÓN	4

INTRODUCCIÓN

En esta práctica hemos trabajado con Git y GitHub para gestionar un repositorio de código y entender los conceptos básicos de control de versiones. Comenzamos haciendo un **fork** de un repositorio en GitHub para crear una copia en nuestra cuenta, y luego lo clonamos en nuestro entorno local utilizando **GitHub Codespaces**. Desde allí, editamos archivos como el README.md, utilizando comandos fundamentales de Git como `git clone`, `git status`, `git add`, `git commit`, `git push` y `git checkout`.

A continuación, se muestra el uso de estos comandos y se comenta que hace cada uno en cada caso:

Git Clone

```
● @AlejandraBotin →/workspaces/p1-fork (main) $ git clone https://github.com/AlejandraBotin/p1-fork.git
Cloning into 'p1-fork'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (6/6), done.
```

- El comando descarga una copia completa del repositorio remoto p1-fork desde GitHub al entorno de trabajo (en este caso, Codespaces). Esto incluye todo el historial del repositorio y su contenido actual, creando una carpeta llamada p1-fork donde se almacena el proyecto.

Git Status

```
● @AlejandraBotin →/workspaces/p1-fork (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  p1-fork/
```

- El comando `git status` muestra el estado actual del repositorio. En este caso:
 - ❖ **On branch main:** Trabajamos en la rama principal `main`.
 - ❖ **Your branch is up to date with 'origin/main' .:** La rama local está sincronizada con la rama remota en GitHub.
 - ❖ **Untracked files:** La carpeta `p1-fork/`, Git no la está rastreando porque no se han añadido al área de preparación (staging area). Para rastrearlos, se usa a continuación `git add`.

Git Add

```
● @AlejandraBotin →/workspaces/p1-fork (main) $ git add README.md
● @AlejandraBotin →/workspaces/p1-fork (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    README.md
    new file:   p1-fork

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
  (commit or discard the untracked or modified content in submodules)
    modified:   p1-fork (modified content)

● @AlejandraBotin →/workspaces/p1-fork (main) $ git add .
```

- Después de ejecutar el comando `git add README.md`, se incluyó el archivo `README.md` en el área de preparación (staging area), y el comando `git status` mostró los cambios preparados y los no preparados. Específicamente:
 1. Cambios preparados para confirmar (Changes to be committed):
 - o **deleted: README.md:** El archivo `README.md` se marcó como eliminado en el área de preparación.
 - o **new file: p1-fork:** La carpeta `p1-fork` fue detectada como un nuevo archivo o directorio.
 2. Cambios no preparados para confirmar (Changes not staged for commit):
 - o **modified: p1-fork:** El contenido de la carpeta `p1-fork` había sido modificado, pero aún no estaba incluido en el área de preparación.
- Después, se ejecutó el comando `git add.`, que agrega todos los archivos y carpetas modificados o no rastreados (untracked) al área de preparación. Este comando se ha usado para incluir los cambios de `p1-fork` que no estaban preparados para confirmar.

Git Commit

```
● @AlejandraBotin →/workspaces/p1-fork (main) $ git commit -m "Descripción de los cambios realizados"
[main 3c823cf] Descripción de los cambios realizados
2 files changed, 1 insertion(+), 24 deletions(-)
delete mode 100644 README.md
create mode 160000 p1-fork
```

- El comando `git commit -m "Descripción de los cambios realizados"` ha registrado oficialmente los cambios en el historial del repositorio local, acompañado de un mensaje descriptivo. El terminal indica:
 - ❖ **[main 3c823cf] Descripción de los cambios realizados:**
 - El commit se ha realizado en la rama `main` y está identificado con el hash único `3c823cf`.
 - ❖ **2 files changed, 1 insertion(+), 24 deletions(-):**
 - Indica que se realizaron cambios en dos archivos.
 - Hubo **1 línea añadida** (1 insertion) y **24 líneas eliminadas** (24 deletions).
 - ❖ **Detalles de los archivos:**
 - **delete mode 100644 README.md:** El archivo `README.md` fue eliminado porque al clonar la carpeta lo teníamos dos veces
 - **create mode 160000 p1-fork:** La carpeta `p1-fork` fue añadida al repositorio como un nuevo archivo o directorio.

Git Push

```
● @AlejandraBotin →/workspaces/p1-fork (main) $ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (2/2), 259 bytes | 259.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AlejandraBotin/p1-fork
07720b5..3c823cf main -> main
```

- El comando `git push origin main` ha subido los cambios confirmados desde el repositorio local al repositorio remoto en GitHub. Aquí está el desglose del mensaje:
 - ❖ **Enumerating objects:**
 - Git está contando los cambios realizados y confirmados (en este caso, 3 objetos, que incluyen las modificaciones y metadatos relacionados).
 - ❖ **Counting objects y Writing objects:**
 - Los objetos fueron empaquetados (comprimidos) y enviados al servidor remoto (GitHub).
 - La transferencia fue exitosa con un total de 259 bytes.

- ❖ To <https://github.com/AlejandraBotin/p1-fork>:
 - Esto confirma que los cambios se han subido correctamente al repositorio remoto alojado en la URL proporcionada.
- ❖ `07720b5..3c823cf main -> main`:
 - Muestra el rango de commits que se subieron, indicando que los cambios realizados en el commit `3c823cf` ya están disponibles en la rama `main` del repositorio remoto.

Git Checkout

```

● @AlejandraBotin → /workspaces/p1-fork (main) $ git checkout -b mi-nueva-rama
Switched to a new branch 'mi-nueva-rama'
○ @AlejandraBotin → /workspaces/p1-fork (mi-nueva-rama) $

```

- El comando `git checkout -b mi-nueva-rama` ha creado una nueva rama llamada `mi-nueva-rama` y automáticamente ha cambiado a esa rama. Aquí está el desglose:
 - ❖ `git checkout -b mi-nueva-rama`:
 - `-b` indica que se debe crear una nueva rama.
 - `mi-nueva-rama` es el nombre de la nueva rama.
 - ❖ `Switched to a new branch 'mi-nueva-rama'`:
 - Confirma que la nueva rama fue creada correctamente y que ahora estás trabajando en esa rama.
 - ❖ `(mi-nueva-rama)` en la línea de comandos:
 - Indica que ahora estamos en la rama `mi-nueva-rama`, en lugar de la rama `main`.

CONCLUSIÓN

El ejercicio permitió entender el flujo básico de trabajo con Git, desde la creación de un fork, la clonación de un repositorio y la gestión de archivos y ramas, hasta la sincronización de cambios entre el repositorio local y remoto. Esto nos proporciona una base sólida para gestionar proyectos de manera organizada, realizar cambios de forma segura y facilitar la colaboración con otros desarrolladores. La experiencia destaca la importancia de documentar y estructurar adecuadamente los cambios realizados, lo que es esencial para un desarrollo eficiente y transparente.