Resumen clase 5

¿Qué es una ruta?

Una ruta (o path) es la dirección de una carpeta o archivo dentro de tu computadora.
Por ejemplo:

```
C:\Usuarios\Juan\Documentos\Proyectos\mi-app
```

- C: → Disco principal
- Usuarios\Juan\Documentos\Proyectos\mi-app → Camino de carpetas hasta llegar a mi-app

¿Qué significa subir un nivel?

Cuando estás dentro de una carpeta, "subir un nivel" significa volver a la carpeta anterior.
Por ejemplo, si estás en mi-app, y subís un nivel, vas a Proyectos.

Estructura de ejemplo

```
C:\
└── Usuarios
    └── Juan
        ├── Escritorio
        └── Documentos
            └── Proyectos
                ├── mi-app
                └── otra-app
```

Comandos comunes en git bash

Comando Git Bash	¿Para qué sirve?
`ls`	Lista archivos y carpetas
`ls -a`	Lista todo, incluyendo archivos ocultos
`cd nombre`	Entra en una carpeta
`cd ..`	Sube un nivel
`pwd`	Muestra la ruta actual
`mkdir nombre`	Crea una nueva carpeta
`rm archivo`	Borra un archivo
`rm -r carpeta`	Borra una carpeta y su contenido
`clear`	Limpia la pantalla

Ejemplo práctico con comandos

1. Estás en esta ruta:

```
C:\Usuarios\Juan\Documentos\Proyectos\mi-app
```

2. Querés subir un nivel (ir a la carpeta Proyectos):

```
cd ..
```

3. Querés entrar en otra-app:

```
cd otra-app
```

4. Querés ver dónde estás:

```
pwd
```

5. Querés ver qué hay en la carpeta actual:

```
ls -a
```

Git

Repositorio local vs. repositorio remoto

- Repositorio local: es el repositorio que tenés en tu computadora. Acá hacés tus cambios, agregás archivos y hacés commits.
- Repositorio remoto: es una copia del repositorio que está en un servidor (por ejemplo GitHub, GitLab o Bitbucket), y sirve para compartir tu proyecto con otros.

Cuando conectas tu repositorio local a un remoto, por convención ese remoto se llama origin.

—

Estados de los archivos en Git

Un archivo puede estar en diferentes estados dentro de Git:

1. **Untracked**: Git no lo está siguiendo todavía. Es un archivo nuevo.
2. **Tracked**: Git lo está siguiendo. Puede estar:
   - Sin cambios
   - Modificado (cambiaste algo)
   - En staging (preparado para commit)

3. **Added (staged)**: significa que lo agregaste al área de preparación con git add.

¿Qué hace git add?

Cuando ejecutas git add archivo o git add ., le decís a Git:
“Quiero guardar este archivo en el próximo commit.”
Pasa de estar solo modificado a estar en stage, listo para ser guardado.

¿Qué es un commit?

Un commit es un punto de guardado en el historial del proyecto, este debe poseer un mensaje.
Guarda el estado exacto de tus archivos en ese momento.
Por eso se lo llama punto de guardado: podés volver a él si algo falla más adelante.

Se hace con:

```
git commit -m "Mensaje descriptivo"
```

¿Qué es git push?

Una vez que tenés commits en tu repositorio local, podés subirlos al repositorio remoto con:

```
git push origin nombre-rama
```

Esto sincroniza tus cambios con lo que está en GitHub o GitLab.

¿Qué es una rama (branch)?

Una rama es una línea paralela de trabajo. Sirve para trabajar en nuevas funcionalidades sin afectar la rama principal.

- Rama secundaria: cualquier rama que no sea la principal. Por ejemplo: feature/login, fix/navbar, hotfix/404.
- Default branch: es la rama principal del repositorio. Por convención suele ser main o master.

¿Qué es la rama master?

- Es la rama principal por convención (aunque hoy en día se también se suele usar main).
- Se la considera la más estable, limpia y funcional, porque en ella está el código listo para producción.
- Las nuevas funcionalidades se desarrollan en ramas secundarias y luego se integran a master con un merge o pull request.

¿Qué es desarrollo y qué es producción?

- Desarrollo (development): es donde trabajás y probás funcionalidades nuevas. Puede tener errores.
- Producción (production): es la versión estable que usan los usuarios finales. Debe estar probada y funcionando bien.
-A veces se usan ramas llamadas dev y main o master para representar estos entornos.

Flujo típico de trabajo en Git

1. **Clonás el repositorio remoto:**

```
git clone URL-del-repo
```

2. **Te movés a la rama principal y creás una rama nueva, dos posibles opciones:**

Salis de la rama actual y vas hacia master
```
git checkout master
```

Salis de la rama actual, haces una copia identica y vas hacia esta, ahora llamada nombre-rama
```
git checkout -b nombre-rama
```

3. **Desarrollás tu funcionalidad:**
   - Modificás archivos
   - git add .
   - git commit -m "Agrega X"

4. **Subís la rama al remoto:**

```
git push origin nombre-rama
```

5. **Abrís un Pull Request (PR) o Merge Request (MR):**
   - Proponés unir tu rama a master o main
   - Otro miembro del equipo revisa el código

6. **Si se aprueba, se hace el merge:**

- Tu código se integra a la rama principal

7. **La rama master ahora tiene tu funcionalidad lista para producción.**

Comandos básicos de Git

- **git init:** inicia un repositorio Git en la carpeta actual.
- **git add . :** agrega todos los archivos al área de preparación (staging) para ser confirmados.
- **git commit -m "mensaje":** guarda los cambios preparados con un mensaje descriptivo.
- **git remote add origin "url":** vincula tu repositorio local con uno remoto (por ejemplo en GitHub).
- **git push -u origin rama:** sube tus cambios al repositorio remoto, y guarda la rama como predeterminada para futuros push.
- **git clone "url":** clona un repositorio remoto en tu máquina.
- **git pull:** descarga y aplica los últimos cambios del repositorio remoto en tu rama actual.
- **git pull origin master:** descarga y aplica los cambios desde la rama master específica del repositorio remoto.
- **git config --global user.name "Tu Nombre":** configura tu nombre para los commits (globalmente).
- **git config --global user.email "tu@email.com":** configura tu email para los commits (globalmente).

En caso de no querer configurar globalmente, por el contrario se configura localmente sólo en el proyecto que estás trabajando, si quitas **--global** .

Comandos adicionales útiles

- **git config --global init.defaultBranch master**: cambia la rama por defecto a master en lugar de main al usar git init.
- **git reset**: quita archivos del área de preparación o revierte cambios (según cómo se use).
- **git merge --abort**: cancela una operación de merge en caso de conflicto.
- **git log --oneline**: muestra el historial de commits de forma resumida (una línea por commit).
- **git checkout -b nombre-rama**: crea una nueva rama y te mueve a ella al mismo tiempo.

Otros comandos importantes

- **git status**: muestra el estado actual del repositorio (qué cambió, qué falta agregar, etc.).
- **git branch**: lista las ramas locales disponibles.
- **git branch nombre**: crea una nueva rama con ese nombre.
- **git checkout nombre-rama**: cambia a otra rama existente.
- **git merge nombre-rama**: une otra rama con la rama actual.

- **git stash**: guarda temporalmente los cambios sin hacer commit (útil si necesitás cambiar de rama rápido).
- **git stash pop**: recupera los cambios que guardaste con git stash.

Diagrama de clase