

INTRUCCIONES BÁSICAS PARA UTILIZAR GITHUB EN EL CONTEXTO DEL TRABAJO DE PLSQL:

Primero, ¿qué es git?. Git es un sistema de control de versiones que permite gestionar cambios en el código de manera eficiente. Aquí encontraréis los comandos básicos para trabajar en equipo con el repositorio de la entrega.

Vamos a utilizar lo más básico de git, por ejemplo, en esta práctica no es necesario crear ramas por lo que no se cubren en este tutorial por simplificar, pero si estáis familiarizados con ellos y os ayuda en vuestro trabajo, no dudéis en usarlas. En resumen, lo más básico es que: cada persona clone el repositorio (git clone); antes de ponerse a trabajar cada día descarga los cambios más recientes (git pull); hace las modificaciones que sean en el código y guardan los cambios localmente (git add → git commit); y finalmente sube los cambios para que los vean las demás personas del equipo (git push).

1) Crear el repositorio en Github y añadir colaboradores:

- el resto de personas del equipo
- la profesora (asmamolar@ubu.es)

2) Clonar repositorio en vuestra máquina virtual desde un terminal.

Cada persona necesitaréis crear un TOKEN de acceso que identifique vuestro usuario de Github. Ver “pasos para crear TOKEN” más abajo:

Comando para clonar el repositorio:

```
git clone https://<TU_USUARIO_GITHUB>:<TU_TOKEN>@github.com/<OWNER>/<REPO>.git
```

Ejemplo para el usuario asmamolar:

```
git clone https://asmamolar:xpzf123456789abcdefghijklmnopqrstuvwxyz@github.com/propietario/trabajo2ABD.git
```

3) Trabajar con el repositorio.

Os recomiendo seguir estos pasos para trabajar en equipo sobre el repositorio.

• Obtener cambios más recientes

(por defecto trabajamos sobre la rama main si no indicamos nada)

```
git pull
```

(Esto descarga la versión más reciente del repositorio para evitar conflictos.)

• Ver cambios pendientes

```
git status
```

(antes de hacer un commit, con este comando puedes revisar los archivos modificados)

Si no hay cambios, nos dirá algo como “Tu rama está actualizada con <nombre rama>”

• Añadir todos los archivos modificados

Antes de guardar los cambios, tienes que decirle a Git qué archivos quieres incluir.

- Todos (indicado con el punto):

```
git add .
```

- Un archivo específico

```
git add <nombre_archivo>
```

- **Guardar los cambios con un mensaje**

```
git commit -m "Descripción de los cambios"
```

(*commit* sirve para guardar el estado actual de los archivos de tu proyecto en el historial. De manera que puedas seguir avanzando sin perder cambios anteriores. Cada vez que realizas un commit, estás registrando los cambios en tu código de forma permanente, aunque posteriormente deshagas esos cambios siempre podrás volver a cada versión “commiteada”).

- **Subir cambios a GitHub**

```
git push
```

(Esto actualiza el repositorio en GitHub con tus cambios.)

* No confundir *commit* con *push*. El comando *commit* guarda los cambios a nivel local, es decir en tu repositorio local, en tu máquina. Mientras que *push* los sube al repositorio remoto (en vuestro caso Github).

Pasos para crear token de acceso personal (PAT) :

Los access token son usados como alternativa para reemplazar la contraseña al autenticarse en GitHub desde la línea de comandos, y actualmente son necesarios para clonar un repositorio desde un sistema Linux. Para crear un TOKEN que identifique a tu usuario:

- Ve a [GitHub → Settings → Developer settings → Personal access tokens](#)
- Crea un nuevo token con los permisos adecuados, en vuestro caso basta con que marqués la opción *repo* (para acceso a repos privados)

Note

hewtoken

What's this token for?

Expiration

This token expires **on Thu, Apr 17 2025**. To set a new expiration date, you must [regenerate the token](#).

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry

- Copia el token y guárdalo para poder insertarlo en el comando para clonar el repositorio.