Herramientas de Desarrollo

Semana 03
Manejo de repositorios, ramas(branching) y commits

Inicio

¿Tienen alguna consulta o duda sobre la clase anterior?

Logro de la Unidad

Al finalizar la unidad, el estudiante gestiona los sistemas de control de versiones para el desarrollo de una solución de software.

Imagen obtenida de: https://www.euroschoolindia.com/wp-content/uploads/2023/08/impact-of-school-leadership.jpg

Utilidad

- ¿Se puede guardar toda la información en un solo espacio o lugar?

Universidad Tecnológica del Perú

contenido

1. Introducción
2. Manejo de Repositorios
3. Ramas (Branching)
4. Commits

Transformación

Universidad Tecnológica del Perú

1 Introducción

Introducción

Git es una herramienta poderosa para el control de versiones, y el manejo de repositorios, ramas y commits es fundamental para la colaboración y el flujo de trabajo en proyectos de software.

A continuación, te explico cómo realizar estas acciones de manera efectiva con ejemplos y una serie de preguntas teóricas y prácticas.

2 Manejo de Repositorios

Manejo de Repositorios en Git

Inicializar un repositorio nuevo: Para crear un repositorio Git vacío en tu máquina local:

git init

Esto crea un nuevo repositorio en el directorio actual.

Clonar un repositorio remoto: Si deseas obtener una copia de un repositorio remoto, puedes usar:

git clone https://github.com/usuario/repositorio.git

Esto descarga una copia del repositorio en tu máquina local.

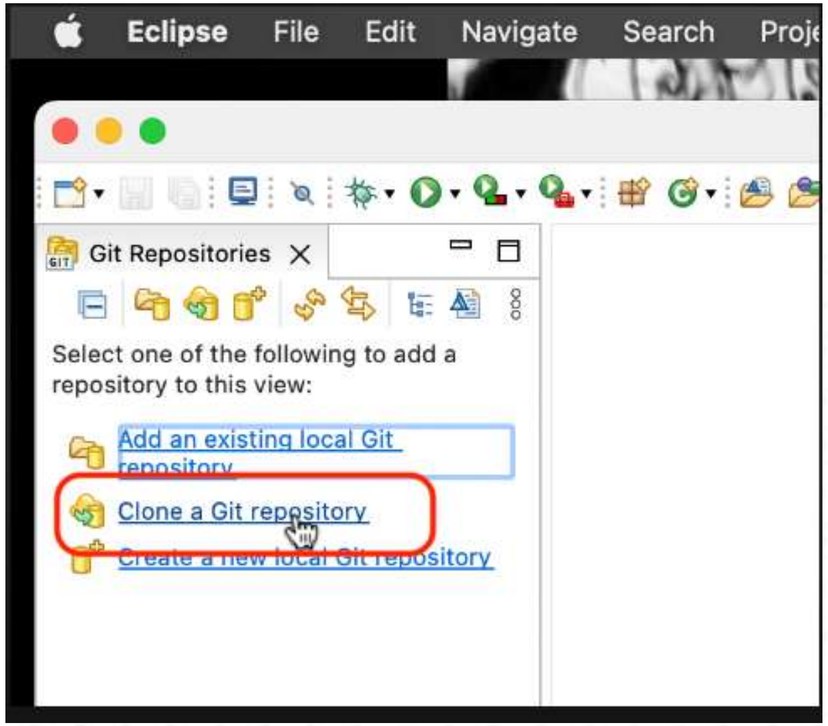
…Manejo de Repositorios en Git

Ver el estado del repositorio: Para revisar el estado del repositorio (archivos modificados, pendientes de agregar o commitear):

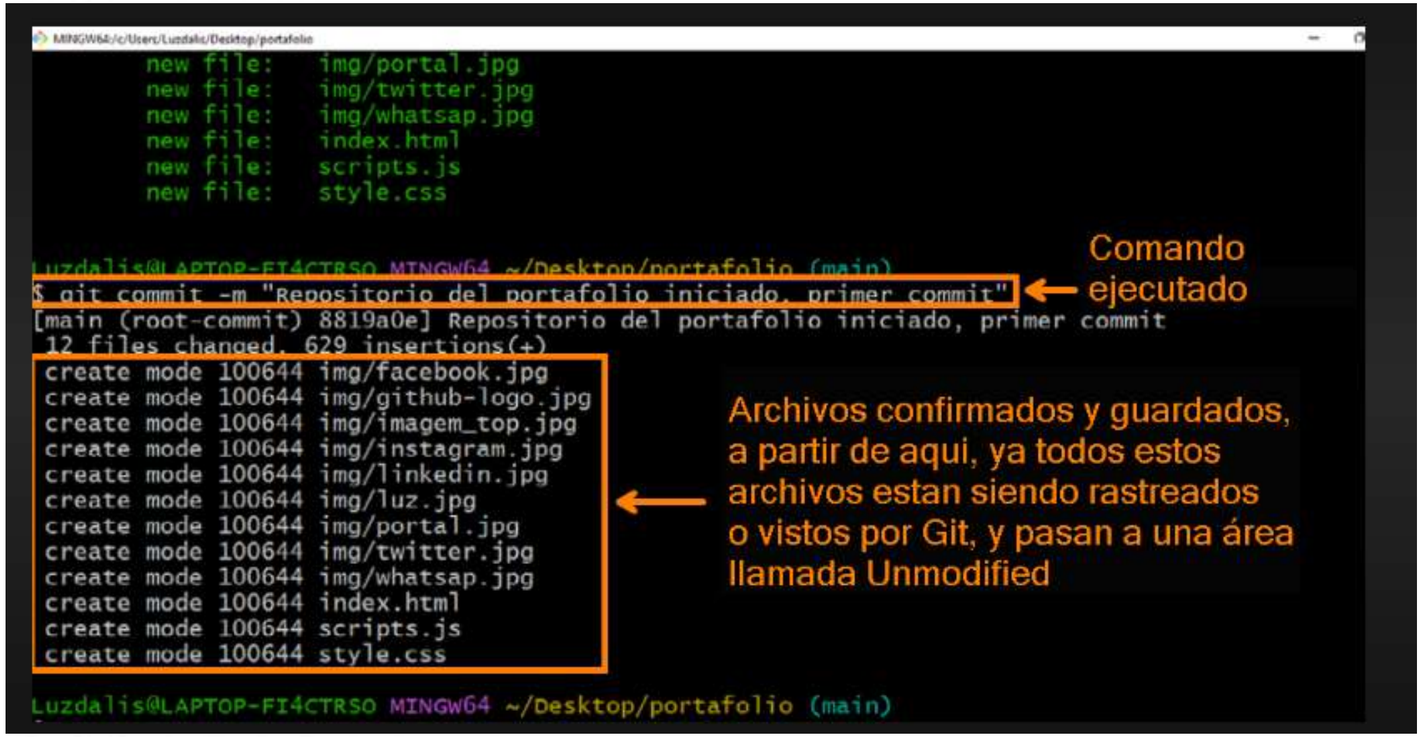
```
git status
```

Subir cambios a un repositorio remoto: Después de hacer cambios y commit, para subir esos cambios a un repositorio remoto:

```
git push origin master
```



Universidad Tecnológica del Perú



3 Ramas (Branching)

Manejo de Ramas (Branching)

Las ramas permiten trabajar en diferentes partes de un proyecto sin interferir con el código principal. Son esenciales para trabajar en nuevas características, corrección de errores, o experimentos.

Crear una nueva rama:

git branch nombre-rama

Esto crea una nueva rama llamada nombre-rama.

Cambiar de rama:

git checkout nombre-rama

Cambia a la rama nombre-rama.

También puedes usar git switch para hacerlo:

git switch nombre-rama

…Manejo de Ramas (Branching)

Crear y cambiar a una nueva rama en un solo paso:

git checkout -b nombre-rama

O usando switch:

git switch -c nombre-rama

Ver las ramas disponibles:

git branch

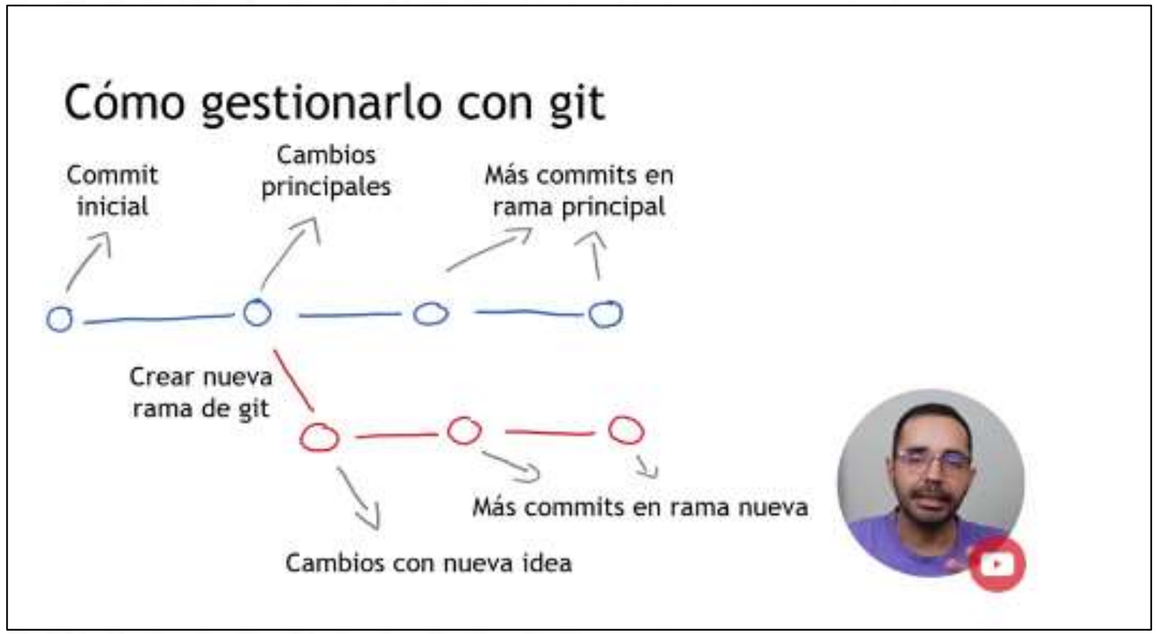
Muestra todas las ramas locales en el repositorio.

Fusionar ramas (Merge): Para fusionar una rama con la rama actual:

git merge nombre-rama

Eliminar una rama: Después de fusionar una rama, puedes eliminarla con:

git branch -d nombre-rama



https://youtu.be/QYIT6k4JOfM

4 Commits

Manejo de Commits

Los commits son instantáneas de los cambios en el proyecto y se utilizan para registrar la evolución del código.

Agregar cambios al área de preparación: Para preparar los archivos para commit:

```
git add archivo.txt
```

Si deseas agregar todos los archivos modificados:

```
git add .
```

Hacer un commit: Para guardar los cambios en el repositorio con un mensaje descriptivo:

```
git commit -m "Mensaje del commit"
```

…Manejo de Commits

Ver el historial de commits: Para revisar el historial de commits realizados en el repositorio:

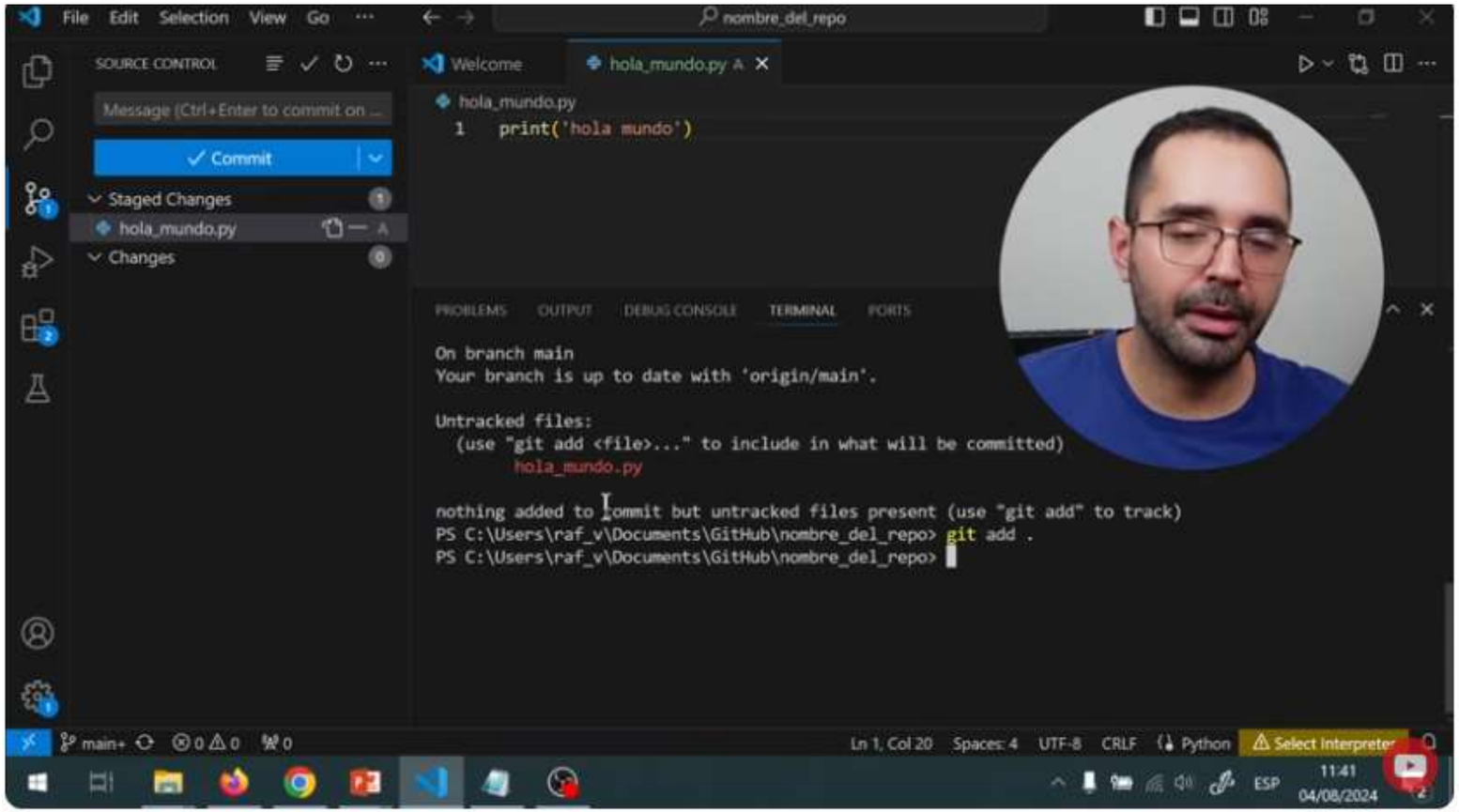
```
git log
```

Deshacer el último commit (sin perder los cambios): Si deseas deshacer el último commit pero mantener los cambios en tu directorio de trabajo:

```
git reset --soft HEAD~1
```

Deshacer un commit y perder los cambios: Para eliminar el último commit y también los cambios realizados:

```
git reset --hard HEAD~1
```



Cómo guardar cambios en GIT con git add y git commit

https://www.youtube.com/watch?v=ss0GnGnnrag 6:40

Práctica

Crea un nuevo repositorio en Git y haz tu primer commit.

Clona un repositorio remoto.

Crea una nueva rama llamada prueba-x y cámbiate a ella.

Haz cambios en un archivo, agrégalo y haz un commit.

Cierre

1. ¿Qué es un repositorio en Git?
2. ¿Qué es una rama (branch) en Git y para qué se utiliza?
3. ¿Qué hace el comando git status?
4. ¿Qué es un commit en Git?

Imagen obtenida de:
https://dpemfoco.com.br/wp-content/uploads/2019/08/Departamento-de-Pessoal-em-Foco-Checklist-Check-list-Checklists.png

Bibliografía

Hernández Bejarno, Miguel. *Ciclo de vida de desarrollo ágil de software seguro.* Fundación Universitaria Los Libertadores. https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl? biblionumber=36016

Guillamón Morales, Alicia. (). *Manual desarrollo de elementos software para gestión de sistemas.*
Editorial CEP, S.L. https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=34982

Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress.
Enlace: https://git-scm.com/book/es/v2

Poulton, N. (2017). *Docker Deep Dive*. Independently published.
Enlace: https://www.nigelpoulton.com/dvd/

Universidad
Tecnológica
del Perú