Práctica de Git y GitHub

# Parte 1: Configuración Inicial

**‘Configurar Git en tu equipo**

git config --global user.name "Tu Nombre"

git config --global user.email [tuemail@example.com](mailto:tuemail@example.com)

En este primer paso vamos a abrir nuestra terminal y añadimos los siguientes comandos para configurar Git con nuestro nombre:

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

# Crear un nuevo repositorio en GitHub

* + Accede a [GitHub](https://github.com/).
  + Haz clic en "New Repository".
  + Asigna un nombre al repositorio y selecciona la opción "Initialize this repository with a README".

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

1. **Clonar el repositorio en tu equipo** git clone URL\_DEL\_REPOSITORIO cd NOMBRE\_DEL\_REPOSITORIO

Captura de pantalla de un videojuego

El contenido generado por IA puede ser incorrecto.

# Parte 2: Trabajo Individual con Git

1. **Crear un archivo de prueba y hacer un commit**

echo "# Proyecto Web" > index.html git add index.html

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

git commit -m "Añadir archivo index.html"

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

# Subir cambios al repositorio remoto

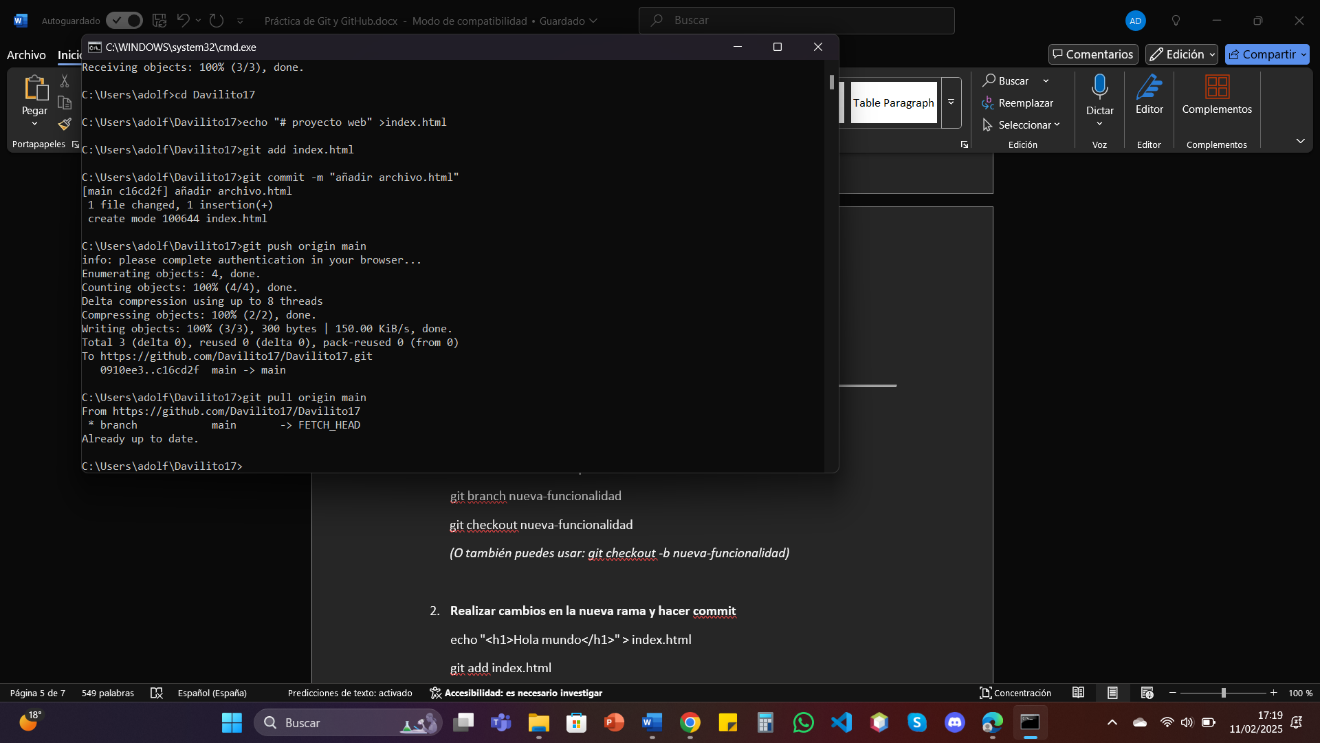
git push origin main

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

# Actualizar el repositorio local con los cambios remotos

git pull origin main



# Parte 3: Trabajo Colaborativo

1. **Crear una nueva rama para una funcionalidad**

git branch nueva-funcionalidad

git checkout nueva-funcionalidad

*(O también puedes usar: git checkout -b nueva-funcionalidad)*

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

# Realizar cambios en la nueva rama y hacer commit

echo "<h1>Hola mundo</h1>" > index.html git add index.html

git commit -m "Añadir título en index.html"

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

# Subir la rama al repositorio remoto

git push origin nueva-funcionalidad

Texto

El contenido generado por IA puede ser incorrecto.

# Realizar un Pull Request (PR) en GitHub

Un Pull Request (PR) es una solicitud para fusionar los cambios de una rama en otra, generalmente de una rama de desarrollo a main. Es una forma de revisar y discutir cambios antes de fusionarlos.

# Pasos para crear un PR:

* + Accede al repositorio en GitHub.
  + Ir a la pestaña "Pull Requests" y hacer clic en "New Pull Request".
  + Seleccionar la rama que contiene los cambios y compararla con main.
  + Revisar los cambios antes de enviarlos.
  + Escribir una descripción detallada del PR explicando los cambios realizados.
  + Hacer clic en "Create Pull Request".
  + Esperar revisiones y feedback de los compañeros o del profesor.

# Revisar y fusionar cambios

* + Otro compañero revisa el PR y lo aprueba.
  + Si todo está correcto, se fusiona con la rama main.
  + Si hay sugerencias o cambios requeridos, deben realizarse antes de la aprobación.
  + Una vez aprobado, hacer clic en "Merge Pull Request" y eliminar la rama si ya no es necesaria.

# Actualizar la rama local con los cambios aprobados

git checkout main git pull origin main

# Parte 4: Resolución de Conflictos

1. **Generar un conflicto (de forma intencionada)**
   * Dos alumnos editan la misma línea de index.html en diferentes ramas y hacen commits.
   * Ambos intentan fusionar sus ramas con main.

# Resolver el conflicto

* + Git avisará del conflicto.
  + Edita manualmente el archivo afectado, manteniendo los cambios correctos.
  + Una vez resuelto, hacer:

git add index.html

git commit -m "Resolver conflicto en index.html"

git push origin main

# Parte 5: Entrega Final

1. Cada equipo debe tener un repositorio con al menos:
   * Un README.md documentado.
   * Un historial de commits bien estructurado.
   * Uso de ramas y Pull Requests.
   * Al menos un conflicto resuelto correctamente.

# Enviar el enlace del repositorio a la plataforma de entrega.

**Evaluación**

* Creación y clonación del repositorio: **20%**
* Uso correcto de commits y ramas: **30%**
* Trabajo colaborativo y Pull Requests: **30%**
* Resolución de conflictos y estructura del repo: **20%**