

Entornos de Desarrollo - Prueba Técnica 1: Git y GitHub básico

Ejercicio 0: Configuración inicial

Este paso ya debería estar hecho de los ejercicios realizados en clase, pero si no lo has hecho aún: **configura** tu *nombre* y *correo electrónico* en git para que los commits, push al repositorio online, etc. tengan tus datos. Si no estás segur@, puedes ejecutar `git config --global --list` para ver en la terminal toda tu configuración global.

Nota: a lo largo de la prueba se pedirá que añadas al commit los comandos utilizados para resolver el ejercicio. A continuación, se presenta un ejemplo de estructura recomendada para que el mensaje de commit quede estructurado en varias líneas:

```
XXX XXXXXX -m Ej1: "descripción del cambio" -m "línea utilizada en bash" -m "otra línea" -m "otra línea"
```

Si por casualidad, al hacer un commit escribes mal el mensaje o lo envías antes de tiempo, puedes corregirlo fácilmente sin necesidad de abrir el editor de texto usando el siguiente comando:

```
git commit --amend -m "Nuevo mensaje correcto del commit"
```

Ejercicio 1: Crear el repositorio local

Usando la terminal (Bash), clona vía HTTPS el repositorio generado al entrar al enlace que te va a facilitar el profesor en [GitHub Classroom](#). A continuación, crea una carpeta llamada **project** dentro del nuevo repositorio, y dentro de esta, tres archivos vacíos llamados AboutUs.md, Temp.md y Title.md. Estos archivos contendrán texto muy corto para facilitar practicar con **git**. Cuando los hayas creado, agrega todos los archivos al área de preparación (staging area) y realiza el commit.

Ejercicio 2: Eliminar archivos

Resulta que Temp.md no pertenece a este proyecto y lo hemos agregado por error. Elimínalo del repositorio **usando git** (no solo borrándolo manualmente desde el explorador). Cuando lo hayas hecho, realiza un nuevo commit confirmando los cambios. A continuación, **copia y pega las líneas que has empleado en la terminal Bash** para realizar este ejercicio.

Ejercicio 3: Modificar contenido

Crea un encabezado de primer nivel para Title.md con el texto *Entornos de Desarrollo*. En cuanto al archivo AboutUs.md, crea otro encabezado de primer nivel con el texto *Sobre Nosotros* y, posteriormente, escribe una única línea de texto que contenga la siguiente descripción del módulo:

```
└─ En el módulo de Entornos de Desarrollo trabajamos con herramientas de control de versiones.
```

Cuando termines de editarlo, guarda el archivo, agrega los cambios al staging area y realiza un nuevo commit.

Ejercicio 4: Crear ramas

Crea una nueva rama llamada testing y cámbiate a ella. Ahora que estás en la rama testing, añade una pequeña ampliación al archivo AboutUs.md. Deberás escribir una nueva línea de texto que diga "Estoy haciendo mi primera prueba técnica de Entornos de Desarrollo usando Git".

Cuando termines, agrega los cambios al staging area y realiza un commit.

Después, vuelve a la rama principal (master o main). Crea otra rama llamada design y cámbiate a ella. En esta rama, modifica el archivo Title.md para que su contenido sea:

Crea también un nuevo archivo llamado `Style.md` y escribe en él una sola línea con el texto:

■ Color: azul;

Agrega los cambios al staging area y realiza un `commit` **añadiendo de nuevo todas las líneas usadas en Bash para resolver el ejercicio.**

Ejercicio 5: Situación de conflicto

Cámbiate de nuevo a la rama principal (`master` o `main`). Edita el archivo `Title.md` y cambia el encabezado por:

■ Entornos de Desarrollo - Principal

Guarda el archivo, agrega los cambios al staging area y realiza un `commit`.

Este cambio servirá para provocar un pequeño conflicto más adelante cuando unamos las ramas.

Ejercicio 6: Fusión y aparición de conflictos

Desde la rama principal, intenta incorporar los cambios de la rama `design` usando el comando de fusión `merge`. Al hacerlo, es posible que aparezca un conflicto en el archivo `Title.md`, ya que ambas ramas han modificado la misma línea de texto.

Observa en el archivo las marcas de conflicto (`<<<<<<, =====, >>>>>>`). En este caso, deberás conservar la versión que proviene de la rama `design`, ya que será la base para continuar con el siguiente ejercicio.

Ejercicio 7: Resolución de conflictos

Soluciona el conflicto manualmente usando un editor de texto sencillo (como el Bloc de notas o `nano`). Recuerda que en el ejercicio anterior decidiste conservar la versión proveniente de la rama `design`.

Ahora, a partir de esa versión, modifica el encabezado de `Title.md` para dejarlo con un texto más definitivo, escribiendo:

■ Entornos de Desarrollo - Versión 1.0

Una vez resuelto el conflicto y actualizado el archivo, elimina las marcas del `merge` (si aún quedaran), guarda los cambios, agrégalos al staging area y realiza un `commit` de `merge` para confirmar que el conflicto ha sido resuelto.

Con este cambio dejarás el archivo libre de conflictos y con el título definitivo del proyecto.

Ejercicio 8: Más cambios

Cámbiate a la rama `design` e introduce un cambio menor en el archivo `Style.md`. Por ejemplo, cambia el color a otro cualquiera (puede ser "verde", "rojo", "gris", etc.). Guarda, agrega los cambios y realiza un `commit`.

Ejercicio 9: Fusión final

Regresa a la rama principal (`master` o `main`) e incorpora de nuevo los cambios realizados en la rama `design`. Si aparece algún conflicto, resuélvelo manualmente priorizando la modificación realizada en la rama `design`. Cuando todo esté correcto, realiza un nuevo `commit` final indicando que es la versión final del proyecto ("Ej9: Versión final").

En este punto, incluir en el mensaje del `commit` las líneas de comandos utilizadas en Bash para resolver el ejercicio 8 y 9.

Ejercicio 10: Repositorio remoto

Recuerda que para esta prueba se te ha asignado un repositorio remoto en GitHub Classroom. Empuja (`push`) todas las ramas creadas para que estén disponible en el remoto.