



**Certified Tech
Developer**
The Ultimate Degree

Git + GitHub - Extra

Práctica integradora

Objetivos

- Vamos a crear un repositorio remoto en GitHub, un repositorio local en nuestra computadora y luego vamos a trabajar con archivos locales, sincronizando los cambios en GitHub.
- ¡Atención! Esta ejercitación se realiza de a pares, así que ¡es hora de buscar a tu dupla!
- **Utilizarán la CLI de Git bash.**

Requisitos

Una cuenta de GitHub: para crearla podemos acceder a <https://github.com>. En caso de que ya tengamos una cuenta, simplemente ingresamos con nuestro usuario.

Tener Git instalado en la computadora: para esto, deberemos seguir la guía de instalación que se encuentra disponible en el campus o en <https://git-scm.com/downloads>.



Desafío 1

Durante las consignas nos referiremos a los participantes como **A** y **B**.

1. Crear un repositorio remoto

El **participante A** deberá crear un nuevo repositorio en <https://github.com>.

Es importante que el repositorio se cree vacío **tildando la opción del README.md**.

2. Invitar a los colaboradores

El **participante A** deberá invitar al **participante B** como colaborador del repositorio que creó anteriormente.

¡Atención! Si el **participante B** acaba de crear su cuenta de GitHub, es posible que no figure en el formulario de búsqueda, deberán esperar unos minutos para poder hacerlo.

3. **Ambos participantes** deberán **clonar** el repositorio remoto.

El **participante A** deberá modificar el archivo Readme.md y escribir en el mismo los tres estados de git y al finalizar cada palabra le agrega un emoji al lado. ejemplo primerEstado 🤖. Acá les dejo un link donde les mostraré que deben usar para que aparezca el emoji

<https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities>

4. Una vez que el **participante A** haya subido el primer archivo, el **participante B** deberá actualizar su repositorio local para que se vea reflejado el cambio hecho por el compañero. Ahora el **participante B** deberá modificar el archivo Readme.md y pondrá los nombres de los estados de git en **negrita**, **cursiva** y **en forma de lista** con la viñeta (Punto). Les dejo el link en donde buscar como hacerlo

<https://docs.github.com/es/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax#uploading-assets> ejemplo

- **primerEstado** 😊

5. Trabajar en archivos diferentes (parte 1)

Ambos participantes deberán crear 1 archivo cada uno (Un index.html y un style.css - acá deben decidir entre ustedes quién hace qué archivo así que **¡a ponerse de acuerdo! -)**

Una vez creados, deberán agregarles contenido, agregarlos al repositorio local y sincronizarlos con el repositorio remoto. Terminado esto, cada participante deberá descargar los cambios realizados por su pareja.

Al finalizar este punto, ambos deberían tener 3 archivos: el README.md y los 2 archivos creados.

6. Trabajar en archivos diferentes (parte 2)

Ambos participantes agregarán contenido a cualquiera de los archivos que hayan creado. A continuación, subirán los cambios al repositorio remoto.

Luego, ambos participantes actualizarán su repositorio local y verificarán que los cambios del otro participante se hayan aplicado.

7. Trabajar en el mismo archivo

Ambos participantes deberán seleccionar el mismo archivo Index.html y hacer modificaciones en su respectiva computadora (Los dos participantes agregaran en la **línea 10** lo siguiente **<p>Bienvenidos a Introducción a la Informática</p>**). y si desean agregar algo más lo pueden hacer. Una vez modificado el archivo en cada repositorio local, ambos deben intentar sincronizarlo con el repositorio remoto.

8. Resolver un conflicto

El primer participante que sincronice el archivo podrá hacerlo sin problema. El segundo recibirá una notificación de que existen cambios en el repositorio remoto.

¡Ahora deben decidir de qué manera resolverán el conflicto!

- Dejando solo el contenido del participante A.

- Dejando solo el contenido del participante B.
- Unificando el contenido de ambos participantes

Una vez decidido el camino, el participante que corresponda hará los cambios necesarios y subirá el archivo actualizado al repositorio remoto.

Para terminar el otro participante actualizará su repositorio local y verificará que los cambios hayan sido aplicados.

9. ¡Ejercitación libre!

Ahora les toca a ustedes: creen nuevos archivos, modifiquen los existentes, generen conflictos y resuélvanlos.

10. **EXTRA:** Crear dos branch diferentes del main, y subir archivos a las mismas

Recuerda:

- a. El comando para crear branch es **git branch nombre_rama**
- b. Para posicionarse en esa rama el comando es **git checkout nombre_rama**
- c. Para subir cambios a la rama creada es **git push origin nombre_rama**



Cierre

Git y GitHub son herramientas indispensables para cualquier desarrollador sin importar el ámbito de trabajo.

También son herramientas que vamos a estar utilizando todo el tiempo, así que recomendamos practicar con ellas hasta dominarlas porque van a hacer nuestro trabajo mucho más eficiente.

¡Hasta la próxima!