

## Git + GitHub

### Práctica integradora

Vamos a validar lo aprendido. Te proponemos el siguiente desafío para que lo intenten resolver de manera grupal, repasemos y validemos cómo lo resolvieron.

¡Empecemos! 🕶️👍✨

## 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!
- Podemos utilizar la terminal o alguna interfaz gráfica para la resolución de las consignas. Sin embargo, recomendamos, por lo menos al principio, usar la terminal. Después veremos otras opciones más gráficas.

## Requisitos

- Una cuenta de **GitHub**: para crearla podemos acceder a [github.com](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 la sección de Git.

## Consignas

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 [github.com](https://github.com).

Es importante que el repositorio se cree vacío (**sin tildar 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. Clonar el repositorio remoto y trabajar sobre él.**

El **participante A** deberá clonar el **repositorio remoto** que creó anteriormente y subir un **archivo README.md** con el **título del repositorio**. Para eso, es recomendable seguir las **instrucciones** que **figuran** en **GitHub** al crear un repositorio.

**4. Clonar un repositorio remoto**

Una vez que el **participante A** haya **subido** el **primer archivo**, el **participante B** deberá **clonar** el **repositorio remoto** creado por el **participante A**.

**5. Trabajar en archivos diferentes (parte 1)**

**Ambos participantes** deberán **crear 3 archivos** cada uno. Los **archivos deben tener** todos **nombres diferentes** (ej. pikachu.txt), 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 7 archivos**: el README.md y los 3 archivos creados por cada uno.

----- **Checkpoint** -----

**6. Trabajar en archivos diferentes (parte 2)**

Para el siguiente paso es necesario descargar el archivo index.html de la parte de Actividades en el sitio web.

**Ambos participantes agregarán** contenido al **archivo index.html** que han descargado. 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 **archivo index.html** y **hacer modificaciones** en su respectiva computadora. Les **sugerimos** escribir un par de **líneas de texto** que **tengan sentido** porque va hacer más sencillo el **próximo paso**. 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:**

- El **comando** para **crear branch** es **git branch nombre\_rama**
- Para **posicionarse** en esa **rama** el comando es **git checkout nombre\_rama**
- 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!