

TRABAJO PRÁCTICO N° 1

1. Ejercicio 1 - Versionado en git y github.

Consulte la guía “cómo vincular un repositorio remoto para crear y vincular su repositorio de github.

Si aún no tiene instalado Git en su PC, proceda a instalar y configurarlo siguiendo la esta guía de [Instalación y configuración de Git](#).

Crear una cuenta de usuario en GitHub (<https://github.com/>) si todavía no posee una (puede seguir la siguiente guía [Crear cuenta en GitHub](#)).

Paso a paso para vincular un repositorio remoto

- Crear en github un repositorio con el nombre tp1 (puede seguir la siguiente guía [Crear un repositorio en GitHub](#))
- Crear una carpeta nueva en el disco para su repositorio llamado tp1.
IMPORTANTE: Controle que ninguna carpeta que pertenezca a la ruta tenga espacios.
- Utilizando el git bash vaya a la carpeta tp1 creada en el punto anterior.
- Ejecute los siguientes comandos:

- a. `echo "# tp1" >> README.md`
- b. `git init`
- c. `git status`
- d. `git add README.md`
- e. `git status`
- f. `git commit -m "primer commit"`
- g. `git branch -M main`
- h. `git remote add origin https://github.com/{su_usuario}/tp1.git`
- i. `git push -u origin main`
- j. Cree un `archivo.txt` que contenga su nombre y suba los cambios al repositorio remoto.

Nota: En el siguiente video https://www.youtube.com/watch?v=poBFum9_ALw encontrará un breve tutorial sobre comandos básicos de git.

2. Ejercicio 2 - Repaso de punteros

Si aún no tiene instalado VS Code o configurado el compilador / debugger de C en su PC, consulte la guía De Instalación y [Configuración de VS Code con C](#).

1. Clone el repositorio ubicado en <https://classroom.github.com/a/kl-E8VQf> (hacer el paso a paso que informa github)
2. a) Investigar el archivo `.gitignore` ¿Por qué es conveniente incluirlo? ¿Cuándo se debe hacer? ¿Cómo configuraría el archivo `.gitignore`?

-Incluya en su repositorio una regla de cómo ignorar un archivo *ignorado.txt*

b) Utilice el archivo `readme.md` para escribir las respuestas.
Utilice como guía los siguientes recursos:
Formato: <https://cutt.ly/jtvP2ER>
Orden: <https://cutt.ly/StvUFis>
3. Genere en el repositorio un archivo que se llame `tp1_1.cpp`.
 - a) Utilizando VS Code, abra el archivo creado y cree el método `main()` para poder compilar el archivo que imprima “hola mundo” por pantalla.
 - b) Repita los **c a e** del paso a paso del Ejercicio 1 para realizar el *commit* inicial del archivo y subirlo a la plataforma
 - c) Revise los archivos subidos en su repositorio de github. ¿Qué archivos vé? ¿Cuál cree que no hace falta que esté?
 - d) Implemente el archivo `.gitignore` correspondiente usando la página <https://www.gitignore.io/> para no incluir los archivos que considere innecesarios
 - e) Declare un puntero en el procedimiento principal que apunte a una variable de algún tipo y muestre por pantalla lo siguiente:
 - 1) El contenido del puntero
 - 2) La dirección de memoria almacenada por el puntero.
 - 3) la dirección de memoria de la variable.
 - 4) la dirección de memoria del puntero.
 - 5) el tamaño de memoria utilizado por esa variable usando la función `sizeof()`
 - f) Hacer el segundo commit del archivo y subirlo a la plataforma
 - g) Si resolvió correctamente los apartados 2 y 3 del punto anterior notará que el resultado es el mismo. ¿a qué se debe? ¿Qué obtiene en el punto 4? ¿Es igual a los anteriores? ¿Por qué? Responder en el archivo *readme.md*
4. Genere en el repositorio un archivo que se llame `tp1_2.cpp` y resuelva las siguientes tareas mediante el uso de funciones:

- a) Haga una función que devuelva el cuadrado de un número
- b) Haga la función anterior, pero devolviendo un tipo void
- c) Al recibir una variable muestre por pantalla la dirección y el contenido de la variable
- d) Dado dos parámetros de entrada, deberá invertir los valores entre ambos.

```
void Invertir(a,b) //deberá devolver en el valor de a en la  
variable b, y en el valor de b el valor de a
```

- e) Dado dos parámetros de entrada, deberá devolverlos de forma ordenada, en el primer parámetro el menor y en el segundo el mayor.

```
void orden(a,b) //deberá devolver en el valor de a el valor más  
chico, y en el valor de b el valor más grande
```

- f) Utilice las funciones anteriores para leer pares de valores e imprimirlos por pantalla.
- g) Al finalizar, debe subir todos los cambios al repositorio, usando los siguientes comandos.

```
i. git add .  
  
ii. git commit -m "ejercicio 2.4 completado"  
  
iii. git push
```