

Guía de ejercicios 1



¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

¿En qué consiste esta guía?

La siguiente guía de estudio tiene como objetivo practicar y ejercitar los contenidos que hemos visto en clase.

¡Vamos con todo!



Tabla de contenidos

Guía de ejercicios 1	1
¿En qué consiste esta guía?	1
Tabla de contenidos	1
Actividad guiada N° 1: Manejando archivos y carpetas	2
¡Manos a la obra! - Estructura de carpetas	3
Actividad guiada N° 2: Utilizando git en un proyecto	4
¡Manos a la obra! - Pasos para iniciar git	8



¡Comencemos!



Actividad guiada N° 1: Manejando archivos y carpetas

Como vimos en las páginas anteriores, existen múltiples comandos que nos ayudarán a navegar a través de los archivos y directorios, facilitando nuestro trabajo. Para familiarizarnos con ellos, haremos un recorrido por los comandos que acabamos de revisar:

- **Paso 1:** crea un nuevo directorio con un comando. Para ello, dirígete a la carpeta raíz de tu computador. Si no estás en esta carpeta, ve a ella escribiendo `cd`.

Ahora escribe en la consola:

```
mkdir proyecto1
```

Esto creará un nuevo directorio llamado `proyecto1`.

- **Paso 2:** crea un archivo desde la terminal con el comando `touch`, con el nombre `index.html`:

```
touch index.html
```

Esto crea el archivo `index.html` en la ubicación actual, o sea la carpeta raíz del computador. Si utilizamos `ls` podremos ver el archivo creado.

- **Paso 3:** ahora que ya sabes crear un directorio y un archivo, copia el archivo `index.html` dentro del directorio que habías creado, o sea en la carpeta `proyecto1`, de la siguiente manera:

```
cp index.html /proyecto1/index.html
```

- **Paso 4:** utilizando este mismo comando también puedes cambiarle el nombre a los archivos copiados. Vuelve al directorio anterior con `cd ..`, luego ocuparás `cp` nuevamente pero con un nuevo nombre de archivo.

```
cp index.html /proyecto1/index2.html
```

Si ingresas a `proyecto1` y listamos los archivos con `ls`, debería aparecer el archivo copiado con el nuevo nombre.

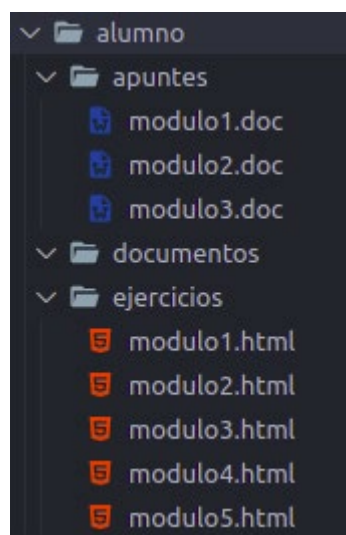
- **Paso 5:** si quieres copiar un archivo dentro de la misma carpeta donde se encuentra, solo debes escribir el nombre del archivo seguido del nombre que le quieres poner.

```
cp index.html index3.html
```

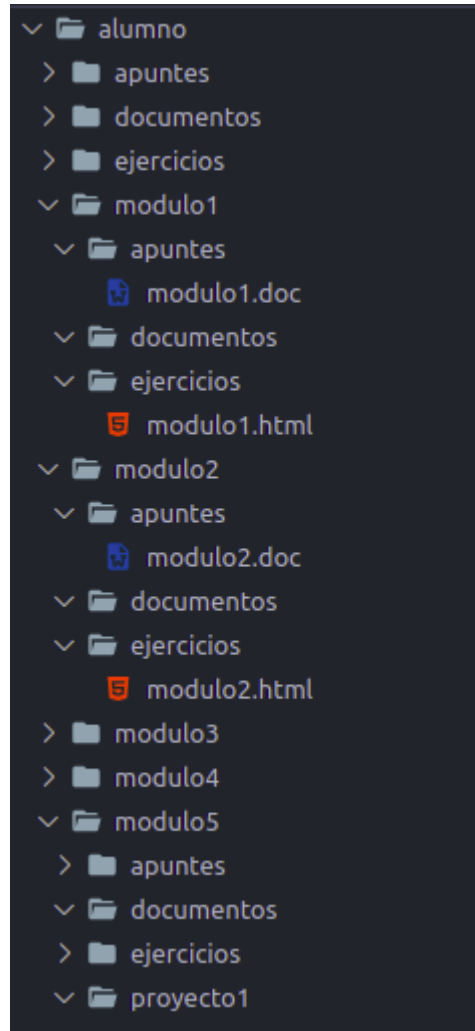


¡Manos a la obra! - Estructura de carpetas

Se pide crear la estructura de carpetas y archivos por medio de consola



Luego, se pide crear una carpeta por cada módulo, manteniendo la estructura de subcarpetas, de la siguiente forma:



1. Se pide eliminar las carpetas obsoletas:

- alumno/apuntes.
- alumno/documentos.
- alumnos/ejercicios.



Actividad guiada N° 2: Utilizando git en un proyecto

Vamos a utilizar git dentro de un proyecto. En nuestro caso, vamos a iniciar git en el sitio creado con HTML y CSS de un proyecto anterior llamado "meet&coffee".

Sigue atentamente los pasos que te presentamos a continuación:

Primero, ubica dónde tienes el proyecto. Para este ejercicio, recuerda que probablemente tienes el proyecto en el escritorio. Abre la consola git bash o directamente en el Visual Studio Code y asegúrate de estar posicionado en la carpeta del proyecto. ¿Listo? Empecemos.

- **Paso 1:** inicializamos git dentro de la carpeta con:

```
git init
```

Observaremos un mensaje indicando que se inició git.

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee
$ git init
Initialized empty Git repository in C:/Users/Ale/Desktop/meet&coffee/.git/
```

Imagen 1. Respuesta `git init`.

Con esta acción hemos determinado que esta carpeta será nuestro working directory, el lugar donde se almacenarán nuestros cambios. Si utilizamos el comando `ls -a` veremos qué se creó la carpeta `.git`.

- **Paso 2:** recordemos cómo agregar los cambios. Para ello se ocupa git add seguido de todos los archivos que queremos agregar.

Con el comando:

```
git status
```

Veremos un mensaje del tipo:

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        assets/
        favicon.png
        index.html

nothing added to commit but untracked files present (use "git add" to track)
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (master)
$
```

Imagen 2. Respuesta `git status`.

En este punto git nos está diciendo que **No** hemos hecho ninguna confirmación y que hay archivos en nuestro directorio de los cuales no está haciendo seguimiento (regularmente se le conoce como *Tracking*), es desde aquí donde entra la metáfora de la caja.

- **Paso 3:** vamos a empezar por agregar el archivo **index.html**. Utilizaremos el comando:

```
git add index.html
```

Después de hacerlo no obtendremos ninguna información, pero si queremos revisar que sucedió podemos utilizar de nuevo `git status` y veremos:

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        assets/
        favicon.png

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (master)
```

Imagen 3. Respuesta 2 `git status`.

Como podemos observar, git nos indica que hemos añadido un archivo, pero que aun tenemos otros que no están agregados.

- **Paso 4:** para añadir el resto de archivos, vamos a utilizar el comando:

```
git add .
```

Incluirá todos los archivos que no han sido añadidos aún.

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   assets/Icon_
    new file:   assets/css/Icon_
    new file:   assets/css/style.css
    new file:   assets/img/Icon_
    new file:   assets/img/bg-hero.png
    new file:   assets/img/coffee-cup.svg
    new file:   assets/img/machine-learning.jpg
    new file:   assets/img/post-1.jpg
    new file:   assets/img/scrumb-sin-scream.jpg
    new file:   assets/img/simposio-vegan.jpg
    new file:   assets/img/speaker.jpg
    new file:   assets/img/we-work.jpg
    new file:   favicon.png
    new file:   index.html

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (master)
```

Imagen 4. Uso git status.

- **Paso 5:** lo confirmaremos utilizando `git status`. Y observamos que no queda ningún archivo por añadir.

El mensaje nos dice que los archivos ya están agregados y que nos falta hacer el commit, es decir, la confirmación. Esto será equivalente a cerrar la caja y ponerle una etiqueta con descripción de los cambios que hicimos.

- **Paso 6:** para hacer la confirmación escribiremos:

```
git commit -m "First Commit meet&Coffee"
```

La opción `-m` nos permite escribir ese mensaje en la misma línea donde confirmamos los cambios.

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (master)
$ git commit -m "First Commit meet&Coffee"
[master (root-commit) 39b195c] First Commit meet&Coffee
14 files changed, 227 insertions(+)
create mode 100644 assets/Icon_
create mode 100644 assets/css/Icon_
create mode 100644 assets/css/style.css
create mode 100644 assets/img/Icon_
create mode 100644 assets/img/bg-hero.png
create mode 100644 assets/img/coffee-cup.svg
create mode 100644 assets/img/machine-learning.jpg
create mode 100644 assets/img/post-1.jpg
create mode 100644 assets/img/scrum-sin-scream.jpg
create mode 100644 assets/img/simposio-vegan.jpg
create mode 100644 assets/img/speaker.jpg
create mode 100644 assets/img/we-work.jpg
create mode 100644 favicon.png
create mode 100644 index.html

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (master)
$ |
```

Imagen 5. Uso de `commit -m`.

¡Listo! Lo logramos, hemos guardado nuestra primera versión.

- **Paso 7:** para asegurarnos, utilizaremos el comando `git log`

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (master)
$ git log
commit 39b195c6d9da708b7ca23e8337145abcb6218a8d (HEAD -> master)
Author: alegonzalezcelis <alegonzalez1993@gmail.com>
Date:   Wed Oct 14 12:16:35 2020 -0300

    First Commit meet&Coffee

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (master)
$ |
```

Imagen 6. Respuesta a `git log`.

Esto nos indicará cuál fue el commit realizado.

La secuencia de letras y números que vemos al comienzo es el **hash**, también se le conoce como **checksum**. Es un identificador único de cada confirmación y sirve para comparar códigos entre distintas versiones, entre otras cosas.

Además aparece el autor de cada confirmación, la fecha cuando fue realizada y el texto de la confirmación. Esto será muy útil para realizar la gestión de cambios en un proyecto donde hayan múltiples personas trabajando.



¡Manos a la obra! - Pasos para iniciar git

Toma un proyecto anterior o incluso la estructura de carpetas que hemos creado en el ejercicio 1 y realiza los pasos para iniciar git y versionar los cambios de tu proyecto.

1. Inicializa git en la carpeta contenedora.
2. Agrega los archivos al stage.
3. Genera el primer commit.
4. Muestra el status.