



Examen 2 y 3

Fecha de la actividad: 22/03/2023

Nombre del alumno: Julian Mario Retamar Natale

Nombre de la asignatura: Entornos de desarrollo

Curso: Desarrollo de aplicaciones multiplataforma

Año: 1º

Correo: julianrnatale@gmail.com

Turno: Tarde

Apartado 1

1. Descarga el siguiente proyecto: xxxxxx, y activa el control de versiones de forma local en la carpeta correspondiente.

```
MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git init
Initialized empty Git repository in C:/Users/MANU/Desktop/EXAMEN/Examen2/.git/

MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git remote add origin https://github.com/Akwarck/masteruah.git

MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git clone https://github.com/adwinmbd/astro-blog-template
Cloning into 'astro-blog-template'...
remote: Enumerating objects: 263, done.
remote: Counting objects: 100% (117/117), done.
remote: Compressing objects: 100% (76/76), done.
remote: Total 263 (delta 42), reused 91 (delta 28), pack-reused 146
Receiving objects: 100% (263/263), 570.66 KiB | 1.90 MiB/s, done.
Resolving deltas: 100% (95/95), done.
```

\$ git init

\$ git remote add origin https://github.com/Akwarck/EJ2_ex2_ED.git

2. Revisa el estado de todos los archivos del repo.

```
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    astro-blog-template/

nothing added to commit but untracked files present (use "git add" to track)
```

\$ git status

3. Añade todos los archivos del repo a seguimiento.

```

$ git add .
warning: adding embedded git repository: astro-blog-template
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> astro-blog-template
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached astro-blog-template
hint:
hint: See "git help submodule" for more information.

```

\$ git add .

4. Comprueba cuántas ramas tiene el proyecto y en la que te encuentras.

```

MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git branch
* master

```

\$ git branch

5. Realizar un commit de los últimos cambios con el mensaje “Primer commit del examen” y ver el estado del repositorio.

```

MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git commit -m "Primer commit del examen"
[master (root-commit) cf563ec] Primer commit del examen
1 file changed, 1 insertion(+)
create mode 160000 astro-blog-template

```

\$ git commit -m "Primer commit del examen"

\$ git status

6. Lista todos los commits que has realizado.

```

MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git log
commit cf563ec983a7e4e18262faf133069db0808982d3 (HEAD -> master)
Author: Julian <julian@example.com>
Date:   Wed Mar 22 18:17:32 2023 +0100

    Primer commit del examen

```

\$ git log

7. Realiza un cambio en alguno de los archivos del proyecto.

```
MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ touch readme.md
```

\$ touch readme.md

\$ notepad readme.md

8. Revisa el estado de los archivos del repo.

```
MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    readme.md

nothing added to commit but untracked files present (use "git add" to track)
```

\$ git status

9. Añade los cambios a seguimiento y realiza un commit (todo con un solo comando).

```
MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git commit -m "cambio a readme.md"
[master e38a171] cambio a readme.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 readme.md
```

\$ git add readme.md

\$ git commit -m "cambio a readme"

10. Realiza un segundo commit con tu nombre y apellidos.

```
MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git commit -m "Julian Mario Retamar Natale"
On branch master
nothing to commit, working tree clean
```

\$ git commit -m "Julian Mario Retamar Natale"

11. Lista todas las ramas del proyecto.

```
MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git branch
* master
```

\$ git branch

12. Crea dos ramas diferentes: dev1, dev2.

```
MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git branch dev1

MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git branch dev2
```

\$ git branch dev1

\$ git branch dev2

13. Muévete a dev1.

```
MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (master)
$ git checkout dev1
Switched to branch 'dev1'
```

\$ git checkout dev1

14. Muévete a dev2.

```
MANU@DESKTOP-J0TQD9R MINGW64 ~/Desktop/EXAMEN/Examen2 (dev1)
$ git checkout dev2
Switched to branch 'dev2'
```

\$ git checkout dev2

Muy bien: si has documentado todo correcto con pantallazos y los comandos utilizados, así como has indicado la url de tu repo: lo has conseguido

EXAMEN 3

3. Control de versiones(2)

- Sincroniza tu directorio del ejercicio anterior con un repositorio sin inicializar de GitHub llamado EJ3_ex_ED_2022.

```
$ git remote add origin https://github.com/Akwarck/EJ3_ex2_ED-2023.git
```

- Crear una nueva rama "refactorizacion" y mostrar las ramas del repositorio.

```
$ git branch refactorizacion
```

- Crear el fichero refactorizacion.txt y añadir el texto siguiente

```
$ touch refactorizacion.txt
```

```
$ notepad refactorizacion.txt
```

"Refactoring is a systematic process of improving code without creating new functionality that can transform a mess into clean code and simple design."

- Añadir los cambios a la zona de intercambio temporal.

```
$ git add refactorizacion.txt
```

- Hacer un commit con el mensaje "Añadido concepto de refactorizacion"

```
$ git commit -m "Añadido el concepto de refactorizacion"
```

- Mostrar la historia del repositorio incluyendo todas las ramas.

```
$ git log
```

```
$ git branch
```

- Crear la rama "patrones".

```
$ git branch patrones
```

- Crear el fichero patrones.txt y añadir la siguiente referencia

"Los patrones de diseño (design patterns) son soluciones habituales a problemas comunes en el diseño de software."

\$ touch patrones.txt

\$ notepad patrones.txt

- Añadir los cambios a la zona de intercambio temporal.

\$ git add patrones.txt

- Hacer un commit con el mensaje "Añadida primera definición de patrones de diseño"

\$ git commit -m "Añadida primera definición de patrones de diseño"

- Mostrar la historia del repositorio incluyendo todas las ramas.

\$ git log

\$ git branch

- Fusionar la rama patrones con la rama master.

\$ git checkout master

\$ git merge patrones

- Mostrar la historia del repositorio incluyendo todas las ramas.

\$ git log

\$ git branch

- Eliminar la rama patrones.

\$ git branch -D

- Mostrar de nuevo la historia del repositorio incluyendo todas las ramas.

\$ git log

\$ git branch

- Crear la rama patrones.

\$ git branch patrones

- Cambiar a la rama patrones.

\$ git checkout patrones

- Cambiar el fichero patrones.txt para que contenga las siguientes referencias:

"Cada patrón es como un plano que se puede personalizar para resolver un problema de diseño particular de tu código."

\$ touch patrones.txt

- Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje

"Añadida definición 2 de patrones de diseño"

\$ git add patrones.txt

\$ git commit -m "Añadida definición 2 de patrones de diseño"

- Cambiar a la rama master.

\$ git checkout patrones

- Cambiar el fichero patrones.txt para que contenga las siguientes referencias:

"Los patrones son un juego de herramientas que brindan soluciones a problemas habituales en el diseño de software. Definen un lenguaje común que ayuda a tu equipo a comunicarse con más eficiencia."

```
$ touch patrones.txt
```

- Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "Añadida definición 3."

```
$ git add patrones.txt
```

```
$ git commit -m "Añadida definicion 3"
```

- Fusionar la rama patrones con la rama master (si existe algún conflicto, resuelvo como mejor creas).

```
$ git checkout master
```

```
$ git merge patrones
```

- Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "Resuelto conflicto de patrones."

```
$ git add .
```

```
$ git commit -m "Resuelto conflicto de patrones"
```

- Mostrar la historia del repositorio incluyendo todas las ramas.

```
$ git log
```

```
$ git branch
```