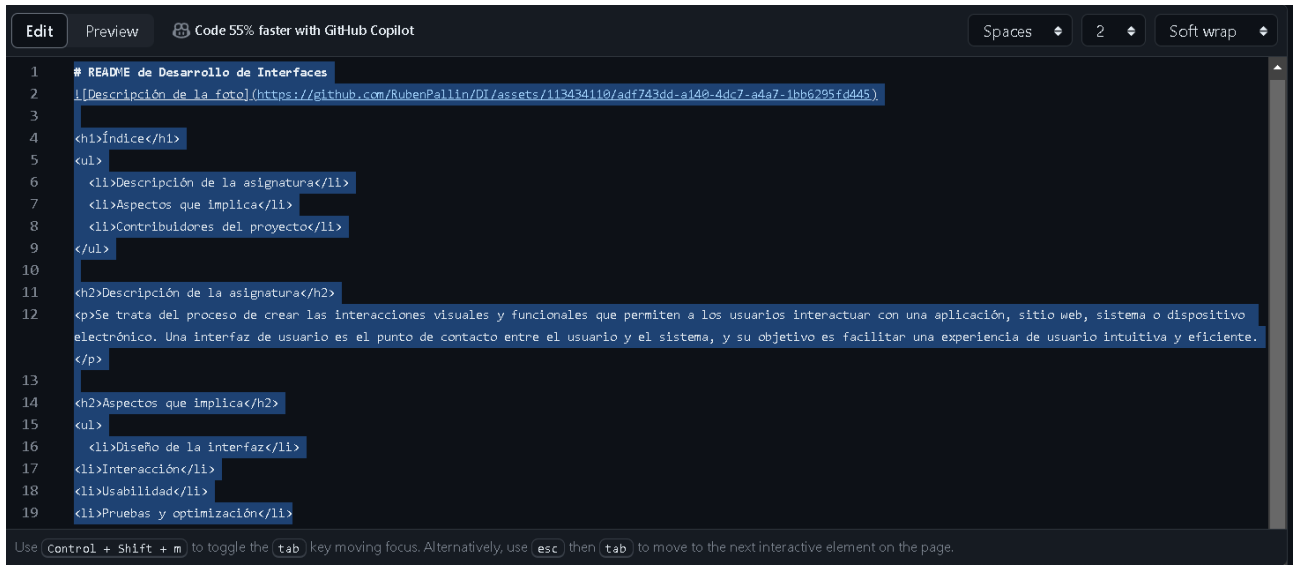


DOCUMENTACIÓN DE LOS EJERCICIOS

1. Clona tu repositorio de Desarrollo de interfaces, configura de manera global tu nombre y e-mail y efectúa un primer commit a main cambiando el archivo README.md (debería estar listo desde el sprint pasado). Una vez hecho esto, accede a GitHub y completa el README.md siguiendo esta guía.

Este fue el código que le agregué a mi README.md

A screenshot of a code editor window with a dark theme. The editor shows the content of a README.md file. The text is in Spanish and uses HTML tags for formatting. The editor has tabs at the top: 'Edit', 'Preview', and 'Code 55% faster with GitHub Copilot'. On the right side, there are settings for 'Spaces' (set to 2), 'Soft wrap', and a dropdown menu. The code content is as follows:

```
1 # README de Desarrollo de Interfaces
2 i[Descripción de la foto](https://github.com/RubenPallin/DI/assets/113434110/adf743dd-a148-4dc7-ada7-1bb6295fd445)
3
4 <h1>Índice</h1>
5 <ul>
6   <li>Descripción de la asignatura</li>
7   <li>Aspectos que implica</li>
8   <li>Contribuidores del proyecto</li>
9 </ul>
10
11 <h2>Descripción de la asignatura</h2>
12 <p>Se trata del proceso de crear las interacciones visuales y funcionales que permiten a los usuarios interactuar con una aplicación, sitio web, sistema o dispositivo electrónico. Una interfaz de usuario es el punto de contacto entre el usuario y el sistema, y su objetivo es facilitar una experiencia de usuario intuitiva y eficiente.
13 </p>
14 <h2>Aspectos que implica</h2>
15 <ul>
16   <li>Diseño de la interfaz</li>
17   <li>Interacción</li>
18   <li>Usabilidad</li>
19   <li>Pruebas y optimización</li>
```

Y así es como se muestra

2. Haz git log y examina el resultado. Añade el siguiente “alias” de Git empleando este comando:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces
$ git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<an>%Creset' --abbrev-commit"
```

Y hacemos el git lg. Esto es lo que nos muestra.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces
$ git lg
* 1f5a0ec - (HEAD -> main, origin/main, origin/HEAD) Initial commit (65 minutes ago) <RubenPallin>
```

3. Completar los ejercicios propuestos en la siguiente web, dado que es un tutorial con soluciones, crea un archivo de texto “Documentación” en el que vayas explicando qué es lo que haces en cada uno de los ejercicios de los siguientes apartados.

EJERCICIO 1

Configurar Git definiendo el nombre del usuario, el correo electrónico y activar el coloreado de la salida. Mostrar la configuración final.

Para configurar nuestro nombre y correo debemos emplear el git config --global seguido de user.name en caso del nombre y user.email en caso del email y entre comillas pones el que quieras.

Para el coloreado debemos hacer lo siguiente:

Si lo listamos aparecerá la información.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces
$ git config --global user.email "your-email-address"

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces
$ git config --global user.email "rpalling@fpccoruna.afundacion.org"

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces
$ git config --global color.ui auto

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces
$ git config --list
user.name=Ruben Pallin
user.email=rpalling@fpccoruna.afundacion.org
core.editor=nano
color.ui=auto
alias.lg=log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s
(bold blue)<%an>%Creset' --abbrev-commit
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=git@github.com:RubenPallin/DesarrolloInterfaces.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
```

EJERCICIO 2

Crear un repositorio nuevo con el nombre del libro y mostrar su contenido.

Empezamos creando una carpeta con el nombre libro.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DI
$ mkdir libro
```

Entramos en ella con el “cd” y mostramos su contenido con el git init y nos dice que está vacío:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces
$ mkdir libro

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces
$ cd libro

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/rubip/DesarrolloInterfaces/libro/.git/

```

Hacemos ls -la

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ ls -la
total 4
drwxr-xr-x 1 rubip rubip 0 Sep 25 12:23 .
drwxr-xr-x 1 rubip rubip 0 Sep 25 12:23 ..
drwxr-xr-x 1 rubip rubip 0 Sep 25 12:23 .git

```

EJERCICIO 3

1. Comprobar el estado del repositorio.
2. Crear un fichero indice.txt con el siguiente contenido:
 Capítulo 1: Introducción a Git
 Capítulo 2: Flujo de trabajo básico
 Capítulo 3: Repositorios remotos
3. Comprobar de nuevo el estado del repositorio.
4. Añadir el fichero a la zona de intercambio temporal.
5. Volver a comprobar una vez más el estado del repositorio.

- **Para comprobar el estado del repositorio hacemos el git status.**

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

```

- **Creamos un fichero con los distintos capítulos llamado "indice.txt", esto lo haremos más sencillo con el cat y al acabar damos Ctrl D y comprobamos que se ha creado:**

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DI/libro
$ cat > indice.txt
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Repositorios remotos

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DI/libro
$ ls -l
total 1
-rw-r--r-- 1 rubip rubip 105 Sep 19 10:10 indice.txt

```

- Volvemos a hacer el git status

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    indice.txt

```

- Luego el git add *

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/l
$ git add indice.txt

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/l
$ git status
On branch master

No commits yet

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

```

Por último, otra vez el git status

EJERCICIO 4

Realizar un commit de los últimos cambios con el mensaje “Añadido índice del libro.” y ver el estado del repositorio.

-Hacemos un commit del índice añadido y volvemos a hacer el status

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DI/libro
$ git commit -m "Añadido índice del libro"
[master (root-commit) b2f0105] Añadido índice del libro
1 file changed, 3 insertions(+)
create mode 100644 indice.txt

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DI/libro
$ git status
On branch master
nothing to commit, working tree clean

```

EJERCICIO 5

Volvemos a hacer el cat para cambiar el contenido del indice.txt:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ cat > indice.txt
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Gestión de ramas
Capítulo 4: Repositorios remotos
```

Seguimos haciendo el git diff que permite comparar cambios en archivos

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git diff
diff --git a/indice.txt b/indice.txt
index 8e62a31..3c4c2a7 100644
--- a/indice.txt
+++ b/indice.txt
@@ -1,3 +1,4 @@
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
-Capítulo 3: Repositorios remotos
+Capítulo 3: Gestión de ramas
+Capítulo 4: Repositorios remotos
\ No newline at end of file
```

Y por último lo añadimos con el git add * y lo comiteamos.

EJERCICIO 6

Para mostrar los cambios de la última versión del repo, lo que haremos será un git show:

Nos muestra lo siguiente.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git show
commit 3fd4d0591fb37559038ff4e689283255c2c0741a (HEAD -> master)
Author: Ruben Pallin <rpallin@fpcoruna.afundacion.org>
Date: Mon Sep 25 12:28:37 2023 +0200

    Añadido capítulo 3 sobre gestión de ramas

diff --git a/indice.txt b/indice.txt
index 8e62a31..3c4c2a7 100644
--- a/indice.txt
+++ b/indice.txt
@@ -1,3 +1,4 @@
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
-Capítulo 3: Repositorios remotos
+Capítulo 3: Gestión de ramas
+Capítulo 4: Repositorios remotos
\ No newline at end of file
```

Ahora para cambiarlo debemos hacer el git commit de siempre pero añadiendo el --amend, seguido del -m "Añadido ...".

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git commit --amend -m "Añadido capítulo 3 sobre gestión de ramas al índice."
[master 8bbc1cb] Añadido capítulo 3 sobre gestión de ramas al índice.
Date: Mon Sep 25 12:28:37 2023 +0200
1 file changed, 2 insertions(+), 1 deletion(-)

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git show
commit 8bbc1cb851aa6b74135463ee81417834bcf67b7d (HEAD -> master)
Author: Ruben Pallin <rpalling@fpcoruna.afundacion.org>
Date: Mon Sep 25 12:28:37 2023 +0200

    Añadido capítulo 3 sobre gestión de ramas al índice.

```

Ahora lo que nos muestra es el commit que hemos actualizado.

Ejercicios de manejo del historial de cambios

EJERCICIO 1

Para mostrar el historial de cambios lo que haremos será un git log.

Añadimos la carpeta capítulos con el mkdir y dentro creamos el capitulo1.txt con el cat, posteriormente añadimos el texto:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ mkdir capitulos

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ cat > capitulos/capitulo1.txt
Git es un sistema de control de versiones ideado por Linus Torvalds.

```

Lo siguiente será añadirlo con el git add y comitearlo con el git commit -m "".

Y por último el git log para ver si se ha commiteado correctamente:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git log
commit c635192baa2814003f2d803c72a550eebdcc3e1f (HEAD -> master)
Author: Ruben Pallin <rpalling@fpcoruna.afundacion.org>
Date: Mon Sep 25 12:34:08 2023 +0200

    Añadido capítulo 1.

```

EJERCICIO 2

Este ejercicio es igual que el anterior pero creando el capítulo 2.

Lo único que cambia es que al final nos pide mostrar la diferencia entre la última versión y las dos versiones anteriores.

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ cat > capitulos/capitulo2.txt
El flujo de trabajo básico con Git consiste en:
1- Hacer cambios en el repositorio.
2- Añadir los cambios a la zona de intercambio temporal.
3- Hacer un commit de los cambios.
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git add .

```

Para ello haremos el git diff git diff HEAD~2..HEAD:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git diff HEAD~2..HEAD
diff --git a/capitulos/capitulo1.txt b/capitulos/capitulo1.txt
new file mode 100644
index 00000000..f4068a7
--- /dev/null
+++ b/capitulos/capitulo1.txt
@@ -0,0 +1 @@
+Git es un sistema de control de versiones ideado por Linus Torvalds.
\ No newline at end of file
diff --git a/capitulos/capitulo2.txt b/capitulos/capitulo2.txt
new file mode 100644
index 00000000..f2a72ed
--- /dev/null
+++ b/capitulos/capitulo2.txt
@@ -0,0 +1,4 @@
+El flujo de trabajo básico con Git consiste en:
+1- Hacer cambios en el repositorio.
+2- Añadir los cambios a la zona de intercambio temporal.
+3- Hacer un commit de los cambios.
\ No newline at end of file

```

EJERCICIO 3

Mismo ejercicio que los anteriores.

Creamos el capitulo3.txt con el cat y le añadimos texto.

Seguimos añadiendolo y comiteandolo.

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ cat > capitulos/capitulo3.txt
Git permite la creación de ramas lo que permite tener distintas versiones del mismo proy
jar de manera simultanea en ellas.
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git add .

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git commit -m "Añadido capítulo 3."
[master c159bec] Añadido capítulo 3.
1 file changed, 1 insertion(+)
create mode 100644 capitulos/capitulo3.txt

```

Lo que cambia es mostrar la diferencia entre la primera y la última versión del repositorio. Por lo que haremos lo siguiente:

Hacemos un git log para así copiar los primeros 5 dígitos de nuestro primer commit:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DI
$ git log
commit 8fad5bcb99e4762c305d07e87888bdc06acff158 (HEAD -> main)
Author: Ruben Pallin <rpalling@fpcoruna.afundacion.org>
Date:   Wed Sep 20 09:08:21 2023 +0200

    Añadido capítulo 3

commit 98a1c518fdce5f4d7fd58272a2d2141ee0c98693
Author: Ruben Pallin <rpalling@fpcoruna.afundacion.org>
Date:   Wed Sep 20 09:04:05 2023 +0200

    Añadido capítulo 2

commit e57ad3fefba846d1ff6b8b00ae2fb3ae2ed8bf60
Author: Ruben Pallin <rpalling@fpcoruna.afundacion.org>
Date:   Wed Sep 20 09:00:14 2023 +0200

    Añadido capítulo 1

commit 6b59d21767c27ec775326810e9f47ddf92a719ee
Author: Ruben Pallin <rpalling@fpcoruna.afundacion.org>
Date:   Wed Sep 20 08:47:35 2023 +0200

    Añadido capítulo 3 sobre gestión de ramas al índice.

```

En mi caso serían estos:

```
commit c159b
```

Y por último haremos el siguiente git diff:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git diff c159b..HEAD

[3]+  Stopped                  git diff

```

EJERCICIO 4

Para añadir al final del fichero la línea que nos dice lo que haremos será:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ echo "Capítulo 5: Conceptos avanzados" >> indice.txt

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git add .

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git commit -m "Añadido capítulo 5 al índice."
[master 2fefb43] Añadido capítulo 5 al índice.
 2 files changed, 1 insertion(+), 1 deletion(-)
 create mode 100644 ..HEAD

```

Una vez añadido, comitemos.

Y por último, para mostrar quién ha hecho cambios sobre el fichero tenemos que hacer el git annotate:


```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git annotate indice.txt
7535d673      (Ruben Pallin  2023-09-25 12:25:53 +0200      1)Capítulo 1: Introducción a Git
7535d673      (Ruben Pallin  2023-09-25 12:25:53 +0200      2)Capítulo 2: Flujo de trabajo básico
o
8bbc1cb8      (Ruben Pallin  2023-09-25 12:28:37 +0200      3)Capítulo 3: Gestión de ramas
2fefb434      (Ruben Pallin  2023-09-25 12:43:28 +0200      4)Capítulo 4: Repositorios remotosCa
pitulo 5: Conceptos avanzados

```

En él se nos muestra nuestro nombre, fecha y hora de creación de cada capítulo.

Ejercicios de deshacer cambios

EJERCICIO 1

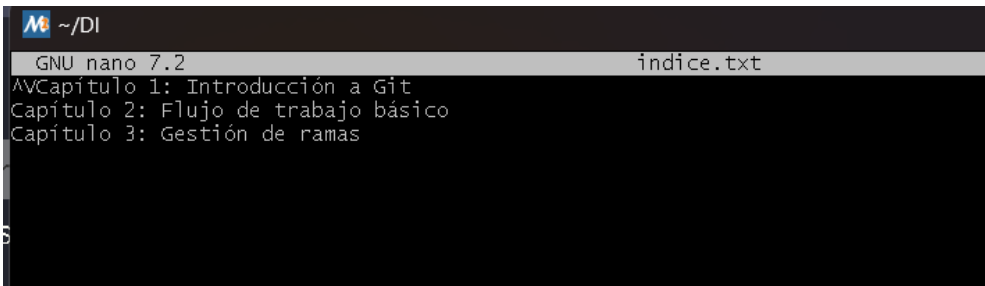
Para eliminar la última línea del fichero indice.txt lo que haremos será entrar dentro del indice.txt

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ nano indice.txt

```

Nos entrará dentro del archivo, borramos la línea y lo guardamos.



```

~/.DI
GNU nano 7.2 indice.txt
AvCapítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Gestión de ramas
$

```

A continuación, haremos un git status en el que nos mostrará que efectivamente hemos modificado nuestro archivo.

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   indice.txt

no changes added to commit (use "git add" and/or "git commit -a")

```

Lo siguiente será hacer el git checkout para deshacer los cambios realizados en el fichero indice.txt.

Y al volver a hacer status veremos que ya no estás el indice.txt en modified.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git checkout -- indice.txt

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master
nothing to commit, working tree clean
```

EJERCICIO 2

Igual que en el anterior ejercicio, para eliminar la última línea del fichero `indice.txt` lo que haremos será entrar dentro del `indice.txt`. Borramos la línea y guardamos.

En cambio ahora, lo guardaremos y veremos con el status que efectivamente se ha eliminado la última línea correctamente.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ nano indice.txt

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git add .

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   indice.txt
```

Para volver a restaurar de nuevo primero deberemos hacer el `git reset` del `indice.txt` y si hacemos el `git status` veremos que está de nuevo con el `modified` pendiente.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   indice.txt

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git reset indice.txt
Unstaged changes after reset:
M    indice.txt

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   indice.txt
```

Volvemos a hacer de nuevo como el ejercicio anterior, el git checkout para deshacer los cambios realizados en el fichero indice.txt.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git checkout -- indice.txt
```

Y al volver a hacer status veremos que ya no estás el indice.txt en modified.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master
nothing to commit, working tree clean
```

EJERCICIO 3

Igual que en los anteriores ejercicios, para eliminar la última línea del fichero indice.txt lo que haremos será entrar dentro del indice.txt

```
rubip@LAPTOP-NPOCB3A4 MINGW64
$ nano indice.txt
```

Y ahí dentro la borramos.

Lo siguiente, para eliminar el capitulo3.txt hacemos el rm en la dirección donde se encuentra este archivo:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ rm capitulos/capitulo3.txt
```

Para crear un nuevo capítulo lo que haremos será hacer el touch que es más cómodo seguido de capítulos y el capítulo que queramos añadir. Listamos y ahí está:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ touch capitulos/capitulo4.txt
```

Ahora añadimos y comprobamos el estado:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git add .

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    capitulos/capitulo3.txt
    new file:   capitulos/capitulo4.txt
    modified:   indice.txt
```

Ahí aparecen los cambios esperando al commit.

Ahora para quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo. Lo que haremos será el git reset. Y git status para ver el estado.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git reset
Unstaged changes after reset:
D   capitulos/capitulo3.txt
M   indice.txt

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:   capitulos/capitulo3.txt
        modified:  indice.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        capitulos/capitulo4.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Seguimos y ahora ejecutaremos el “git checkout - -.” que se utiliza para descartar los cambios no confirmados en los archivos que se encuentran en el directorio de trabajo y le volvemos a dar status

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git checkout -- .

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        capitulos/capitulo4.txt

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

A continuación haremos un git clean -f para eliminar el capitulo4.txt.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git clean -f
Removing capitulos/capitulo4.txt

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git status
On branch master
nothing to commit, working tree clean
```

Ya no hay nada.

EJERCICIO 4

igual que en los anteriores ejercicios, para eliminar la última línea del fichero indice.txt lo que haremos será entrar dentro del indice.txt

```
rubip@LAPTOP-NPOCB3A4 MINGW64  
$ nano indice.txt
```

Y ahí dentro la borramos.

Repetimos el mismo proceso para eliminar el capitulo3.txt, hacemos un rm del capítulo que hemos dicho. Listamos y ya no está.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro  
$ rm capitulos/capitulo3.txt
```

Lo siguiente que haremos será commitearlo.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro  
$ git commit -a -m "Borrado accidental"  
[master 7a5add1] Borrado accidental  
2 files changed, 2 deletions(-)  
delete mode 100644 capitulos/capitulo3.txt
```

Y un git status seguido de un git log para ver que se ha añadido correctamente.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro  
$ git status  
On branch master  
nothing to commit, working tree clean  
  
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro  
$ git log  
commit 7a5add1a0dc883f8a6b34cd78a9168156605230a (HEAD -> master)  
Author: Ruben Pallin <rpalling@fpacoruna.afundacion.org>  
Date: Mon Sep 25 12:59:54 2023 +0200  
  
Borrado accidental
```

A continuación, para deshacer el último commit pero mantener los cambios anteriores en el directorio de trabajo y la zona de intercambio temporal, habrá que hacer el siguiente comando:

git reset --soft HEAD~1 que usamos para deshacer el último commit en tu rama actual.

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro  
$ git reset -- soft HEAD~1
```

Después volveremos a hacer el mismo commit que antes.

Pero esta vez, para deshacer el último commit y los cambios anteriores del directorio de trabajo volviendo a la versión anterior del repositorio.

Hacemos el siguiente comando:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro  
$ git reset --hard HEAD~1
```

Ejercicios de gestión de ramas

EJERCICIO 1

Para crear una nueva rama bibliografía y mostrar las ramas del repositorio lo que tenemos que hacer es:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git branch bibliografia
```

A continuación, haremos un `git branch -v` que se utiliza para listar todas las ramas en tu repositorio de Git y así comprobar si se ha creado:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git branch -av
  bibliografia 7a5add1 Borrado accidental
* master       7a5add1 Borrado accidental
```

EJERCICIO 2

Creamos el `capitulo4.txt` y le añadimos el texto como ya habíamos hecho anteriormente, con el `cat`:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ cat > capitulos/capitulo4.txt
En este capítulo veremos cómo usar GitHub para alojar repositorios en remoto
```

Seguimos con lo de siempre, añadimos y le damos commit:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git add .

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git commit -m "Añadido capítulo 4."
[master f43b173] Añadido capítulo 4.
1 file changed, 1 insertion(+)
create mode 100644 capitulos/capitulo4.txt
```

Por último, para mostrar la historia del repositorio incluyendo todas las ramas, lo que tendremos que hacer será el siguiente comando:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git log --graph --all --oneline
* f43b173 (HEAD -> master) Añadido capítulo 4.
* 7a5add1 (bibliografia) Borrado accidental
* 2fefb43 Añadido capítulo 5 al índice.
* c159bec Añadido capítulo 3.
* 46b5f0c Añadido capítulo 2.
* c635192 Añadido capítulo 1.
* 8bbc1cb Añadido capítulo 3 sobre gestión de ramas al índice.
* 7535d67 Añadido índice del libro.

```

Este se utiliza para mostrar un registro de los commits.

EJERCICIO 3

En este ejercicio nos pide cambiar a la rama que previamente creamos, por lo tanto haremos el git checkout y la rama:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git checkout bibliografia
Switched to branch 'bibliografia'

```

Una vez dentro creamos el bibliografia.txt con el texto dentro mediante el cat:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ cat > bibliografia.txt
- Chacon, S. and Straub, B. Pro Git. Apress.

```

Como siempre añadimos y le damos commit:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git add .

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git commit -m "Añadida primera referencia bibliográfica."
[bibliografia 4ac00b8] Añadida primera referencia bibliográfica.
1 file changed, 1 insertion(+)
create mode 100644 bibliografia.txt

```

Y por último, como en el ejercicio anterior, mostramos la historia del repositorio incluyendo todas las ramas. Con el git log --graph --all --oneline:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git log --graph --all --oneline
* 4ac00b8 (HEAD -> bibliografia) Añadida primera referencia bibliográfica.
| * f43b173 (master) Añadido capítulo 4.
|/
* 7a5add1 Borrado accidental
* 2fefb43 Añadido capítulo 5 al índice.
* c159bec Añadido capítulo 3.
* 46b5f0c Añadido capítulo 2.
* c635192 Añadido capítulo 1.
* 8bbc1cb Añadido capítulo 3 sobre gestión de ramas al índice.
* 7535d67 Añadido índice del libro.

```

EJERCICIO 4

Para Fusionar la rama bibliografia con la rama master primero cambiaremos a nuestra rama master con el git checkout y haremos un git merge:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git checkout master
Switched to branch 'master'
```

Nos entrará en este documento y habrá que cerrarlo

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git merge bibliografia
Merge made by the 'ort' strategy.
 bibliografia.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 bibliografia.txt
```

Lo siguiente, para mostrar la historia del repositorio incluyendo todas las ramas es el comando que ya hicimos anteriormente:

git log --graph --all --oneline

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git log --graph --all --oneline
* 5d340a2 (HEAD -> master) Merge branch 'bibliografia'
| \
| * 4ac00b8 (bibliografia) Añadida primera referencia bibliográfica.
* | f43b173 Añadido capítulo 4.
|/
* 7a5add1 Borrado accidental
* 2fefb43 Añadido capítulo 5 al índice.
* c159bec Añadido capítulo 3.
* 46b5f0c Añadido capítulo 2.
* c635192 Añadido capítulo 1.
* 8bbc1cb Añadido capítulo 3 sobre gestión de ramas al índice.
* 7535d67 Añadido índice del libro.
```

En el que se puede observar el mergeo con la rama bibliografia.

Ahora eliminamos la rama bibliografia que hemos creado con el git branch -d bibliografia:

```
rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git branch -d bibliografia
Deleted branch bibliografia (was 4ac00b8).
```

Y por último repetimos el segundo paso de mostrar la historia del repositorio con el git log --graph --all --oneline.


```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git log --graph --all --oneline
* 5d340a2 (HEAD -> master) Merge branch 'bibliografia'
|
| * 4ac00b8 Añadida primera referencia bibliográfica.
| * | f43b173 Añadido capítulo 4.
|/
* 7a5add1 Borrado accidental
* 2fefb43 Añadido capítulo 5 al índice.
* c159bec Añadido capítulo 3.
* 46b5f0c Añadido capítulo 2.
* c635192 Añadido capítulo 1.
* 8bbc1cb Añadido capítulo 3 sobre gestión de ramas al índice.
* 7535d67 Añadido índice del libro.

```

EJERCICIO 5

Volvemos a crear la rama bibliografia de nuevo con el git branch.

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git branch bibliografia

```

Y a continuación, nos cambiamos a ella con el git checkout.

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git checkout bibliografia
Switched to branch 'bibliografia'

```

Ahora dentro de la rama cambiamos el contenido de bibliografia.txt entrando con el cat, por el siguiente:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ cat > bibliografia.txt
- Scott Chacon and Ben Straub. Pro Git. Apress.
- Ryan Hodson. Ry's Git Tutorial. Smashwords (2014)

```

Lo siguiente darle commit con el siguiente texto:

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git commit -a -m "Añadida nueva referencia bibliográfica."
[bibliografia 0551c26] Añadida nueva referencia bibliográfica.
1 file changed, 2 insertions(+), 1 deletion(-)

```

Volvemos a cambiar a la rama main en mi caso con el git checkout.

Y modificar con hicimos en la rama bibliografia el bibliografia.txt con el cat.

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git checkout master
Switched to branch 'master'

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ cat > bibliografia.txt
- Chacon, S. and Straub, B. Pro Git. Apress.
- Loeliger, J. and McCullough, M. Version control with Git. O'Reilly.

```

Añadimos y le damos commit.

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git commit -a -m "Añadida nueva referencia bibliográfica."
[master 4d091c6] Añadida nueva referencia bibliográfica.
1 file changed, 2 insertions(+), 1 deletion(-)

```

Lo siguiente será mergear bibliografia.

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git merge bibliografia
Auto-merging bibliografia.txt
CONFLICT (content): Merge conflict in bibliografia.txt
Automatic merge failed; fix conflicts and then commit the result.

```

Nos dará un error que tendremos que corregir en el bibliografia.txt

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ nano bibliografia.txt

```

Lo siguiente, será commitearlo.

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git commit -a -m "Solucionado conflicto bibliografía."
[master ea14e80] Solucionado conflicto bibliografía.

```

Y para acabar haremos el siguiente comando:

git log --graph --all --oneline

```

rubip@LAPTOP-NPOCB3A4 MINGW64 ~/DesarrolloInterfaces/libro
$ git log --graph --all --oneline
* ea14e80 (HEAD -> master) Solucionado conflicto bibliografía.
| \
|  * 0551c26 (bibliografia) Añadida nueva referencia bibliográfica.
| * | 4d091c6 Añadida nueva referencia bibliográfica.
| /
| * 5d340a2 Merge branch 'bibliografia'
| \
|  * 4ac00b8 Añadida primera referencia bibliográfica.
| * | f43b173 Añadido capítulo 4.
| /
| * 7a5add1 Borrado accidental
| * 2fefb43 Añadido capítulo 5 al índice.
| * c159bec Añadido capítulo 3.
| * 46b5f0c Añadido capítulo 2.
| * c635192 Añadido capítulo 1.
| * 8bbc1cb Añadido capítulo 3 sobre gestión de ramas al índice.
| * 7535d67 Añadido índice del libro.

```