



# **Git**

Práctica Guiada.

# Git

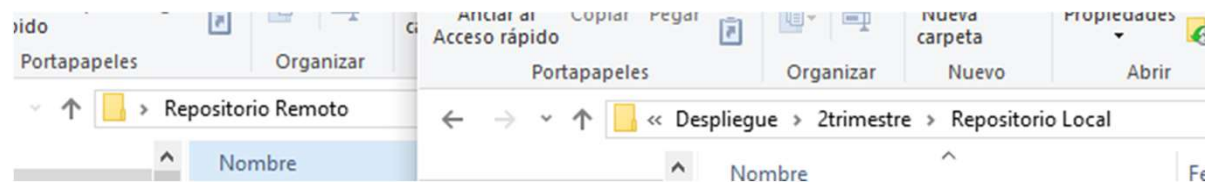
Git es un sistema de control de versiones que nos permite realizar un seguimiento de los cambios en el código a lo largo del tiempo.

Git es un sistema usado ampliamente en el trabajo colaborativo en empresas. El funcionamiento es simple, desde un repositorio remoto se descarga el código haciendo una copia del repositorio en local, se trabaja, se implementan cambios y se devuelve al repositorio remoto desde el que otros colaboradores pueden descargarse el código y trabajar.

Para nuestra práctica necesitaremos dos carpetas, una hará de repositorio remoto y la otra de repositorio local.

# Git

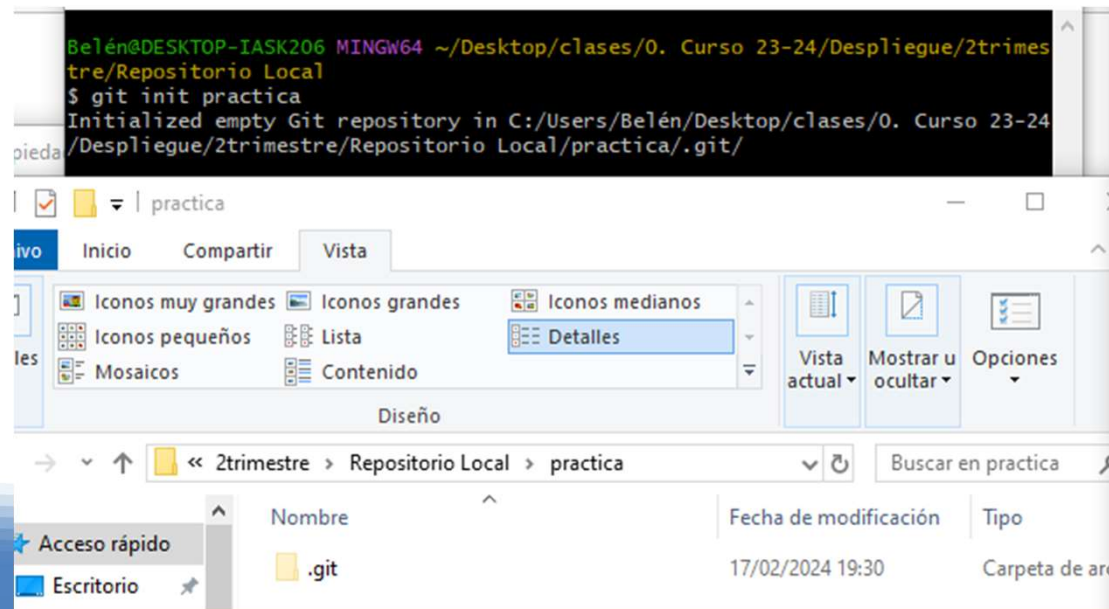
En el escritorio creamos una carpeta y la otra en cualquier parte del árbol de directorios de nuestro equipo.



En principio las dos carpetas están vacías.

# Git

En este caso lo que vamos a hacer es crear un proyecto de 0 y subirlo al remoto. Para ello en el repositorio local abrimos la terminal de git y con el comando “git init nombreProyecto” iniciamos un nuevo repositorio git en este caso en local. Este comando crea un fichero oculto .git dentro del proyecto que hemos creado, este fichero será el que vaya registrando los cambios. Para verlo necesitaremos mostrar ficheros ocultos (desde windows en la pestaña vista).



# Git

Si entramos en nuestro repositorio en la consola ya veremos que estamos en la rama master, vamos a renombrarla como rama main con el comando “git branch -M main”

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local
$ ls
practica/

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local
$ cd practica

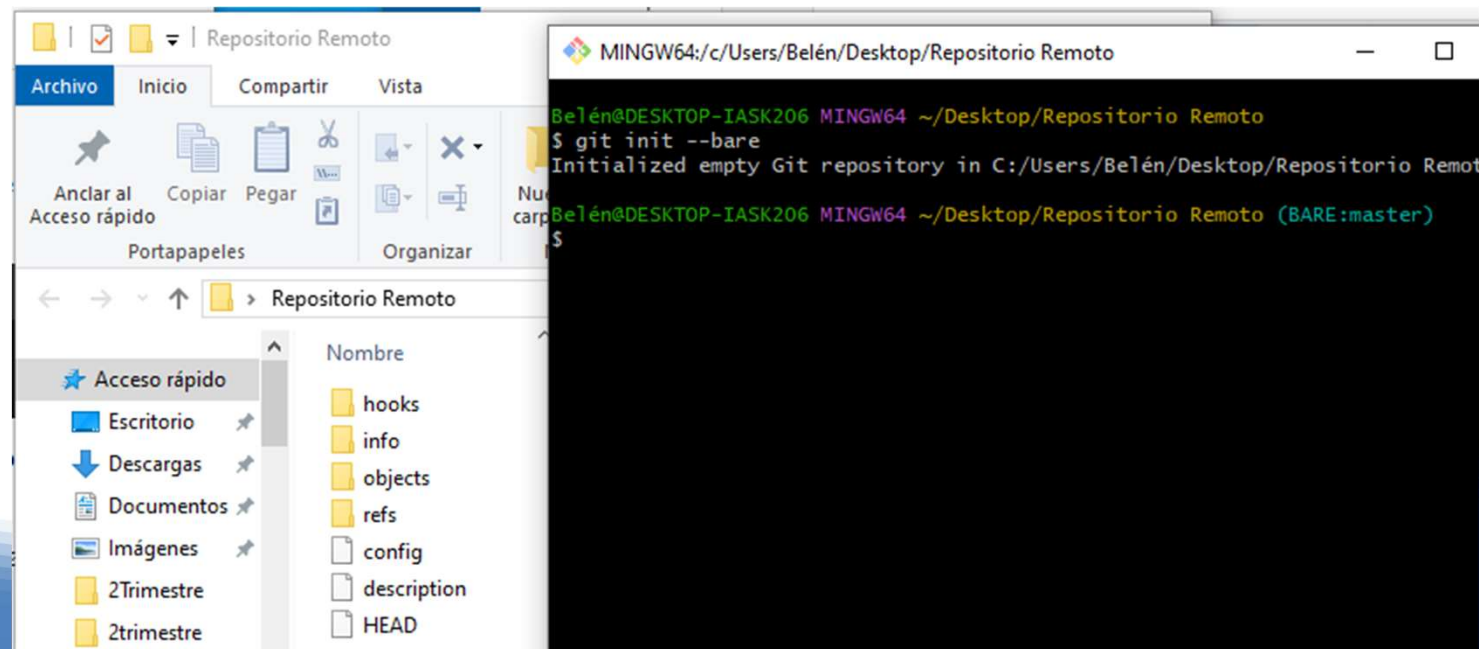
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (master)
$ git branch -M main

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
```

Aquí vemos como se ha pasado de rama master a rama main.

# Git

Tenemos configurado el repositorio local pero debemos configurar el repositorio remoto, donde se va a guardar nuestro código. En la carpeta del repositorio local iniciamos un nuevo repositorio vacío con “git init - -bare”, si vemos se nos ha creado una estructura de directorios dentro de nuestro repositorio Remoto.



# Git

Solo nos queda enlazar ambos repositorios para poder trabajar. Nos vamos a nuestro repositorio local y le decimos cual va a ser el repositorio remoto, normalmente utilizaríamos una url pero como la ruta está en nuestro local usaremos ["file:///ruta/de/nuestro/repositorio"](file:///ruta/de/nuestro/repositorio) Con el siguiente código:

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git remote add origin file:///C:/Users/Belén/Desktop/Repositorio Remoto

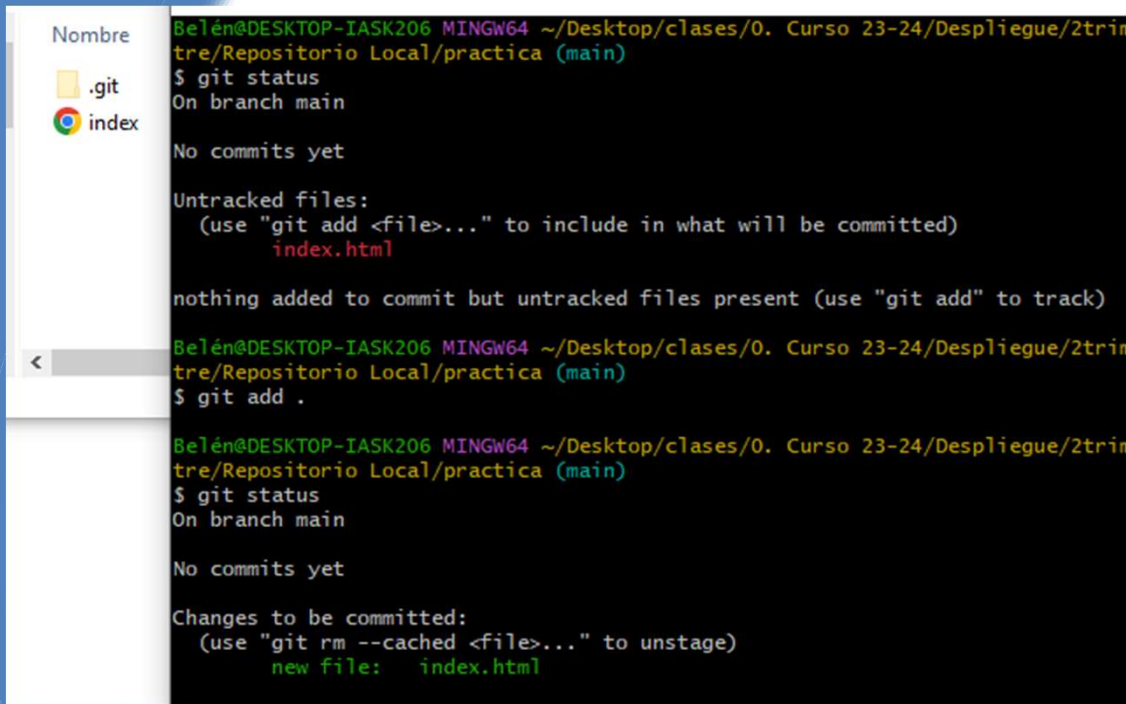
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git remote -v
origin file:///C:/Users/Belén/Desktop/Repositorio Remoto (fetch)
origin file:///C:/Users/Belén/Desktop/Repositorio Remoto (push)

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$
```

Ya estamos listos para trabajar.



# Git



```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git status
On branch main

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git add .

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git status
On branch main

No commits yet

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

En nuestro repositorio local hemos agregado un fichero, en la consola de git con el comando `git status` comprobamos que existe el fichero pero no está preparado para realizar un commit sobre el. Para eso necesitaremos añadirlo con “`git add .`” ( en el caso de que existieran varios ficheros y solo quisiéramos incluir uno de ellos el comando `git add` debe ir seguido del nombre de los ficheros que queramos incluir. ). Si volvemos a realizar `git status` veremos el fichero en verde señal de que ya está preparado.



# Git

Para subir los cambios al repositorio antes debemos realizar un commit, este comando prepara todos los cambios añadidos para subirlos al repositorio remoto. Es aconsejable realizar un comentario descriptivo de los cambios que se producen en el commit:

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$ git commit -m "Mi primer commit"
[main (root-commit) e2b27fe] Mi primer commit
1 file changed, 44 insertions(+)
create mode 100644 index.html
```

Con git log podemos ver los cambios pendientes de subir:

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$ git log
commit e2b27fe8be5026b6bb55db11068a54e1ec17b4dd (HEAD -> main)
Author: beguik <69437315+beguik@users.noreply.github.com>
Date: Sat Feb 17 20:00:18 2024 +0100

    Mi primer commit

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$
```

# Git

Solo nos queda subir los cambios al repositorio remoto, para eso usamos el comando git push

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 793 bytes | 793.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To file:///C:/Users/Belén/Desktop/Repositorio Remoto
 * [new branch]      main -> main

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$
```

# Git

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$ git status
On branch main
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:   index.html

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

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$ git diff
diff --git a/index.html b/index.html
index e69281e..ad699d4 100644
--- a/index.html
+++ b/index.html
@@ -7,7 +7,7 @@
   <link rel="stylesheet" href="styles.css"> <!-- Este será tu archivo CSS compi
lado -->
</head>
<body>
-
+<p>Este es un cambio que realizo para mi práctica de git</p>
  <header>
    <h1>Mi Página de Práctica</h1>
    <p>¡Experimenta con Sass y dale estilo a esta página!</p>
```

Hemos realizado la primera subida pero ahora podemos cambiar nuestro fichero y ver las diferencias con el fichero guardado en un repositorio. Con git status vemos los ficheros modificados, con git diff vemos cual ha sido la diferencia, en verde saldrá lo añadido, en rojo lo eliminado.

# Git

Con “git add .” Lo añado y con git status compruebo de nuevo que ya está incluido para realizar commit. Realizo el commit.

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$ git add .

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$ git commit -m "realizo un cambio de prueba"
[main 956c24b] realizo un cambio de prueba
1 file changed, 1 insertion(+), 1 deletion(-)
```

# Git

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$ git push
fatal: The current branch main has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin main

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
```

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimes
tre/Repositorio Local/practica (main)
$ git push --set-upstream origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 359 bytes | 359.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To file:///C:/Users/Belén/Desktop/Repositorio Remoto
    e2b27fe..956c24b  main -> main
branch 'main' set up to track 'origin/main'.
```

Estamos listos para realizar un push pero nos puede surgir un fallo porque la rama main no está configurada en el origen (repositorio remoto). En ese caso deberemos configurarla con:

```
git push - -set-upstream origin main.
```



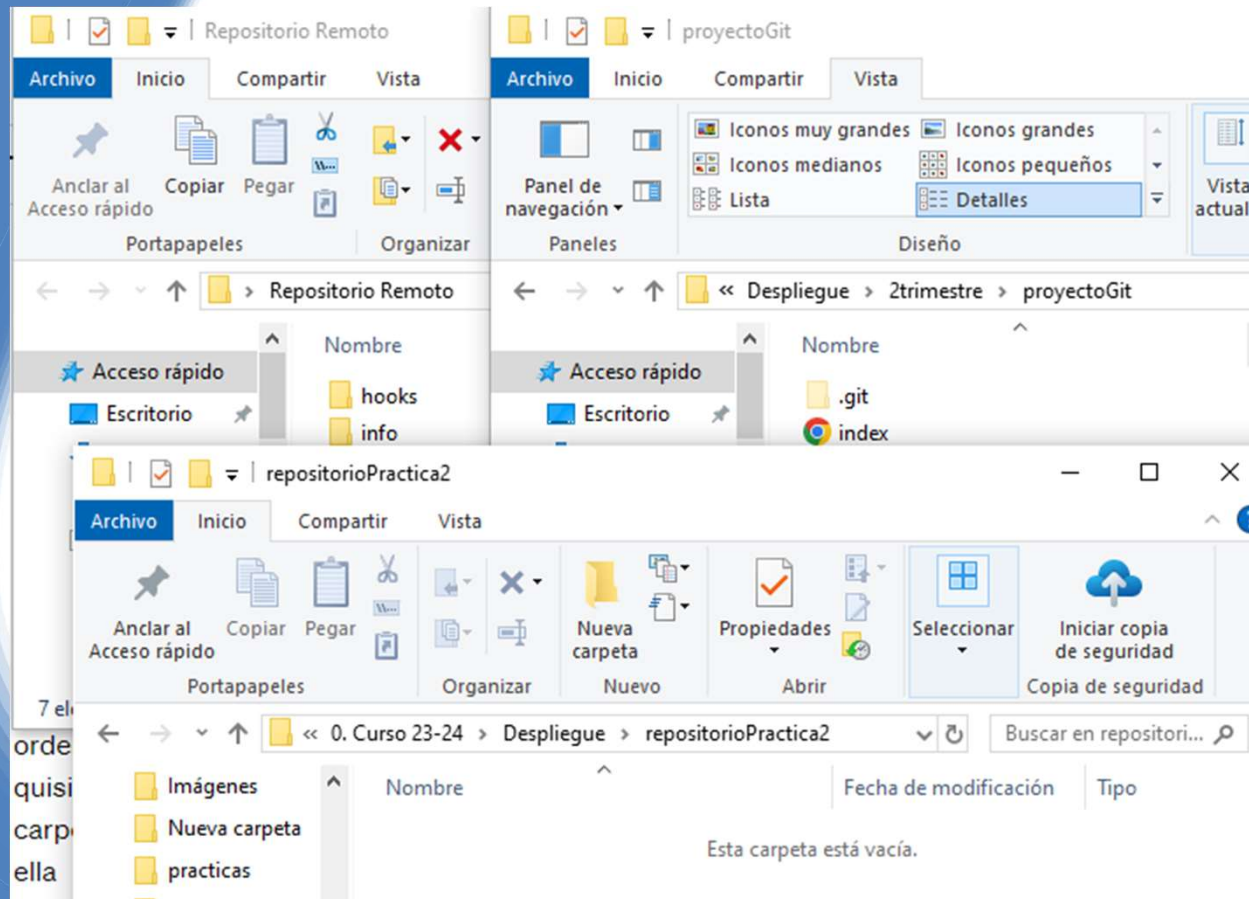
# Git

Bien, imaginemos que entramos nuevos en una empresa donde se trabaja con git (que será muy probable), tenemos que acceder a los repositorios en los que los compañeros ya llevan trabajando tiempo.... Como accedemos a ellos para trabajar nosotros y comenzar a desarrollar. Para trasladar eso a la práctica vamos a suponer que la carpeta remota sería nuestro servidor central, y la carpeta que hemos creado antes para nuestra práctica sería el equipo de cualquier compañero, nosotros como nuevos en la empresa tendríamos nuestro escritorio vacío, pues con este símil crearemos una nueva carpeta ajena a las dos que hemos trabajado anteriormente.



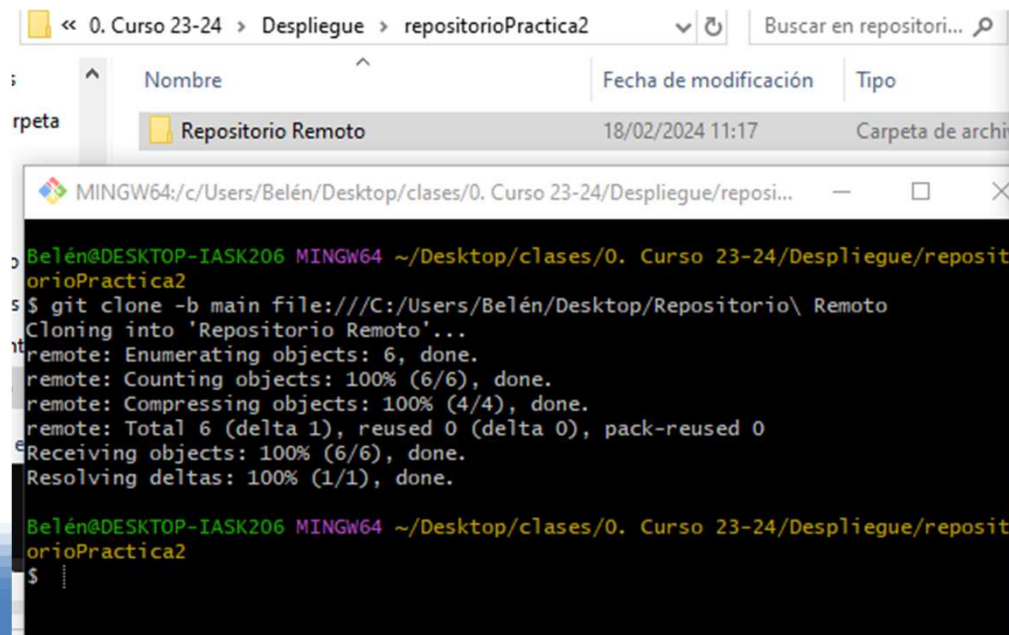
# Git

Vemos que tenemos por un lado el Repositorio Remoto, por otro el proyecto git con el que hemos trabajado antes y por último el repositorio nuevo que está vacío.



# Git

Abrimos una terminal git en nuestro nuevo repositorio y clonamos en este caso la rama main del repositorio remoto, comprobamos que se ha creado correctamente porque en nuestra carpeta que antes estaba vacía aparece el repositorio remoto



The screenshot shows a Windows File Explorer window and a Git terminal window. The File Explorer window displays the path '0. Curso 23-24 > Despliegue > repositorioPractica2' and contains a folder named 'Repositorio Remoto' with a modification date of '18/02/2024 11:17'. The terminal window, titled 'MINGW64:/c/Users/Belén/Desktop/clases/0. Curso 23-24/Despliegue/reposi...', shows the command 'git clone -b main file:///C:/Users/Belén/Desktop/Repositorio\ Remoto' being executed. The output of the command is as follows:

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/repositorioPractica2
$ git clone -b main file:///C:/Users/Belén/Desktop/Repositorio\ Remoto
Cloning into 'Repositorio Remoto'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
Resolving deltas: 100% (1/1), done.

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/repositorioPractica2
$
```

# Git

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue
orioPractica2
$ cd Repositorio\ Remoto

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue
orioPractica2/Repositorio Remoto (main)
$ git branch
* main

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue
orioPractica2/Repositorio Remoto (main)
$ git checkout -b pruebas
Switched to a new branch 'pruebas'

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue
orioPractica2/Repositorio Remoto (pruebas)
$ git branch
main
* pruebas

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue
orioPractica2/Repositorio Remoto (pruebas)
$
```

Imaginemos ahora que somos nuevos y no queremos cagarla.... No sería muy práctico trabajar sobre la rama main porque si luego la subimos al repositorio podríamos cargarnos algo que antes funcionaba..... Vamos a andar con pies de plomo y vamos a hacer una rama a partir de la rama main y nuestros cambios de esta forma no afectarán a la rama main:

Con git checkout -b “nombre de la rama” creamos la rama a partir de la actual y además nos movemos a ella, con git branch listamos las ramas que existen en nuestro repositorio local y con un asterisco nos marca en la que estamos.

# Git

Ahora podemos modificar, comprobar, commitear y subir este commit al repositorio remoto, para ello hay que indicar en el push el nombre de la nueva rama:

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/orioPractica2/Repositorio Remoto (pruebas)
$ git status
On branch pruebas
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:   index.html

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

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/orioPractica2/Repositorio Remoto (pruebas)
$ git add .

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/orioPractica2/Repositorio Remoto (pruebas)
$ git status
On branch pruebas
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/orioPractica2/Repositorio Remoto (pruebas)
$ git commit -m "modificación desde el segundo repositorio"
[pruebas c84891c] modificación desde el segundo repositorio
1 file changed, 1 insertion(+)

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/orioPractica2/Repositorio Remoto (pruebas)
$ git push -u origin pruebas
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 370 bytes | 370.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To file:///C:/Users/Belén/Desktop/Repositorio Remoto
 * [new branch]      pruebas -> pruebas
branch 'pruebas' set up to track 'origin/pruebas'.
```



# Git

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git branch
* main

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git fetch origin
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 350 bytes | 4.00 KiB/s, done.
From file:///C:/Users/Belén/Desktop/Repositorio Remoto
* [new branch]      pruebas -> origin/pruebas

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git branch -a
* main
remotes/origin/main
remotes/origin/pruebas
```

Pero volvamos a nuestro primer repositorio local, si listamos en él las ramas veremos que solo aparece la que el creó, es decir la rama main, ¿como accedemos entonces a la rama nueva que hemos creado en el segundo repositorio?. Primero deberemos actualizar los cambios que hayan podido producirse en el repositorio remoto:

Con `git branch` listamos y vemos como estamos en la rama que creamos al comienzo, actualizamos la información del repositorio remoto con `git fetch origin` y listamos las ramas del repositorio remoto con `git branch -a`.

# Git

Una vez que tengo conocimiento de las diferentes ramas puedo ir a la que quiera, en este caso a la nueva de pruebas.

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git checkout -b prueba origin/pruebas
Switched to a new branch 'prueba'
branch 'prueba' set up to track 'origin/pruebas'.

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (prueba)
$ git branch
  main
* prueba
```

Con git log podemos comprobar todos los cambios realizados en la rama:

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (prueba)
$ git log
commit c84891cb60f62467884552263bdb7f0023cfdb85 (HEAD -> prueba, origin/pruebas)
Author: beguik <69437315+beguik@users.noreply.github.com>
Date:   Sun Feb 18 11:30:16 2024 +0100

    modificación desde el segundo repositorio

commit 956c24b8d5a5f1f44f07affa2ff046afafeeece5 (origin/main, main)
Author: beguik <69437315+beguik@users.noreply.github.com>
Date:   Sun Feb 18 10:57:52 2024 +0100

    realizo un cambio de prueba

commit e2b27fe8be5026b6bb55db11068a54e1ec17b4dd
Author: beguik <69437315+beguik@users.noreply.github.com>
Date:   Sat Feb 17 20:00:18 2024 +0100

    Mi primer commit
```



# Git

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (prueba)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git merge prueba
Updating 956c24b..c84891c
Fast-forward
 index.html | 1 +
 1 file changed, 1 insertion(+)

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git commit -m "Incorporación de la rama prueba a la rama main"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio Local/practica (main)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To file:///C:/Users/Belén/Desktop/Repositorio Remoto
 956c24b..c84891c  main -> main

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/Despliegue/2trimestre/Repositorio
```

Imaginemos por un momento que yo he visto los cambios realizados en la rama prueba y me parecen correctos, quiero incorporar los cambios a la rama main. Me debo volver a colorar en la rama main y realizar un merge, es decir una unión, el merge incorporará los cambios de la rama prueba en la rama main y subiremos los cambios al repositorio remoto.

# Git

Podemos comprobar los cambios de la siguiente forma, situados en la rama main del primer repositorio si accedemos al fichero con el que hemos estado trabajando podremos ver la modificación que hayamos realizado en el segundo repositorio.

Por último es cierto que se puede trabajar con tantas ramas como se quiera pero una vez que los cambios se han acometido y la rama ya no tiene validez sería recomendable eliminarla ya que si no hacemos limpieza al final tendremos una estructura demasiado compleja.

# Git

Para eliminar la rama prueba primero tendremos que asegurarnos de que no estamos en la rama que queremos eliminar, luego la eliminaremos localmente y por último la eliminaremos del repositorio remoto.

```
Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/
o Local/practica (main)
$ git branch -d prueba
Deleted branch prueba (was c84891c).

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/
o Local/practica (main)
$ git branch
* main

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/
o Local/practica (main)
$ git branch -a
* main
  remotes/origin/main
  remotes/origin/pruebas

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/
o Local/practica (main)
$ git push origin --delete pruebas
To file:///C:/Users/Belén/Desktop/Repositorio Remoto
[deleted]

Belén@DESKTOP-IASK206 MINGW64 ~/Desktop/clases/0. Curso 23-24/
o Local/practica (main)
$ git branch -a
* main
  remotes/origin/main
```

# Git

Esta es la estructura general de como se trabaja en un entorno de preproducción mejorando el código de determinado proyecto, o añadiendo nuevas funcionalidades. En el trabajo real pueden surgir conflictos con las actualizaciones puesto que puede que un compañero realice modificaciones en la misma rama que tu estás trabajando y a la hora de realizar cambios se produzcan desajustes entre ramas. En esos casos lo mejor es una buena organización.

En primer lugar cuando os toque trabajar en un desarrollo descargaros la rama en la que vais a trabajar, crear la vuestra propia, para que no se solape con la de vuestros compañeros, trabajar en ella hasta que el código esté correcto, y una vez que eso ocurra vuestro superior será el encargado de revisar los cambios y añadirlos a la rama principal si considera que son correctos.