

Practica GIT y GITHUB en Docker (2ª Parte)

Realiza las siguientes actividades capturando pantalla de todos los pasos.

1. Ficheros README.md, .gitignore y HEAD

1. Edita el fichero readme.md y complétalo.

-Nos vamos a la carpeta donde está configurado el repositorio que en este caso es en la carpeta de Documents/Despliegue/Practicas-Git

```
Practica-Git : bash — Konsole
wally97@Wally-Linux:~/Documents/Despliegue$ ls -l
total 1828
drwxrwxr-x 3 wally97 wally97 4096 nov 13 13:40 0.Javadocs
-rw-rw-r-- 1 wally97 wally97 300331 nov 13 13:41 0.Javadocs.zip
-rw-rw-r-- 1 wally97 wally97 141916 oct 21 15:22 '5.Si modificamos el index.html ,¿A que contenedor afectará x
c3 o c4?.jpg'
drwxrwxr-x 3 wally97 wally97 4096 nov 20 13:05 de_Dios_Sanchez_Jesus_LMSGI02_Tarea
drwxrwxr-x 2 wally97 wally97 4096 nov 14 10:52 Examen
-rw-rw-r-- 1 wally97 wally97 122688 oct 21 15:14 pantallazo1.jpg
-rw-rw-r-- 1 wally97 wally97 135132 oct 21 15:13 pantallazo2.jpg
-rw-rw-r-- 1 wally97 wally97 64701 oct 21 15:07 pantallazo3.jpg
-rw-rw-r-- 1 wally97 wally97 65281 oct 21 15:10 pantallazo4.jpg
drwxrwxr-x 2 wally97 wally97 4096 oct 23 18:00 'Practica 5.1'
drwxrwxr-x 3 wally97 wally97 4096 nov 20 21:12 Practica-Git
-rw-rw-r-- 1 wally97 wally97 487856 oct 21 15:22 'Tarea 4.2 Bind mount para compartir datos.zip'
drwxrwxr-x 2 wally97 wally97 4096 oct 23 18:37 'Tarea 5.2: Conexión de redes a contenedores'
-rw-rw-r-- 1 wally97 wally97 270046 oct 23 18:40 'Tarea 5.2: Conexión de redes a contenedores.zip'
drwxrwxr-x 2 wally97 wally97 4096 oct 23 19:42 'Tarea 6.1: Crear tu propia imagen a partir de un contenedor'
-rw-rw-r-- 1 wally97 wally97 243507 oct 23 19:42 'Tarea 6.1: Crear tu propia imagen a partir de un contenedor.
zip'
wally97@Wally-Linux:~/Documents/Despliegue$ cd Practicas-Git
bash: cd: Practicas-Git: No such file or directory
wally97@Wally-Linux:~/Documents/Despliegue$ cd Practica-Git
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ ls -l
total 8
-rw-rw-r-- 1 wally97 wally97 122 nov 20 21:11 index.html
-rw-rw-r-- 1 wally97 wally97 14 nov 20 21:01 README.md
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ nano README.md
```

```
Practica-Git : nano — Konsole
GNU nano 7.2                                README.md *
# Practica-Git
## Repositorio practica git
```

CTRL+O para guardar y CTRL+X para salir

2. Actualízalo en tu repositorio remoto

Con la secuencia de comandos ,git add para añadir nuevos archivos,git commit -m “mensaje”

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git status
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
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ nano README.md
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git add .
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git commit -m "Editado readme"
[main 954679e] Editado readme
1 file changed, 1 insertion(+), 1 deletion(-)
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git push
Username for 'https://github.com': jdds97
Password for 'https://jdds97@github.com':
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 650 bytes | 650.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/jdds97/Practica-Git
 fea7c0e..954679e main -> main
```

para poner un mensaje al cambio y git push para subir los cambios al repositorio remoto.

3. Crea un fichero gitignore para que GIT no haga un seguimiento de las copias de seguridad.

-Creamos con el comando nano un archivo .gitignore y decimos que todos los archivos que empiecen por la extensión .bak sean ignorados.

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ nano .gitignore
```



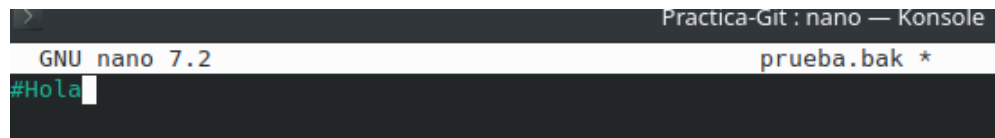
The screenshot shows the nano editor interface with the title 'Practica-Git : nano — Konsole'. The status bar at the top indicates 'GNU nano 7.2' and '.gitignore *'. The main text area contains the following content: `#Copias de seguridad` and `*.bak` with the cursor at the end of the second line.

CTRL+O para guardar y CTRL+X para salir

4. Para demostrar su funcionamiento, no es necesario compilar un programa java, simplemente crea un fichero con la extensión adecuada.

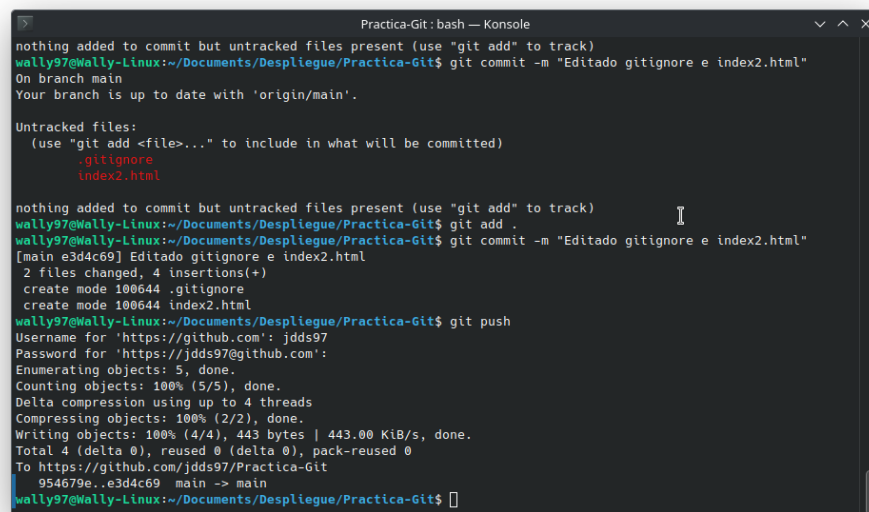
-Creamos con el comando nano un archivo .bak ,más adelante actualizaremos los repositorios y podremos ver que no se ha subido el archivo prueba.bak

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ nano prueba.bak
```



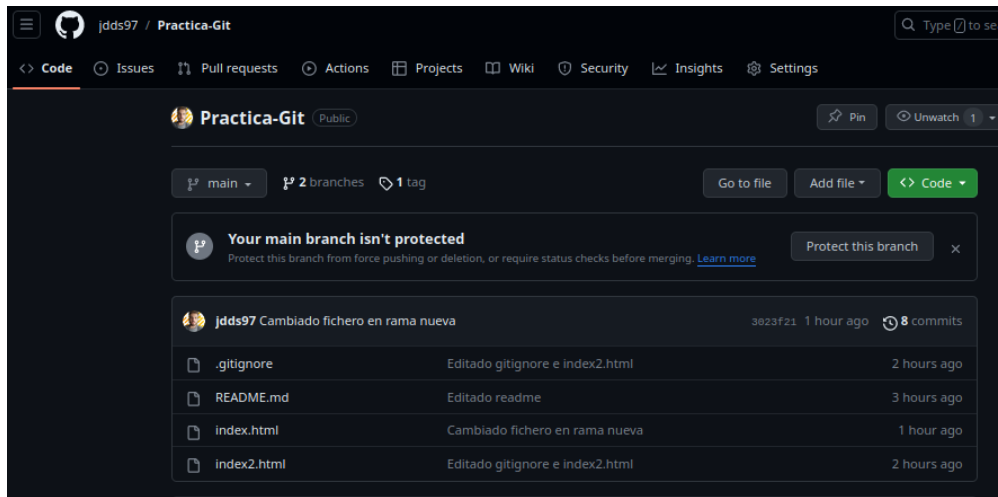
The screenshot shows the nano editor interface with the title 'Practica-Git : nano — Konsole'. The status bar at the top indicates 'GNU nano 7.2' and 'prueba.bak *'. The main text area contains the following content: `#Hola` with the cursor at the end of the line.

CTRL+O para guardar y CTRL+X para salir



The screenshot shows a terminal window titled 'Practica-Git: bash — Konsole'. The output of the following commands is displayed:
1. `git commit -m "Editado gitignore e index2.html"`: Shows that the branch is up to date with 'origin/main'.
2. `git add .`: Shows untracked files (.gitignore and index2.html) and adds them to the staging area.
3. `git commit -m "Editado gitignore e index2.html"`: Shows the commit details: 2 files changed, 4 insertions(+), creating new modes for .gitignore and index2.html.
4. `git push`: Shows the push process, including username and password prompts, and the upload progress to the remote repository.

5. Actualiza los repositorios remotos



6. Captura un pantallazo de los logs

-Con el comando git log podremos ver todos los logs de nuestro repositorio.

```
Practica-Git: git — Konsole
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git log
commit e3d4c693b137d4821a13b86a6c80d542cb220410 (HEAD -> main, origin/main, origin/HEAD)
Author: Jesús de Dios Sánchez <jesusedtossanchez@gmail.com>
Date: Tue Nov 21 11:29:21 2023 +0100

    Editado gitignore e index2.html

commit 954679e3ca2f1d7955839fc5e86afea820c8ba58
Author: Jesús de Dios Sánchez <jesusedtossanchez@gmail.com>
Date: Tue Nov 21 11:10:05 2023 +0100

    Editado readme

commit d6ccff9cff2d562fe0296e1798c4dbc1081ecfe3
Author: Jesús de Dios Sánchez <jesusedtossanchez@gmail.com>
Date: Tue Nov 21 11:07:33 2023 +0100

    Editado readme

commit fea7c0e6ae2f4f1255cb9598b822ad50a876e0b9
Author: Jesús de Dios Sánchez <jesusedtossanchez@gmail.com>
Date: Mon Nov 20 21:12:08 2023 +0100

    Añadido documento html nuevo

commit b836ca41a9e5353f9732798dd5891af93aa0a99c
Author: Jesús de Dios Sánchez <jesusedtossanchez@gmail.com>
Date: Mon Nov 20 21:09:22 2023 +0100
```

2. Versiones y ramas

1. Etiqueta tu repositorio como versión 1.0

-Con el comando git tag -a + versión y -m "Mensaje"

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git tag -a v1.0 -m "Versión 1.0 - Hola mundo"
```

2. Actualiza el repositorio

-Con el git pull actualizaremos el repositorio local

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git pull
Already up to date.
```

3. Actualiza las etiquetas

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git push --tags
Username for 'https://github.com': jdds97
Password for 'https://jdds97@github.com':
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 197 bytes | 197.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/jdds97/Practica-Git
* [new tag]          v1.0 -> v1.0
```

-Subiremos las etiquetas con el comando git push --tags.

4. Comprueba el resultado

-Con el comando git tag -n podremos ver todas las tags que tenemos actualmente asociadas a nuestro repositorio

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git tag -n
v1.0          Versión 1.0 - Hola mundo
```

Crea una rama llamada “experimentafuncion”

-Con el comando git branch + nombre crearemos una nueva rama en nuestro repositorio.

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git branch experimentafuncion
```

5. Salta a la nueva rama y modifica el fichero

-Con el comando git checkout saltaremos a la rama experimentafuncion ,y con la instrucción nano index.html editaremos el fichero.

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git checkout experimentafuncion
Switched to branch 'experimentafuncion'
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ nano index.html
Practica-Git: nano — Konsole

GNU nano 7.2                                index.html *
<h1>Nuevo documento HTML!</h1>
<h1>Aperitivo : Salmorejo</h1>
<h1>Comida : Puchero</h1>
<h1>Postre : Tarta de queso</h1>
<h1>Modificado fichero rama</h1>
```

CTRL+O para guardar y CTRL+X para salir

6. Actualiza el repositorio remoto y comprueba su funcionamiento

-Con la secuencia de comando git add,git commit -m "" y git push podremos hacerlo siempre que especifiquemos en el git push que se va a hacer en una rama diferente.

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ nano index.html
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git status
On branch experimentafuncion
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")
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git add .
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git commit -m "Cambiado fichero en rama nueva"
[experimentafuncion 3023f21] Cambiado fichero en rama nueva
1 file changed, 1 insertion(+), 1 deletion(-)

wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git push
fatal: The current branch experimentafuncion has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin experimentafuncion

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

wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git push --set-upstream origin experimentafuncion
Username for 'https://github.com': jdds97
Password for 'https://jdds97@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 349 bytes | 349.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'experimentafuncion' on GitHub by visiting:
remote:   https://github.com/jdds97/Practica-Git/pull/new/experimentafuncion
remote:
To https://github.com/jdds97/Practica-Git
 * [new branch]      experimentafuncion -> experimentafuncion
branch 'experimentafuncion' set up to track 'origin/experimentafuncion'.
```

7. Fusiona ambas ramas

-Cambiando a la rama main con el comando git checkout main para irnos a la que no tiene los cambios hechos,pondremos entonces el comando git merge experimentafuncion para que se fusionen los cambios de ambas ramas ,y una vez hecho esto hacer un git push para que se guarden los cambios.

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git merge experimentafu
Updating e3d4c69..3023f21
Fast-forward
 index.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git status
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
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git push
Username for 'https://github.com': jdds97
Password for 'https://jdds97@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/jdds97/Practica-Git
 e3d4c69..3023f21  main -> main
```

8. Comprueba el resultado.

```
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ git branch
experimentafuncion
* main
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ ls -l
total 16
-rw-rw-r-- 1 wally97 wally97 29 nov 21 14:04 index2.html
-rw-rw-r-- 1 wally97 wally97 154 nov 21 14:06 index.html
-rw-rw-r-- 1 wally97 wally97 6 nov 21 11:28 prueba.bak
-rw-rw-r-- 1 wally97 wally97 44 nov 21 11:09 README.md
wally97@Wally-Linux:~/Documents/Despliegue/Practica-Git$ nano index.html
```

Como podemos ver tiene el mismo contenido que en la rama anterior con el nuevo h1 de “Modificado fichero rama.”

```
GNU nano 7.2 index.html
<h1>Nuevo documento HTML!</h1>
<h1>Aperitivo : Salmorejo</h1>
<h1>Comida : Puchero</h1>
<h1>Postre : Tarta de queso</h1>
<h1>Modificado fichero rama</h1>
```