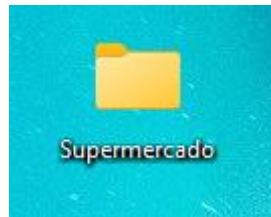
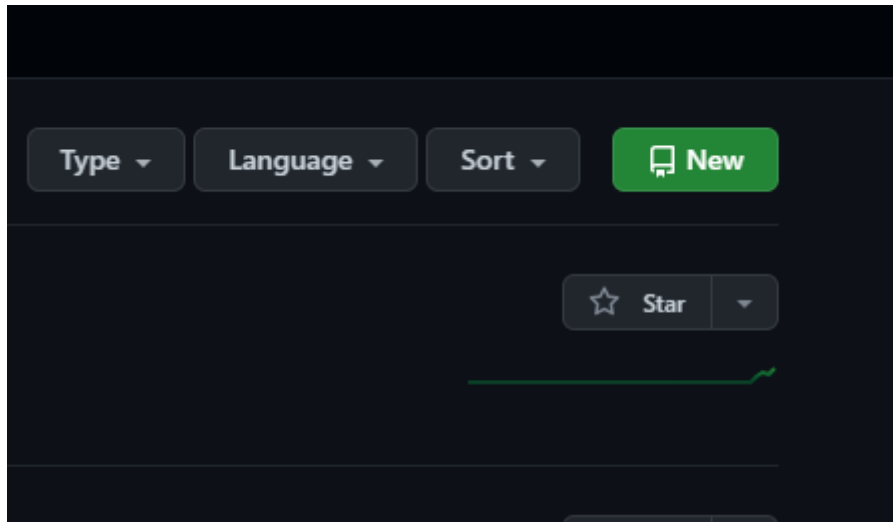


En primer lugar, crearemos la carpeta supermercado para inicializar en ella el repositorio local





En github en el apartado “your repositories” podremos encontrar en la parte superior derecha la opción new la cual nos permitirá crear un repositorio nuevo.



La página requerirá un nombre válido de repositorio, y el repositorio en cuestión podrá ser publico o privado. En la parte inferior derecha podemos encontrar en verde el botón de “crear repositorio”.


Required fields are marked with an asterisk (\*).


Owner \*  Kamikka / Repository name \*

 Supermercado is available.

Great repository names are short and memorable. Need inspiration? How about [cautious-octo-succotash](#) ?

Description (optional)

☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:


☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore  
.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license  
License: **None**

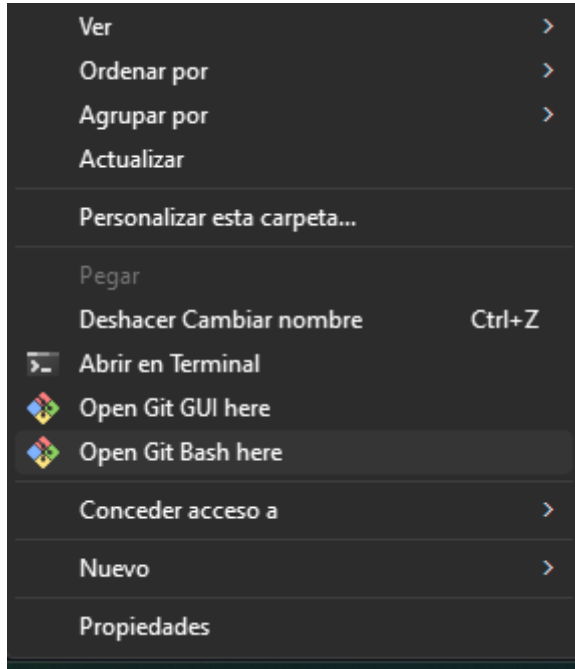
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

[Create repository](#)

En este punto ya tenemos el repositorio remoto creado y vamos a crear el repositorio local.

Para esto debemos hacer click derecho dentro de la carpeta en la que queremos iniciar un repositorio y abrir una consola de git bash.



Una vez abierta la consola debemos introducir el siguiente comando para iniciar un repositorio en la carpeta elegida.

```
MINGW64:/c/Users/alumnofp/Desktop/Supermercado
FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado
$ git init
Initialized empty Git repository in C:/Users/alumnofp/Desktop/Supermercado/.git/
FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$
```

A partir de lo que tenemos queda enlazar el repositorio remoto con el local y eso se hace mediante el siguiente comando.

```
MINGW64:/c/Users/alumnopf/Desktop/Supermercado
FPA11+alumnopf@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ git remote add origin https://github.com/Kamikka/Supermercado.git
FPA11+alumnopf@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ |
```

En este punto tenemos los repositorios enlazados pero aún hay que especificar la rama principal con la que trabajaremos, por lo tanto deberemos crear un archivo cualquiera para poder hacer un primer “push” que nos permita dejar la configuración del enlace de repositorios lista para utilizarla cuando se desee.

Este ha sido el proceso completo de este paso:

```
MINGW64:/c/Users/alumnopf/Desktop/Supermercado
FPA11+alumnopf@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ touch readme.txt
FPA11+alumnopf@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ git add .
FPA11+alumnopf@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ git commit -m "Mi primer commit"
[master (root-commit) c381c40] Mi primer commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 readme.txt
FPA11+alumnopf@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 214 bytes | 214.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Kamikka/Supermercado.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
FPA11+alumnopf@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ |
```

Después creamos el fichero “inventario.txt”, lo añadimos y guardamos los cambios mediante estos comandos respectivamente.

```
MINGW64:/c/Users/alumnofp/Desktop/Supermercado

FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ touch inventario.txt

FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ git add .

FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ git commit -m "[ADD] inventario"
[master 716c9ba] [ADD] inventario
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 inventario.txt

FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$
```

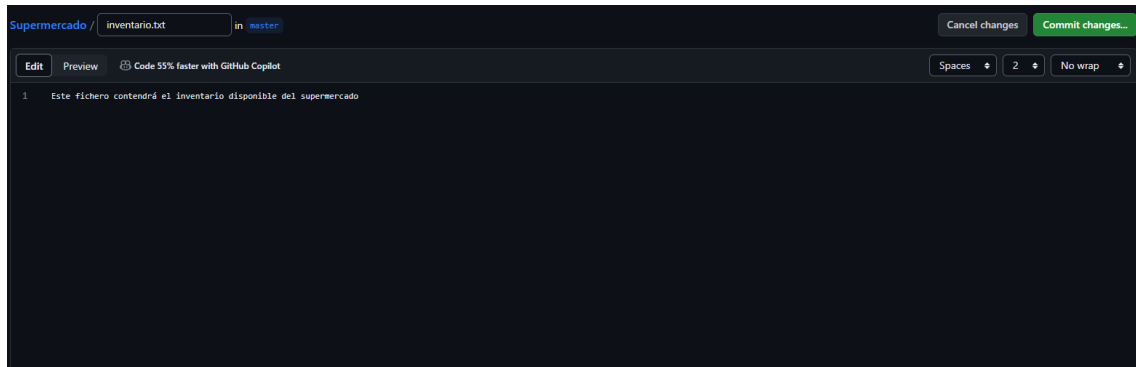
Para subir los cambios del repositorio local al repositorio remoto utilizaremos “push”.

```
MINGW64:/c/Users/alumnofp/Desktop/Supermercado

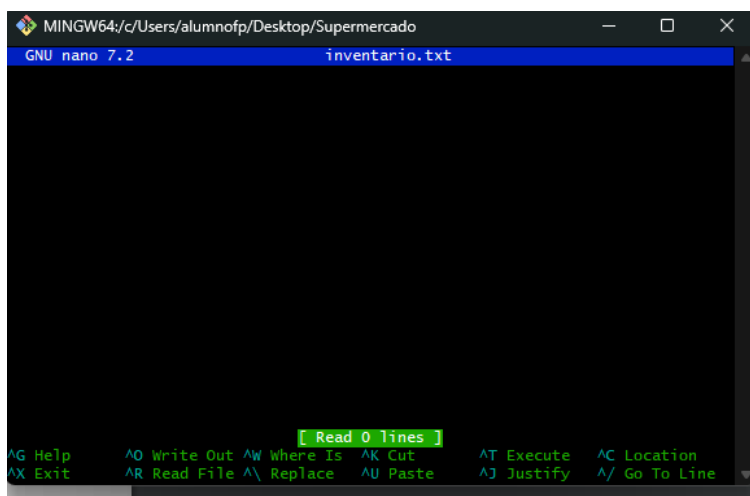
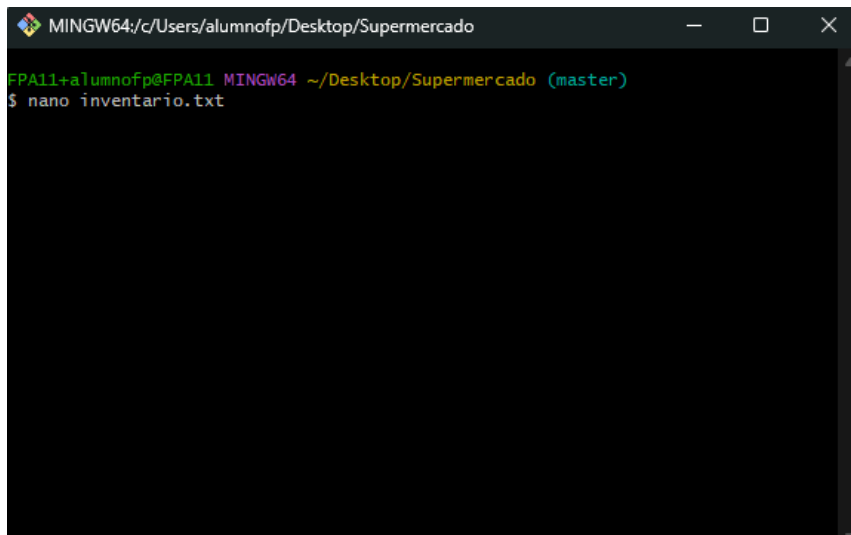
FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 255 bytes | 255.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Kamikka/Supermercado.git
   c381c40..716c9ba  master -> master

FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$
```

En este punto vamos a hacer cambios en el repositorio remoto, una vez hechos los cambios deseados los guardaremos haciendo click sobre “commit changes” arriba a la derecha.



Para comprobar si ha habido algún cambio en el repositorio local, accedemos al archivo mediante nano y como podemos observar no se ha efectuado ningún cambio.



Descargaremos los cambios mediante “pull”

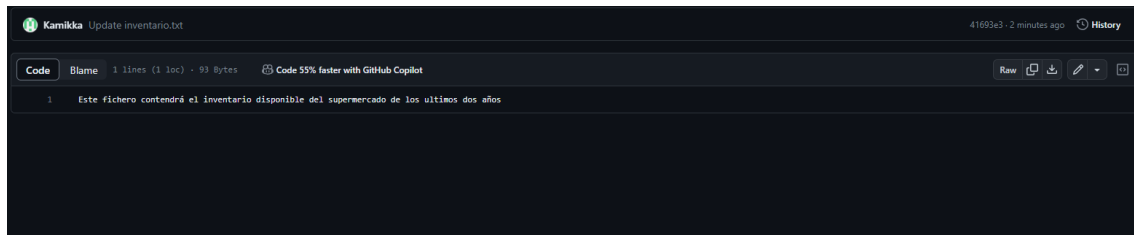
```
MINGW64:/c/Users/alumnofp/Desktop/Supermercado
FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 725 bytes | 90.00 KiB/s, done.
From https://github.com/Kamikka/Supermercado
   716c9ba..7d03d09  master    -> origin/master
Updating 716c9ba..7d03d09
Fast-forward
 inventario.txt | 1 +
 1 file changed, 1 insertion(+)

FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ |
```

Modificaremos los repositorios tanto local (con nano) como remoto (con la opción de editar) y bajaremos el archivo del repositorio remoto para que cree un conflicto.

```
MINGW64:/c/Users/alumnofp/Desktop/Supermercado
GNU nano 7.2 inventario.txt
Este fichero contendrá el inventario disponible del supermercado de este mes

[ Read 2 lines (Converted from DOS format) ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```



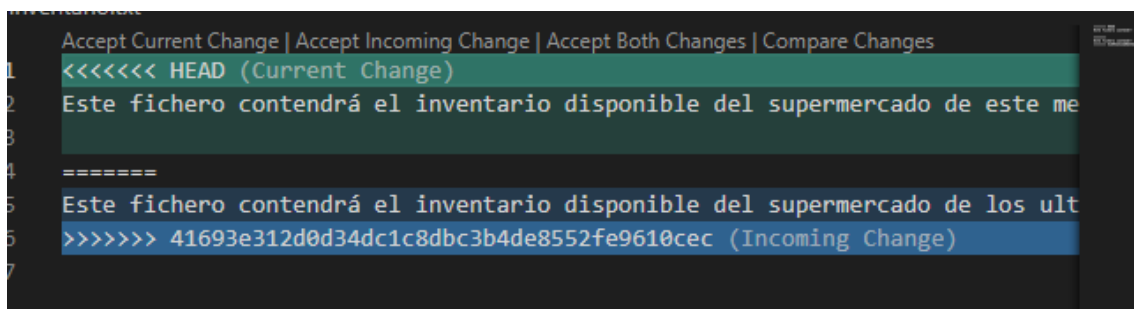
Mediante el comando “mergetool” se abrirá la herramienta de resolución de conflictos y podremos resolver el conflicto para que todo quede correcto.

```
MINGW64:/c/Users/alumnopf/Desktop/Supermercado
To https://github.com/Kamikka/Supermercado.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/Kamikka/Supermercado.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

FPA11+alumnopf@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 742 bytes | 61.00 KiB/s, done.
From https://github.com/Kamikka/Supermercado
   7d03d09..41693e3  master    -> origin/master
Auto-merging inventario.txt
CONFLICT (content): Merge conflict in inventario.txt
Automatic merge failed; fix conflicts and then commit the result.

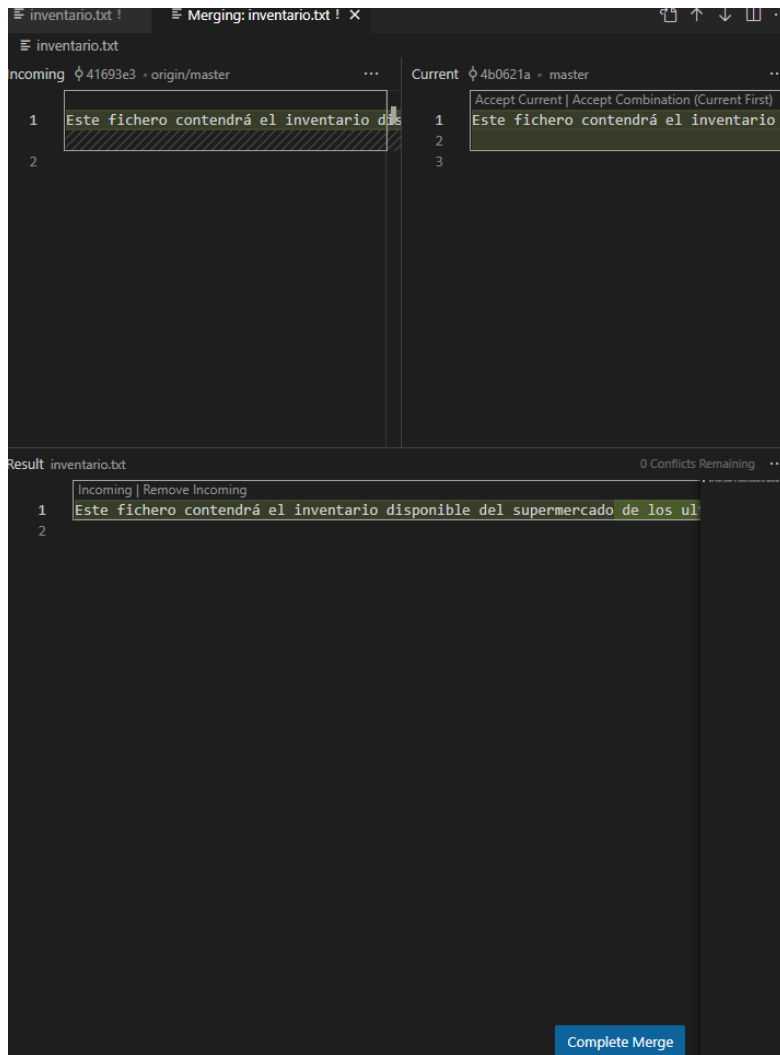
FPA11+alumnopf@FPA11 MINGW64 ~/Desktop/Supermercado (master|MERGING)
$ git mergetool
```

Escogemos el cambio que queremos que permanezca



Seleccionamos la opción “resolve in mergetool”

Seleccionamos “Complete merge”



Y por último hacemos un “commit” para guardar los cambios y completar el merge de los archivos

```

MINGW64:/c/Users/alumnofp/Desktop/Supermercado
Unpacking objects: 100% (3/3), 742 bytes | 61.00 KiB/s, done.
From https://github.com/Kamikka/Supermercado
  7d03d09..41693e3 master -> origin/master
Auto-merging inventario.txt
CONFLICT (content): Merge conflict in inventario.txt
Automatic merge failed; fix conflicts and then commit the result.

FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master|MERGING)
$ git mergetool
Merging:
inventario.txt

Normal merge conflict for 'inventario.txt':
  {local}: modified file
  {remote}: modified file
inventario.txt seems unchanged.
was the merge successful [y/n]? y

FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master|MERGING)
$ git commit -m "ultimo commit"
[master 1e2f1d9] ultimo commit

FPA11+alumnofp@FPA11 MINGW64 ~/Desktop/Supermercado (master)
$

```