



Actividad repositorios GitHub

Gabriel Stiven Chavez Castro

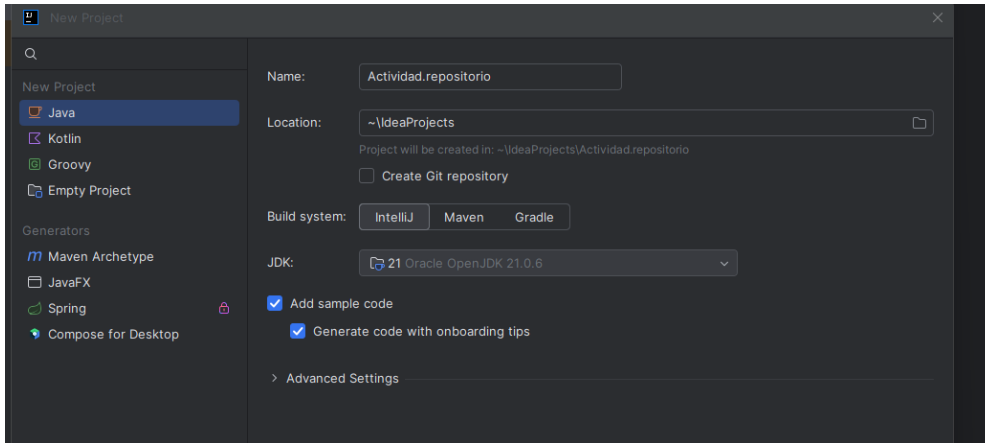
Facultad de Ingeniería, Universidad de Cundinamarca

Grupo 301T Programación II

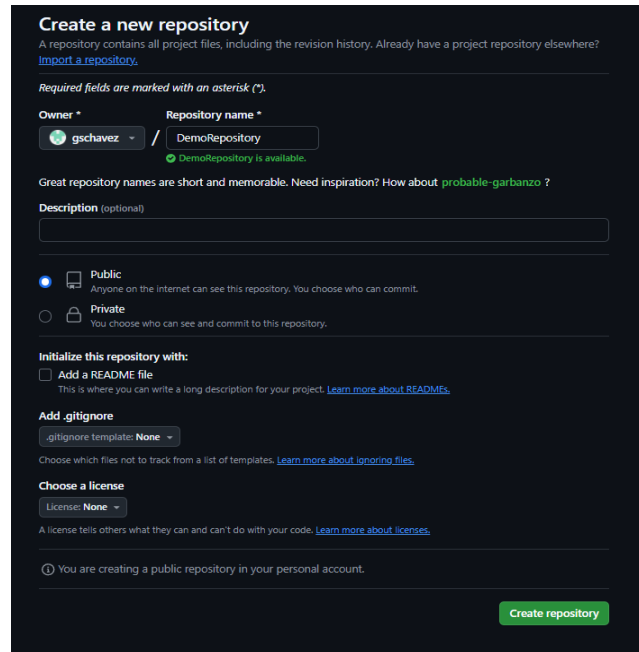
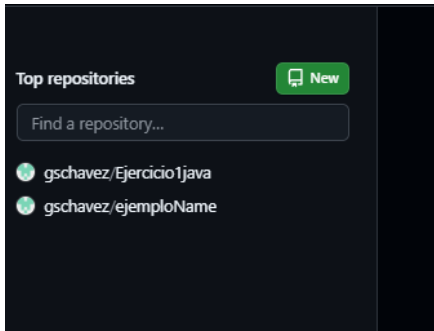
19 de febrero de 20

Manual para el uso del lenguaje git, Creación, uso de repositorios y ramas

1) Primero creamos un proyecto nuevo; desde el contexto de IntelliJ IDEA, jdk 21



2) Creamos el repositorio en GitHub



- 3) Asociamos el proyecto local con el remoto del github copeando el código y pegándolo en la consola.

```
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.
```

...or create a new repository on the command line

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/gschavez/test.git
git push -u origin main
```

```
Terminal  Local x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\stive\IdeaProjects\test> echo "# test" >> README.md
PS C:\Users\stive\IdeaProjects\test> git init
Initialized empty Git repository in C:/Users/stive/IdeaProjects/test/.git/
PS C:\Users\stive\IdeaProjects\test> git add README.md
PS C:\Users\stive\IdeaProjects\test> git commit -m "first commit"
[master (root-commit) b51c4ff] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\Users\stive\IdeaProjects\test> git branch -M main
PS C:\Users\stive\IdeaProjects\test> git remote add origin https://github.com/gschavez/test.git
PS C:\Users\stive\IdeaProjects\test> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 237 bytes | 237.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/gschavez/test.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\stive\IdeaProjects\test>
PS C:\Users\stive\IdeaProjects\test>

test > src > Main > main
```

4) Ahora en la consola del proyecto creado en el entorno de desarrollo, nos logeamos a git.

-Con ayuda de los comandos;

*Git config --global user.name (nos ayuda a vincular el nombre de usuario)

*Git config --global user.email (nos ayuda a vincular el correo)

-Vinculamos el nombre y el correo respectivamente.

```
stive@StivenCh2906 MINGW64 ~/IdeaProjects/Actividad.repositorio (main)
$ git config --global user.name "StivensCh"

stive@StivenCh2906 MINGW64 ~/IdeaProjects/Actividad.repositorio (main)
$ git config --global user.email "gschavez@ucundinamarca.edu.co"
```

-Comprobamos que ya aparezcan en la lista con el git config --list (este código muestra todas las configuraciones de git activas en el entorno)

```
> filter.lfs.required=true
user.name=StivensCh
user.email=gschavez@ucundinamarca.edu.co
core.repositoryformatversion=0
core.filemode=false
```

//Ahora si quisiéramos quitar estos cambios, se puede con ayuda de los siguientes comandos //

*git config --global --unset user.email (desvinculamos el correo)

```
stive@StivenCh2906 MINGW64 ~/IdeaProjects/Actividad.repositorio (main)
$ git config --global --unset user.email
```

*git config --global --unset user.name (desvinculamos el nombre)

```
stive@StivenCh2906 MINGW64 ~/IdeaProjects/Actividad.repositorio (main)
$ git config --global --unset user.name
```

-Comprobamos que no estén en la lista con ayuda del comando visto anteriormente

*Git config --list

```
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/gschavez/DemoRepository.d
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
```

5) Escribir el código del “Main”

-Realizamos los cambios que queremos subir al repositorio remoto



```
1
2 public class Main {
3     public static void main(String[] args) {
4
5         System.out.println("Hola!!! ");
6
7     }
8 }
9 }
```

6) Subir los cambios al repositorio de la rama main con:

*Git status (nos muestra el estado actual del repositorio)

*Git add . (agrega todos los archivos modificados y nuevos cambios, preparándolos para el commit)

*Git commit -m (guarda los cambios añadidos en el historial del repositorio con un mensaje descriptivo)

*Git push origin (sube los commits locales a la rama especificada)

```

PS C:\Users\stive\IdeaProjects\Actividad.repositorio> git status
On branch main
Your branch is up to date with 'origin/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:   src/Main.java

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\stive\IdeaProjects\Actividad.repositorio> git commit -m "se agrega el primer mensaje"
On branch main
Your branch is up to date with 'origin/main'.

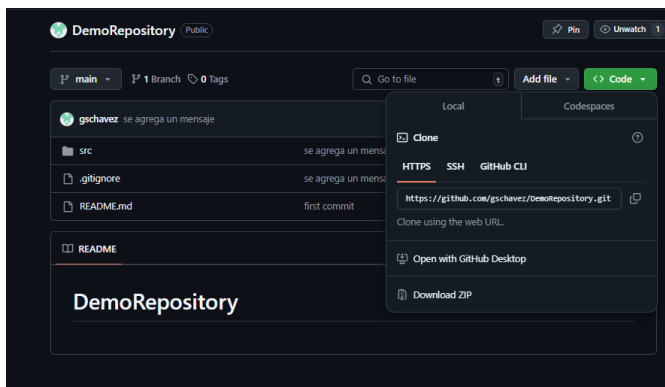
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/Main.java

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\stive\IdeaProjects\Actividad.repositorio> git add .
warning: in the working copy of 'src/Main.java', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\stive\IdeaProjects\Actividad.repositorio> git push origin main

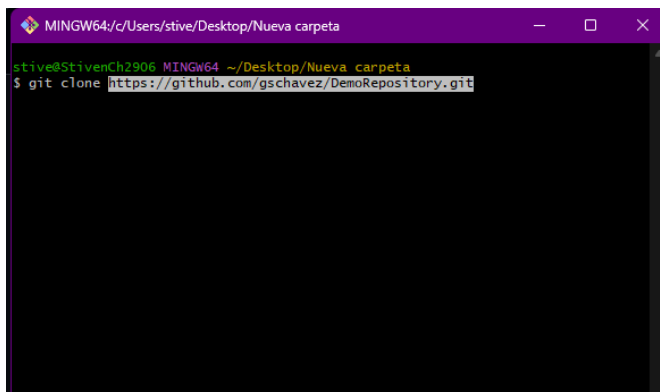
```

7) Clonar el repositorio con git clone

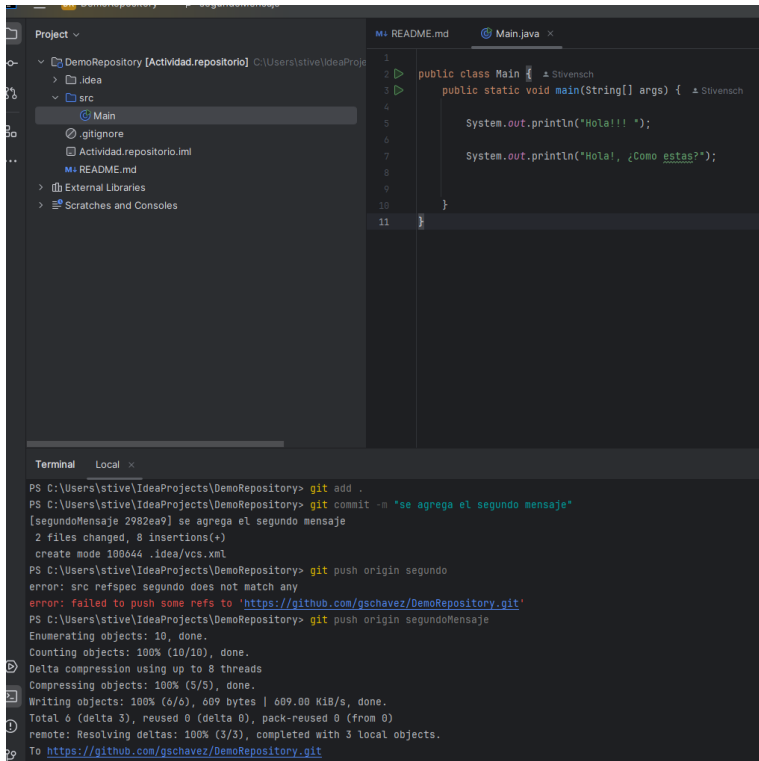
-después de subir los cambios a la rama principal, clonaremos el proyecto remoto para agregar ramas nuevas.



-Pegamos la url del repositorio en la gitbash para clonarlo



8) En el proyecto clonado creamos una rama nueva y se suben los cambios que hallamos echo al repositorio



The screenshot shows the IntelliJ IDEA interface. On the left, the 'Project' view displays the file structure: 'DemoRepository' (root), 'idea' (subdirectory), 'src' (subdirectory), 'Main.java' (file), '.gitignore' (file), 'Actividad.repositorio.iml' (file), 'README.md' (file), 'External Libraries' (directory), and 'Scratches and Consoles' (directory). The 'Main.java' file is open in the editor, showing the following code:

```
1 public class Main {
2     public static void main(String[] args) {
3
4         System.out.println("Hola!!! ");
5
6         System.out.println("Hola!, ¿Como estas?");
7
8     }
9 }
10
11 }
```

The 'Terminal' window at the bottom shows the following output:

```
PS C:\Users\stive\IdeaProjects\DemoRepository> git add .
PS C:\Users\stive\IdeaProjects\DemoRepository> git commit -m "se agrega el segundo mensaje"
[segundoMensaje 2982ea9] se agrega el segundo mensaje
2 files changed, 8 insertions(+)
create mode 100644 .idea/vcs.xml
PS C:\Users\stive\IdeaProjects\DemoRepository> git push origin segundo
error: src refspec segundo does not match any
error: failed to push some refs to 'https://github.com/gschavez/DemoRepository.git'
PS C:\Users\stive\IdeaProjects\DemoRepository> git push origin segundoMensaje
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 609 bytes | 609.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/gschavez/DemoRepository.git
```

para esto usamos;


git Branch (vemos las ramas instanciadas en la consola)

git switch -c (creamos una rama y nos colocamos sobre esta misma)

git push origin (con este comando subimos los cambios al repositorio remoto)

Comprobamos que los cambios estén en la rama creada.

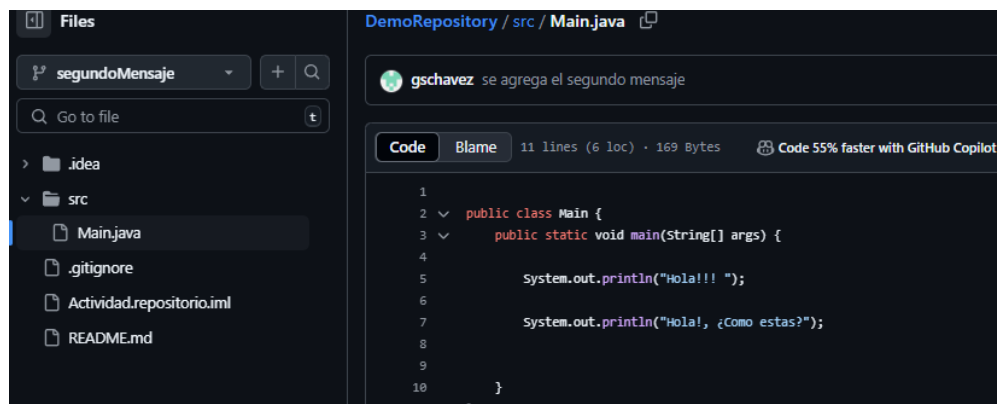
main



The screenshot shows the IntelliJ IDEA interface. The 'Files' view on the left has 'main' selected in the dropdown menu. The 'Main.java' file is open in the editor, showing the following code:

```
1 public class Main {
2     public static void main(String[] args) {
3
4         System.out.println("Hola!!! ");
5
6     }
7 }
8 }
```

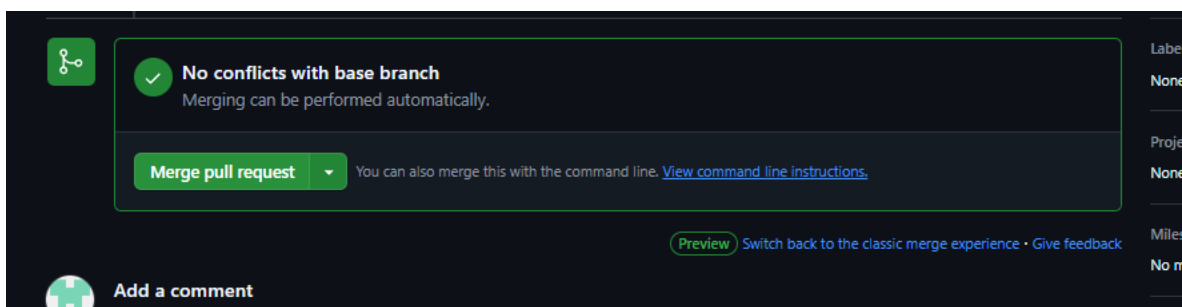
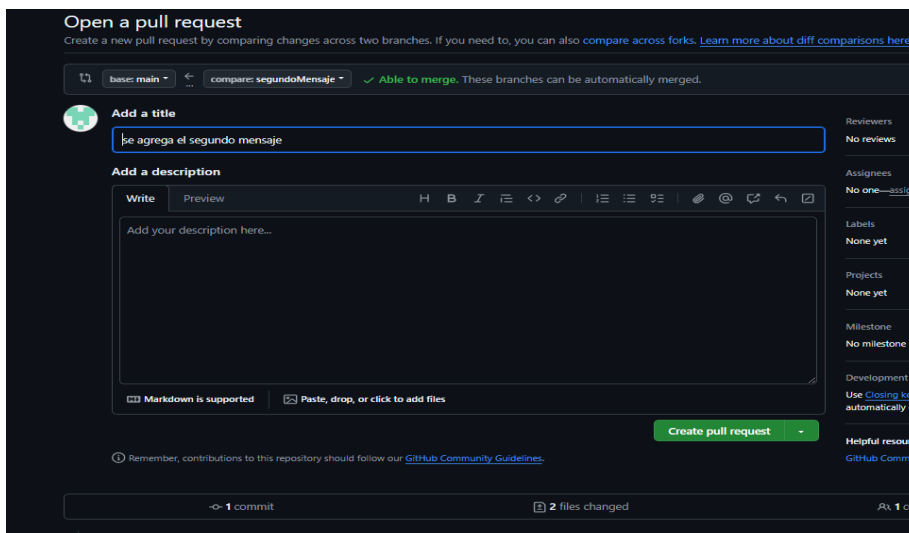
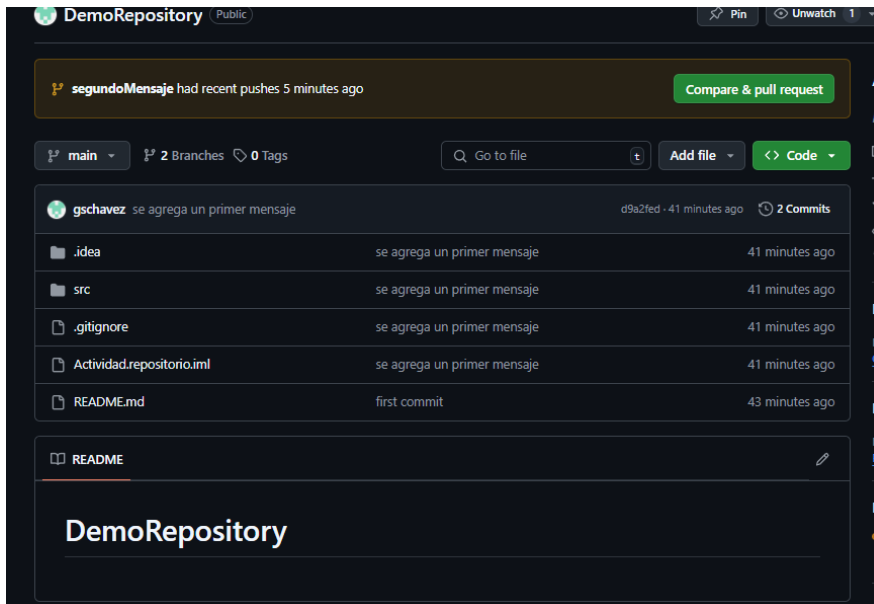
segundoMensaje



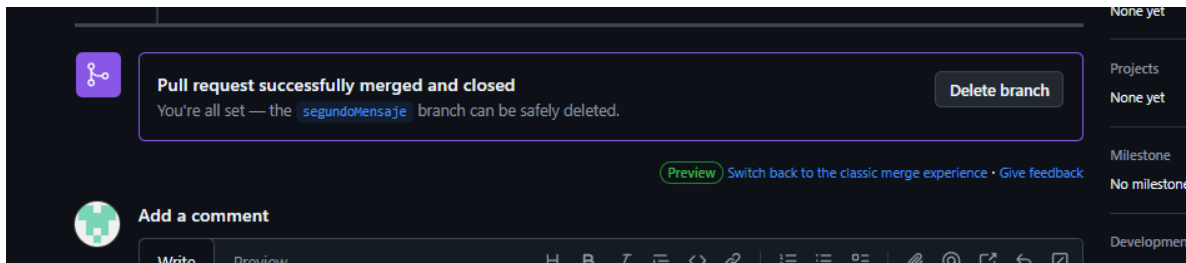
The screenshot shows the IntelliJ IDEA interface. The 'Files' view on the left has 'segundoMensaje' selected in the dropdown menu. The 'Main.java' file is open in the editor, showing the following code:

```
1 public class Main {
2     public static void main(String[] args) {
3
4         System.out.println("Hola!!! ");
5
6         System.out.println("Hola!, ¿Como estas?");
7
8     }
9 }
10 }
```

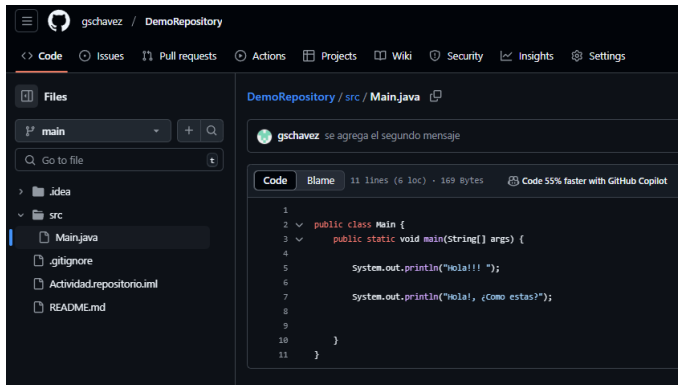
9) Hacemos pull request para que la rama nueva se una a el Master



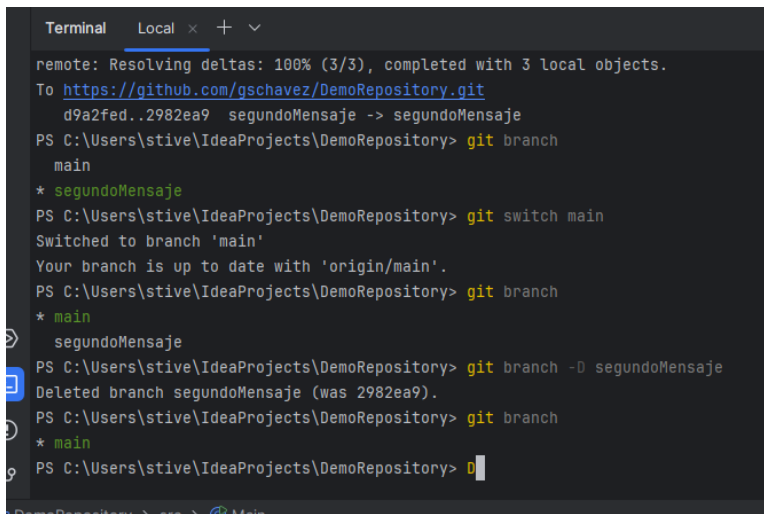
Borramos la rama del repositorio



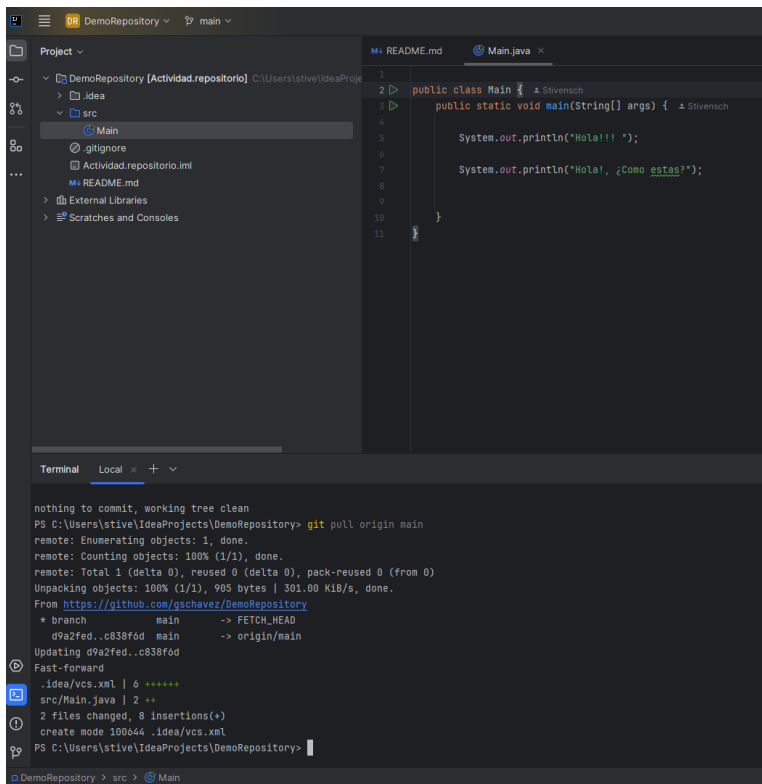
Comprobamos que la rama se haya agregado al master



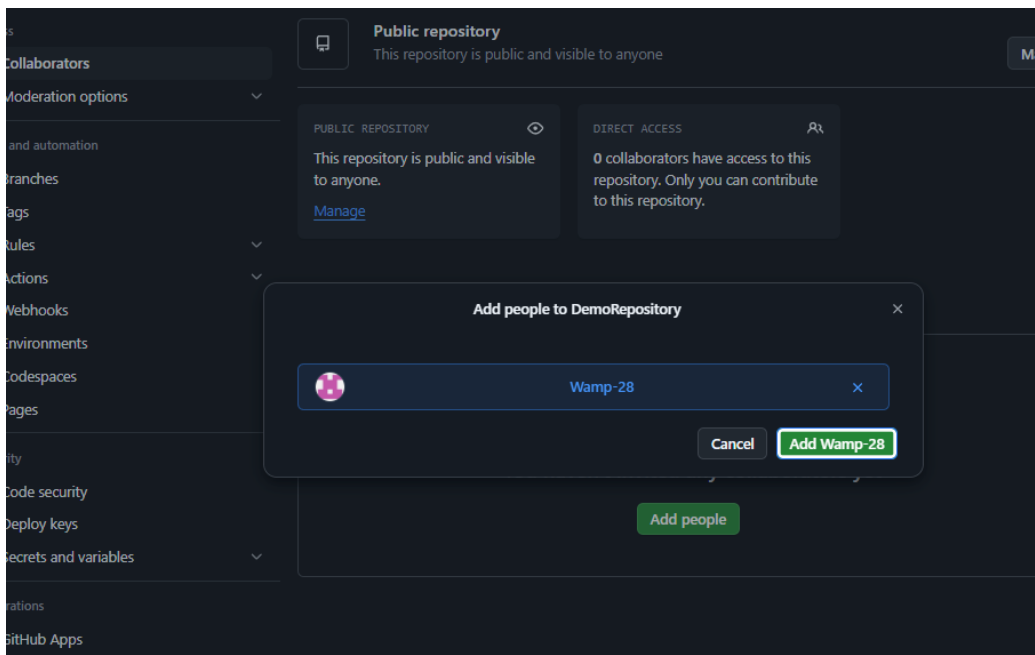
También borramos la rama del local con git branch -d



Bajamos el repositorio al local



10) Compartir el proyecto



11) Ahora con git log revisamos todos los commit realizados en orden cronológico

```

stive@StivenCh2906 MINGW64 ~/IdeaProjects/Actividad.repositorio (main)
$ git log
commit d9a2fed4d7b7f54d03784552e67724036b43b764 (HEAD -> main, origin/main)
Author: Stivensch <gschavez@ucundinamarca.edu.co>
Date:   Wed Feb 19 19:29:33 2025 -0500

    se agrega un primer mensaje

commit 3262a11d7d4bb2ea5502cf3233ca5e68a04c77a5
Author: Stivensch <gschavez@ucundinamarca.edu.co>
Date:   Wed Feb 19 19:27:11 2025 -0500

    first commit

stive@StivenCh2906 MINGW64 ~/IdeaProjects/Actividad.repositorio (main)
$

```

*Además con el git reflog podremos ver el historial de cambios locales y commits

```

stive@StivenCh2906 MINGW64 ~/IdeaProjects/Actividad.repositorio (main)
$ git reflog
d9a2fed (HEAD -> main, origin/main) HEAD@{0}: commit: se agrega un primer mensaje
3262a11 HEAD@{1}: Branch: renamed refs/heads/master to refs/heads/main
3262a11 HEAD@{3}: commit (initial): first commit

stive@StivenCh2906 MINGW64 ~/IdeaProjects/Actividad.repositorio (main)
$

```

*Con el git log --oneline se verá de forma mas compacta el historial, en una sola línea

```

stive@StivenCh2906 MINGW64 ~/IdeaProjects/Actividad.repositorio (main)
$ git log --oneline
d9a2fed (HEAD -> main, origin/main) se agrega un primer mensaje
3262a11 first commit

```

*Con el git Branch -r; se verán las ramas remotas

```

PS C:\Users\stive\IdeaProjects\DemoRepository> git branch -r
origin/HEAD -> origin/main
origin/main
origin/prueba
origin/segundoMensaje

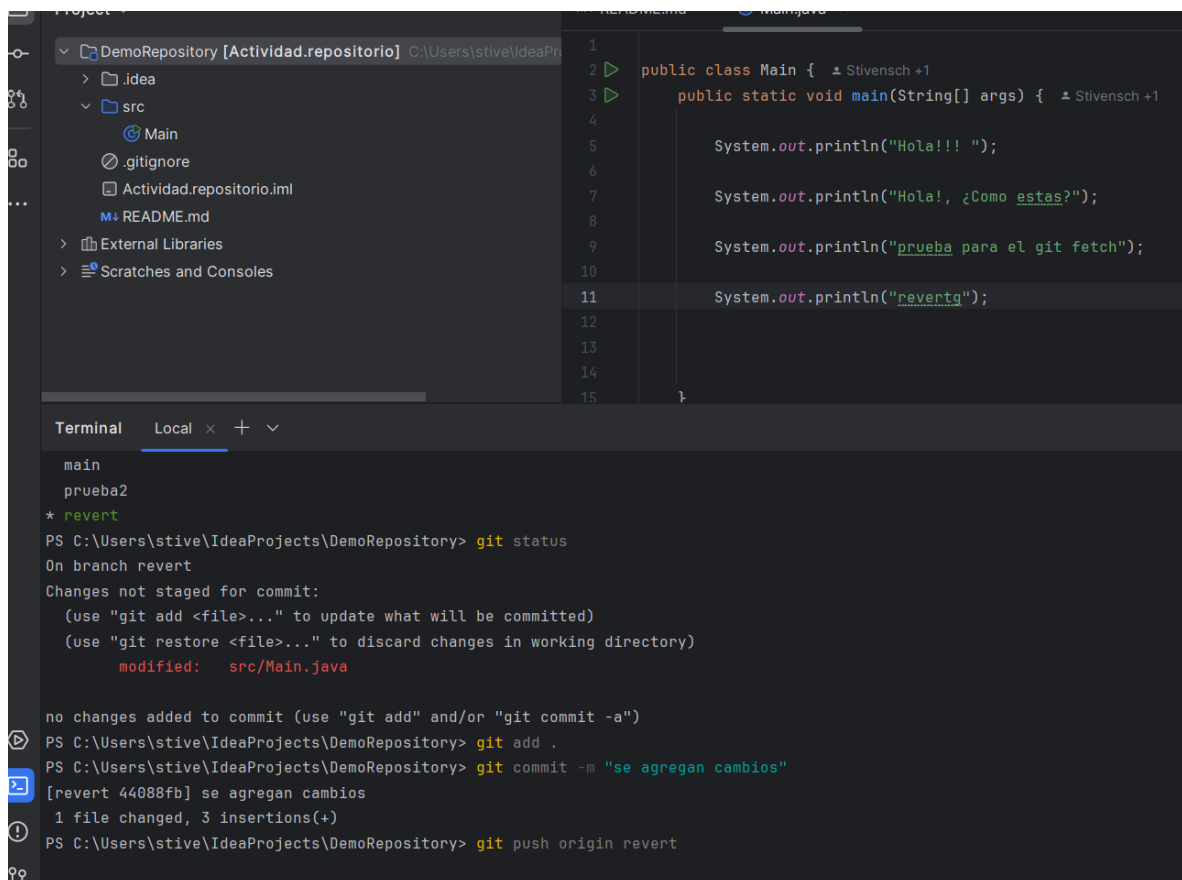
```

*Con el git fetch --all podremos actualizar las ramas remotas sin afectar el local

```
$ git fetch --all
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 2), reused 2 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (4/4), 356 bytes | 17.00 KiB/s, done.
From https://github.com/gschavez/DemoRepository
* [new branch] prueba2 -> origin/prueba2

stive@StivenCh2906 MINGW64 ~/IdeaProjects/Actividad.repositorio (main)
$
```

*Git revert; creamos una rama nueva con cambios y los subimos al repositorio remoto



```
1 public class Main {
2     private static final String STIVENSCH = "Stivensch";
3     public static void main(String[] args) {
4
5         System.out.println("Hola!!! ");
6
7         System.out.println("Hola!, ¿Como estas?");
8
9         System.out.println("prueba para el git fetch");
10
11        System.out.println("revert");
12    }
13 }
14
15 }

Terminal Local x + v
main
prueba2
* revert
PS C:\Users\stive\IdeaProjects\DemoRepository> git status
On branch revert
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:   src/Main.java

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\stive\IdeaProjects\DemoRepository> git add .
PS C:\Users\stive\IdeaProjects\DemoRepository> git commit -m "se agregan cambios"
[revert 44088fb] se agregan cambios
1 file changed, 3 insertions(+)
PS C:\Users\stive\IdeaProjects\DemoRepository> git push origin revert
```

Con ayuda de estos comandos revisamos la id del commit al que queremos hacerle el revert, y lo hacemos

*Git log --online*Git revert (retrocede cambios hechos en la rama local)

*Git merge para subir cambios desde consola

```
PS C:\Users\stive\IdeaProjects\DemoRepository> git merge mergeTest  
Already up to date.
```

Conclusión;

El lenguaje git es una herramienta muy útil para navegar y gestionar proyectos en github (o en otras plataformas) ayuda a optimizar el flujo y orden en cuanto a cambios o actualizaciones en un proyecto grande; es importante tener un buen manejo de este lenguaje pues es relevante en el ámbito profesional.