

IntelliJ y GitHub

Juan Cifuentes

Facultad de Ingeniería, Universidad de Cundinamarca

Programación II

Ing. Alexander M

19 de febrero del 2025

GitHub es una plataforma usada para el trabajo colaborativo en proyectos de software, facilita la gestión del código, el trabajo colaborativo y el seguimiento de los cambios a lo largo del proyecto

Por otra parte, IntelliJ IDEA es un IDE (entorno de desarrollo integrado) muy utilizado especialmente por usuarios del lenguaje Java el cual ofrece integración directa con git y GitHub

En este documento se hará una explicación paso a paso de cómo desarrollar un repositorio en GitHub y enlazarlo con un proyecto de IntelliJ IDEA

## Primer paso: tener Instalado Git, IntelliJ IDEA y JAVA en la versión 21.

Comprobaremos que versión de Git y Java tenemos con los siguientes comandos en el CMD

Git comando: `git --version`

```
C:\Users\Juan Diego>Git --version
git version 2.45.1.windows.1
```

Java comando: `java --version` (en mi caso uso la versión 22)

```
C:\Users\Juan Diego>java --version
java 22.0.2 2024-07-16
Java(TM) SE Runtime Environment (build 22.0.2+9-70)
Java HotSpot(TM) 64-Bit Server VM (build 22.0.2+9-70, mixed mode, sharing)
```

### 1.2 verificar la información de git con el código `Git config --list`

```
C:\Windows\System32>git config --list
diff.astextplain.textconv=astextplain
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
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
```

A continuación, se mostrará que significa cada configuración que se aprecia en la imagen

1. `diff.astextplain.textconv=astextplain`
  - Define una conversión de texto para diff, permitiendo que ciertos archivos sean tratados como texto al comparar diferencias.
2. `filter.lfs.clean=git-lfs clean -- %f`
  - Relacionado con Git LFS (Large File Storage). Este comando limpia los archivos antes de enviarlos al repositorio, optimizando el almacenamiento de archivos grandes.
3. `filter.lfs.smudge=git-lfs smudge -- %f`
  - También de Git LFS, este comando recupera archivos grandes cuando se extraen desde el repositorio, reemplazándolos por su contenido original.
4. `filter.lfs.process=git-lfs filter-process`
  - Define el proceso de filtrado para Git LFS, manejando la conversión entre versiones almacenadas y recuperadas de archivos grandes.
5. `filter.lfs.required=true`
  - Indica que Git LFS es obligatorio en este repositorio.
6. `http.sslbackend=openssl`
  - Especifica que Git usará OpenSSL como backend para manejar conexiones seguras (HTTPS).
7. `http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt`
  - Define la ubicación del archivo de certificados SSL usados por Git para verificar conexiones HTTPS seguras.
8. `core.autocrlf=true`
  - Maneja la conversión de saltos de línea (`\r\n` en Windows vs. `\n` en Unix).
  - `true`: Convierte automáticamente `\r\n` en `\n` al almacenar en el repositorio y lo revierte a `\r\n` al extraer en Windows.
9. `core.fscache=true`
  - Habilita el uso de un caché del sistema de archivos para mejorar el rendimiento de Git en Windows.
10. `core.symlinks=false`
  - Indica que los enlaces simbólicos no están habilitados en este sistema (por defecto en Windows).

11. pull.rebase=false

- Configura git pull para realizar una fusión (merge) en lugar de un rebase cuando se actualiza la rama local.

12. credential.helper=manager

- Usa el Git Credential Manager para almacenar credenciales de manera segura en el sistema.

13. credential.https://dev.azure.com.usehttppath=true

- Permite el uso de rutas HTTP completas para autenticar en Azure DevOps.

14. init.defaultbranch=master

- Define que la rama por defecto al inicializar un repositorio será master en lugar de main.

En esta lista de configuraciones hace falta la configuración de usuario, tanto el email como el nombre

Comando para agregar email: `config --global user.email "email"`

```
C:\Windows\System32>git config --global user.email jdiegocifuentes@ucundinamarca.edu.co
```

Comando para agregar el nombre: `config --global user.name "nombre"`

```
C:\Windows\System32>git config --global user.name JuanC
```

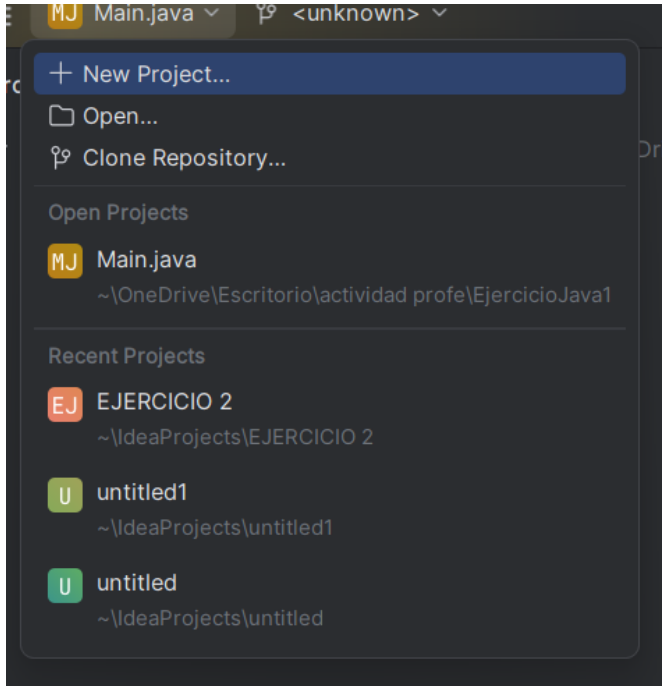
comprovamos con `Git config --list` (últimas 2 líneas)

```
C:\Windows\System32>git config --list
diff.astextplain.textconv=astextplain
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
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=jdiegocifuentes@ucundinamarca.edu.co
user.name=JuanC
```

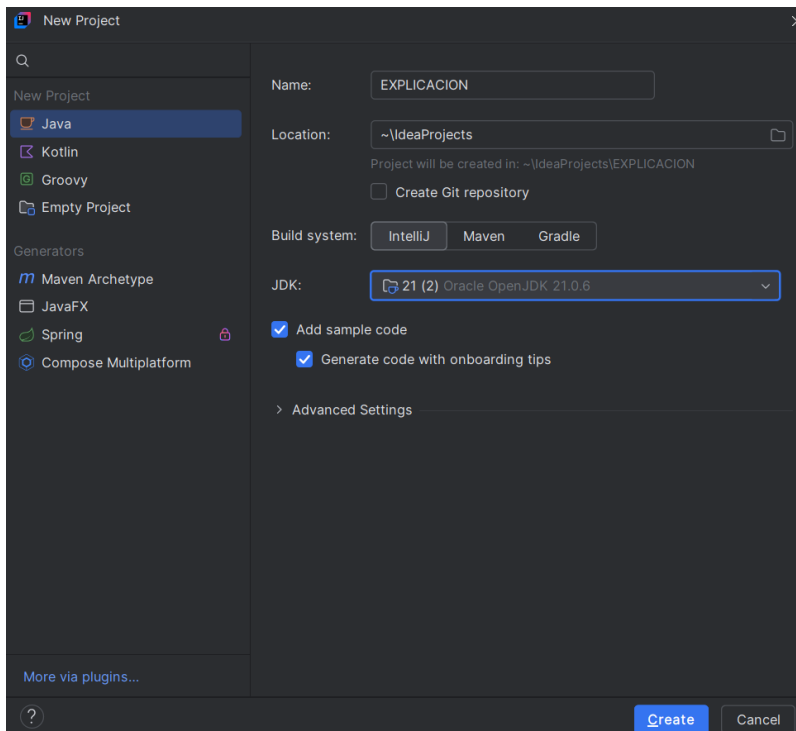
PARA ELIMINAR LOS USER. AL CODIGO LE AGREGAMOS UN --UNSET

**Segundo paso:** abrimos IntelliJ IDEA y creamos un proyecto

2.1 elegimos la opcion new project



2.2 le ponemos el nombre que deseamos, en mi caso EXPLICACION y seleccionamos en JDK la versión de nuestro java (versión 21) el resto de opciones lo dejamos como este predefinido



2.3 Una vez tengamos nuestro proyecto hecho, borraremos el código escrito por la aplicación (imagen1) para dar paso a nuestro código (imagen2)

Imagen1

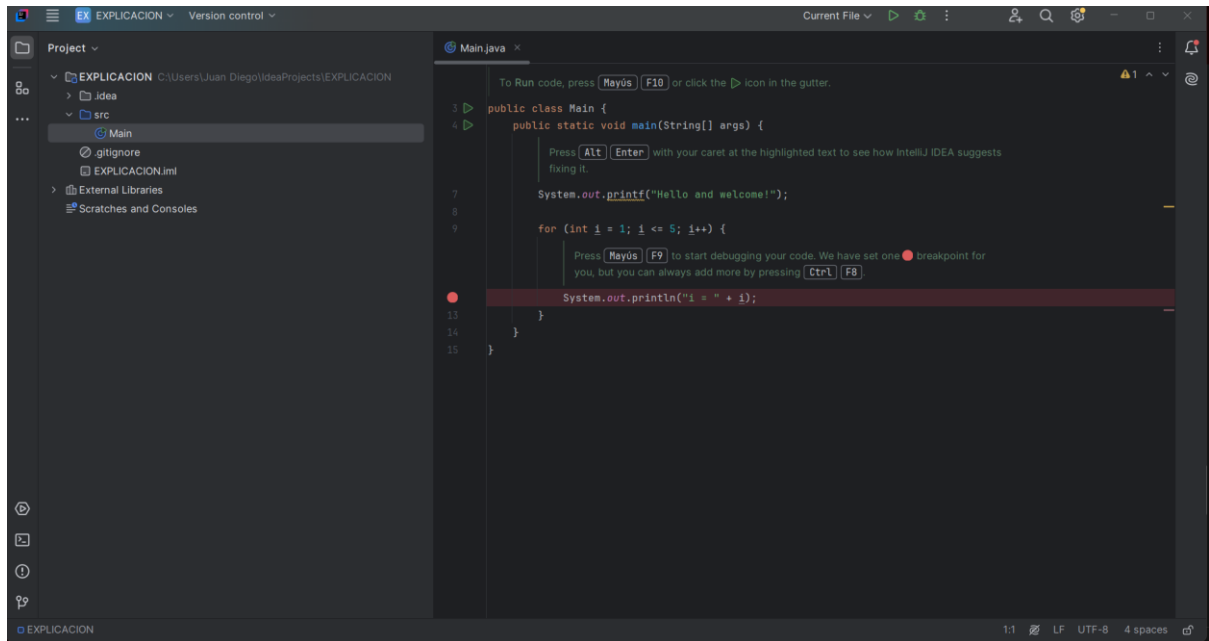
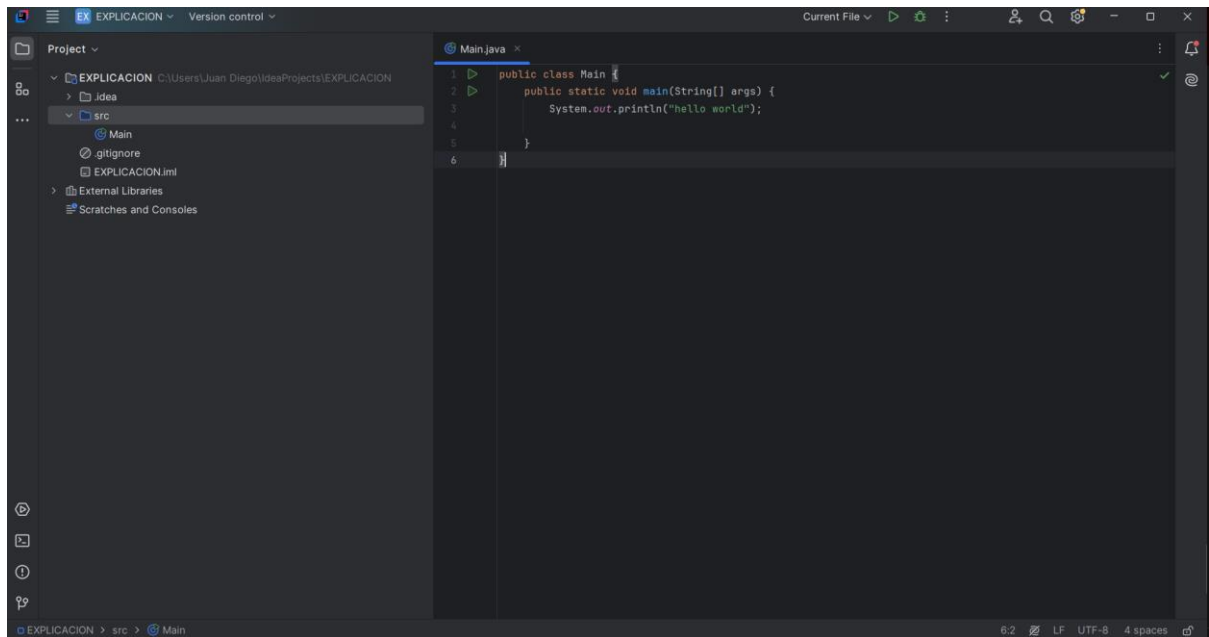
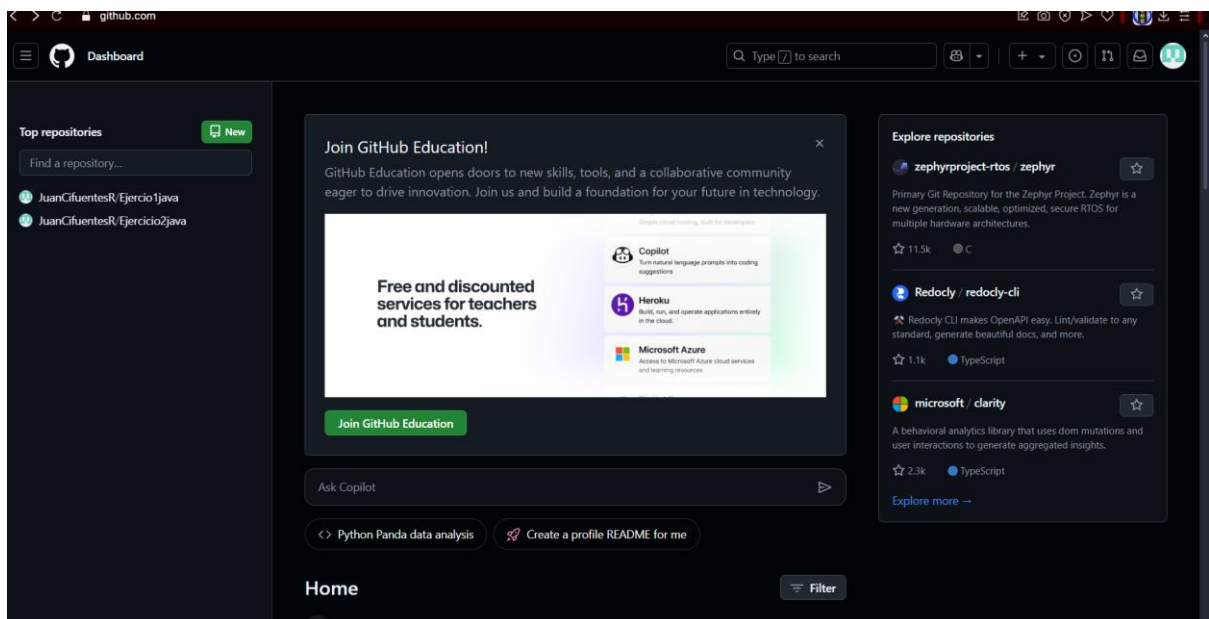


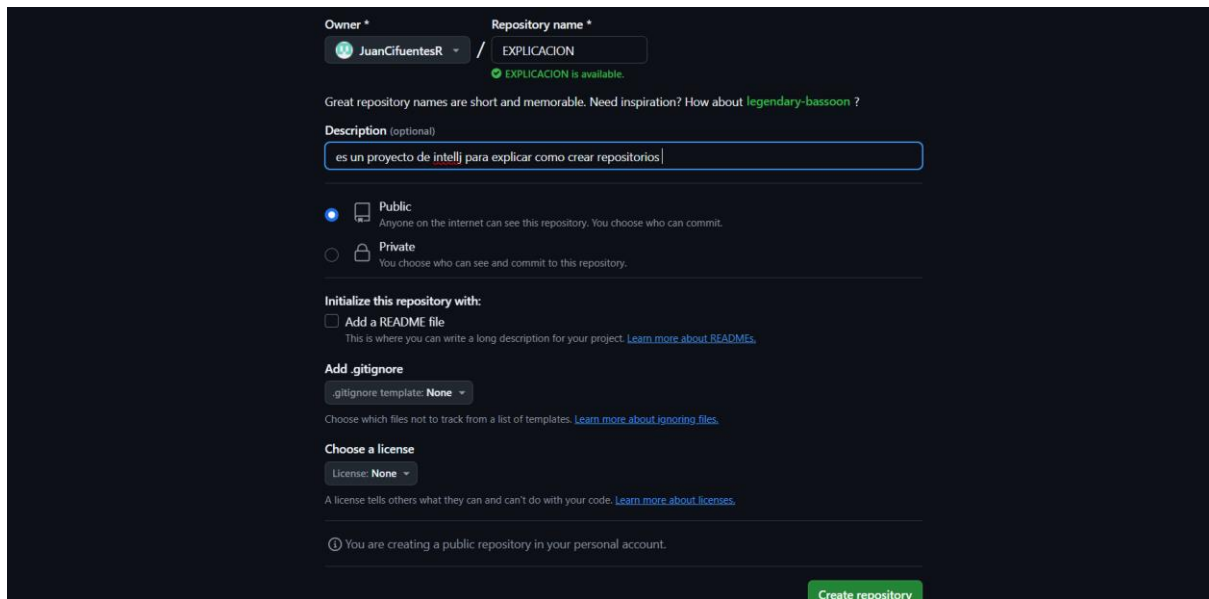
Imagen2



Tercer paso: abrimos GitHub.com en nuestro pc, en nuestra cuenta y buscamos el botón al lado izquierdo de la página que diga de NEW



3.1 ponemos el nombre que deseamos al repositorio y una breve descripción las demas opciones por el momento las dejamos como están y lo creamos



Owner \* JuanCifuentesR / Repository name \* EXPLICACION  
EXPLICACION is available.

Great repository names are short and memorable. Need inspiration? How about [legendary-bassoon](#)?

Description (optional)  
es un proyecto de intelli para explicar como crear repositorios

☒ 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

3.2 en la nueva pantalla que nos aparece (imagen1) copiamos y pegamos el primer código que nos aparece para luego copiarlo en la terminal (imagen2) de nuestro proyecto (imagen3)

imagen1

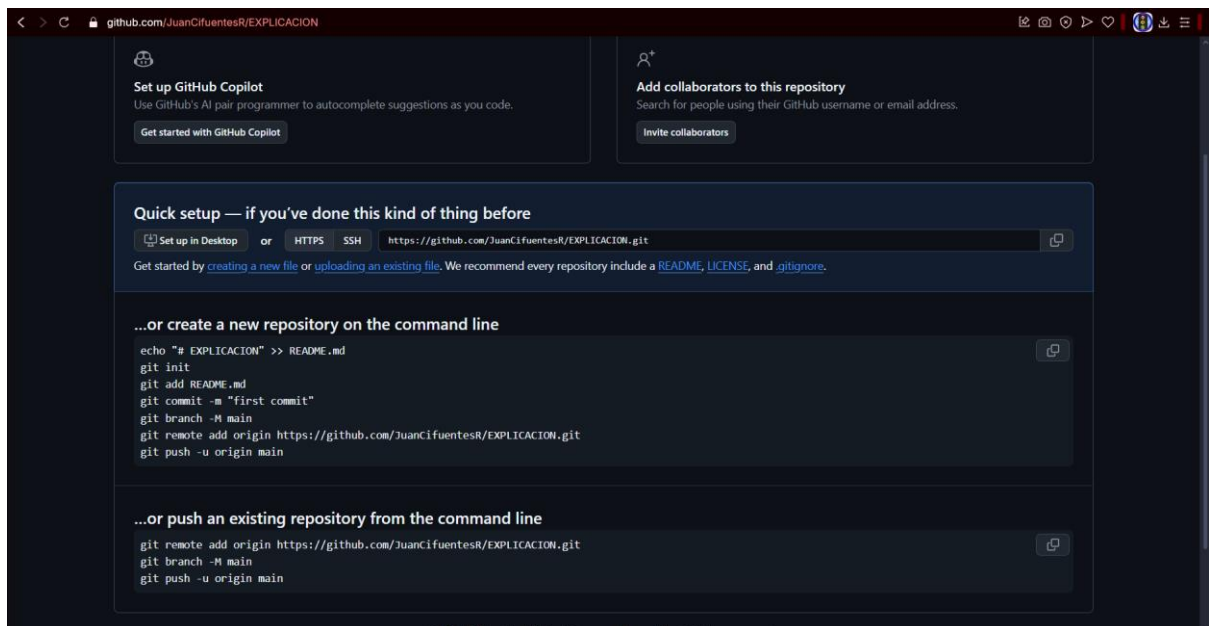


Imagen2



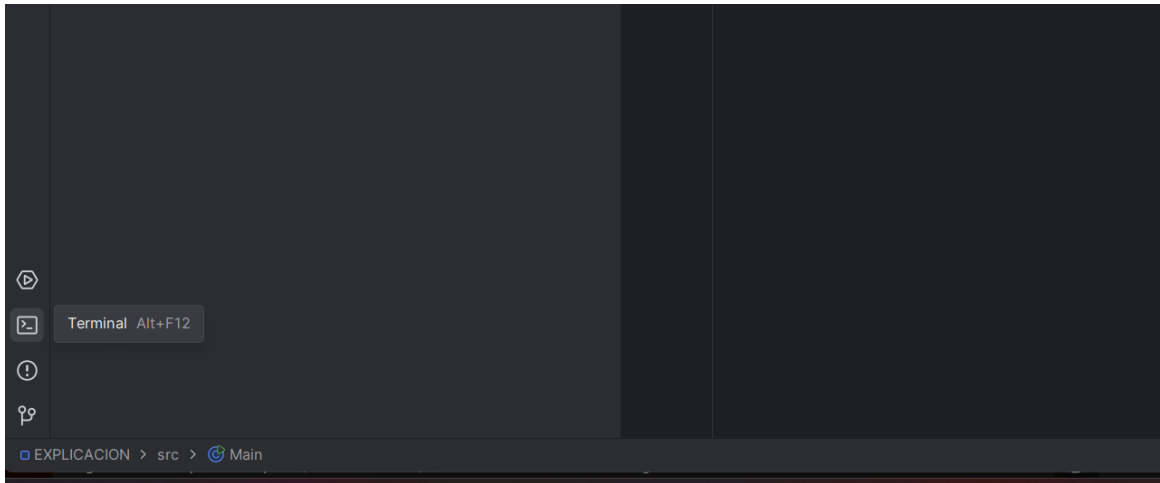
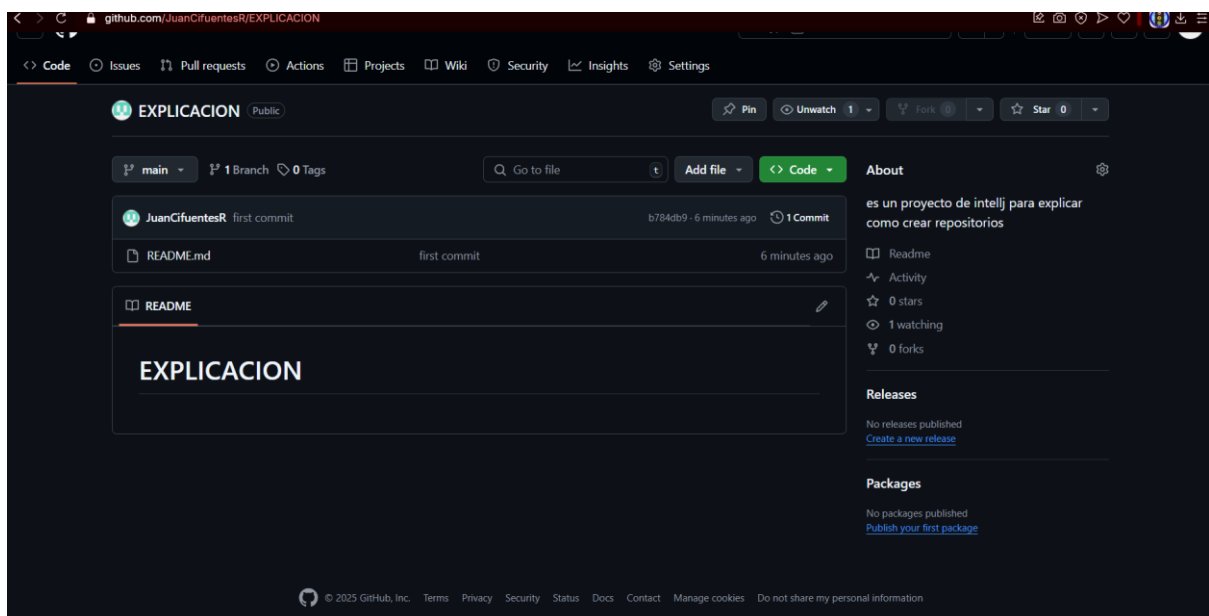


Imagen3

```
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git init
Initialized empty Git repository in C:/Users/Juan Diego/IdeaProjects/EXPLICACION/.git/
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git add README.md
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git commit -m "first commit"
[master (root-commit) b784db9] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git branch -M main
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git remote add origin https://github.com/JuanCifuentesR/EXPLICACION.git
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git push -u origin main
```

**Cuarto paso:** refrescamos la página y ya tenemos nuestro repositorio creado, a continuación, se mostrará como editar el proyecto de intelliJ y guardarlo en GitHub



4.1 en nuestro proyecto agregamos las lineas de codigo que necesitamos



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

4.2 en la terminal ponemos los siguientes codigos

Git status: nos sirve para ver cambios que no hemos guardado en git

```
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        .idea/
        EXPLICACION.iml
        src/

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION>
```

Git add .: nos sirve para “empaquetar” todo lo que no hemos guardado

Git commit -m “texto”: siver para poner comentarios de la novedad que hicimos en el codigo

```

PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git add .
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git commit -m "agrego el dia de la semana"
[main 01c49ec] agrego el dia de la semana
 7 files changed, 71 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 .idea/.gitignore
 create mode 100644 .idea/misc.xml
 create mode 100644 .idea/modules.xml
 create mode 100644 .idea/vcs.xml
 create mode 100644 EXPLICACION.iml
 create mode 100644 src/Main.java

```

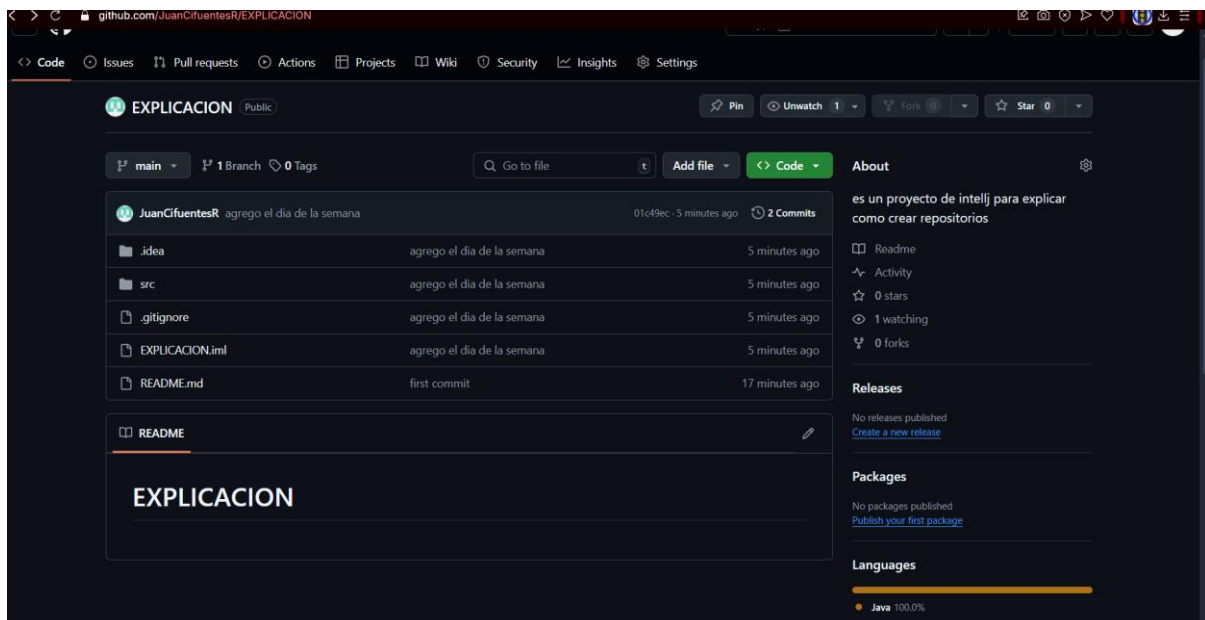
Git push: nos sirve para subir los cambios a GitHub

```

PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git push origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (11/11), 1.71 KiB | 439.00 KiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/JuanCifuentesR/EXPLICACION.git
   b784db9..01c49ec  main -> main
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION>

```

4.3 recargar la página y ver los cambios (si se quiere ver el código abres src luego main y veras el código)



The screenshot shows the GitHub repository page for 'EXPLICACION' by user 'JuanCifuentesR'. The repository is public and has 0 stars and 0 forks. The commit history shows a single commit '01c49ec' from 5 minutes ago, titled 'agrego el dia de la semana'. The file list includes '.idea', 'src', '.gitignore', 'EXPLICACION.iml', and 'README.md'. The README content is visible, showing the title 'EXPLICACION'.

Quinto paso: ramas (administración y creación)

Como dice en la plataforma de GitHub las ramas son “Las ramas te permiten desarrollar características, corregir errores, o experimentar con seguridad las ideas nuevas en un área contenida de tu repositorio. Siempre puedes crear una rama a partir de otra rama existente. Habitualmente, puedes crear una rama nueva desde la rama predeterminada de tu repositorio.” (GitHub .s.f)

Ahora veremos cómo se crean y se usan

### 5.1 desplegar la opción main para ver las ramas



### 5.2 en la terminal de IntelliJ ingresar los siguientes códigos para ver, crear, y manipular las ramas

Git branch: ver las ramas

```
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git branch
* main
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION>
```

Git switch -c "nombre de la rama": sirve para crear una rama

```
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git switch -c "JuanCifuentes"
Switched to a new branch 'JuanCifuentes'
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION>
```

Git switch "nombre de la rama": sirve para cambiar de rama

(estamos en la rama creada recientemente)

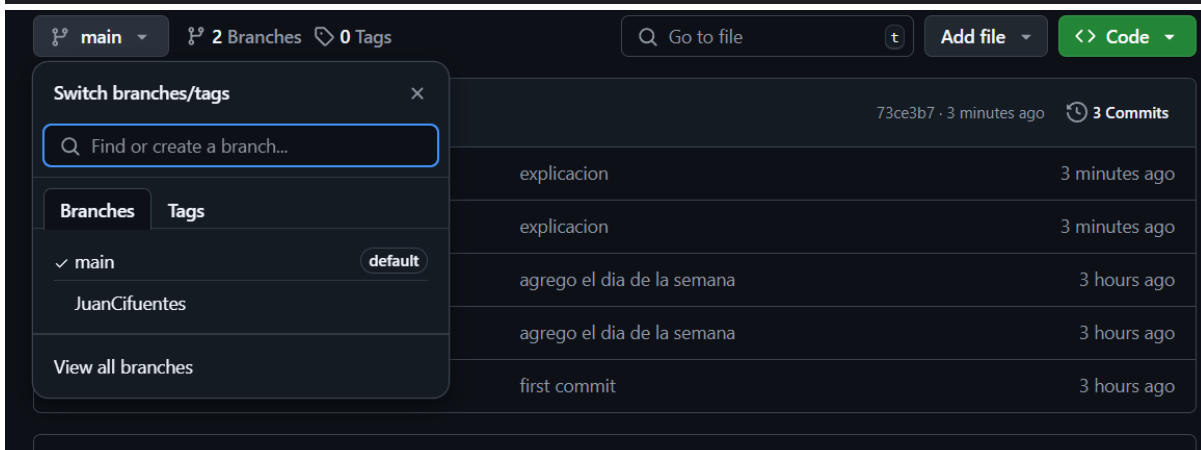
```
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git branch
* JuanCifuentes
  main
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION>
```

(usamos el código para cambiar a main y verificamos con git branch)

```
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git switch "main"
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git branch
  JuanCifuentes
* main
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION>
```

Git push origin “nombre de la rama”: sirve para poder subir a GitHub nuestra nueva rama la cual aparecera en el apartado main

```
PS C:\Users\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION> git push origin JuanCifuentes
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'JuanCifuentes' on GitHub by visiting:
remote:   https://github.com/JuanCifuentesR/EXPLICACION/pull/new/JuanCifuentes
remote:
To https://github.com/JuanCifuentesR/EXPLICACION
 * [new branch]      JuanCifuentes -> JuanCifuentes
PS C:\Users\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION>
```



The screenshot shows the GitHub web interface for the repository 'JuanCifuentesR/EXPLICACION'. The 'Switch branches/tags' dropdown is open, showing 'main' as the current branch and 'JuanCifuentes' as a new branch. The main content area displays a list of commits, including 'explicacion' and 'agrego el dia de la semana'.

Branch	Commit	Time
main	73ce3b7	3 minutes ago
JuanCifuentes	explicacion	3 minutes ago
JuanCifuentes	explicacion	3 minutes ago
JuanCifuentes	agrego el dia de la semana	3 hours ago
JuanCifuentes	agrego el dia de la semana	3 hours ago
JuanCifuentes	first commit	3 hours ago

Git fetch --all: su función es “descargar” las ramas remotas del repositorio en GitHub y  
 Git branch --r: funciona para mostrar las ramas remotas

```

PS C:\Users\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION> git fetch --all
PS C:\Users\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION> git branch -r
      origin/HEAD -> origin/main
      origin/main
PS C:\Users\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION>

```

## Sexto paso: clonación de proyectos

Como menciona la plataforma de GitHub “Puedes clonar un repositorio de GitHub.com en el equipo local o en un codespace, para que sea más fácil corregir conflictos de combinación, agregar o quitar archivos e insertar confirmaciones más grandes” (GitHub s.f)

6.1 copiamos la url del repositorio que se desea clonar, en mi caso, mi mismo repositorio, para luego crear una carpeta vacía y allí dar clic derecho -> mostrar más opciones -> open git brash here y en la consola que se abre ponemos el siguiente codigo

Git clone enlace del repositorio

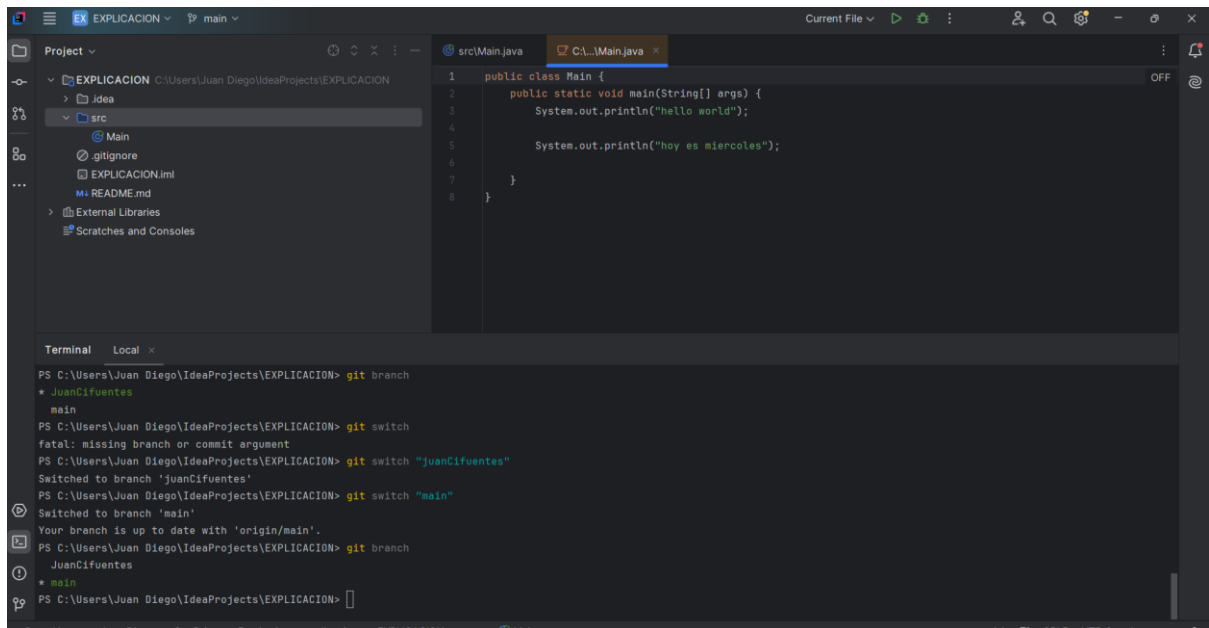
```

Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/explicacion
$ git clone https://github.com/JuanCifuentesR/EXPLICACION
Cloning into 'EXPLICACION'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 14 (delta 0), reused 14 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (14/14), done.

```

6.2 revisar la carpeta nuevamente y se habrá clonado el repositorio, abrir la carpeta -> src -> main . De esta manera se nos abraira el proyecto en IntelliJ IDEA

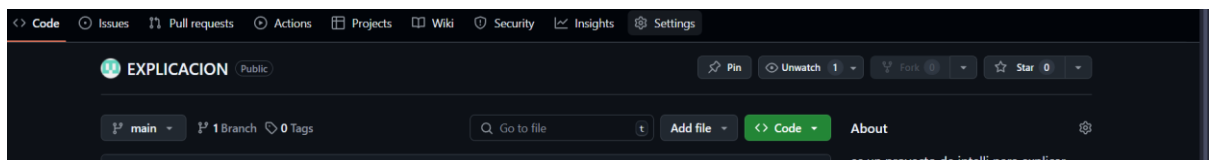
(en mi caso me aparece 2 veces ya que es mi mismo proyecto y repositorio)



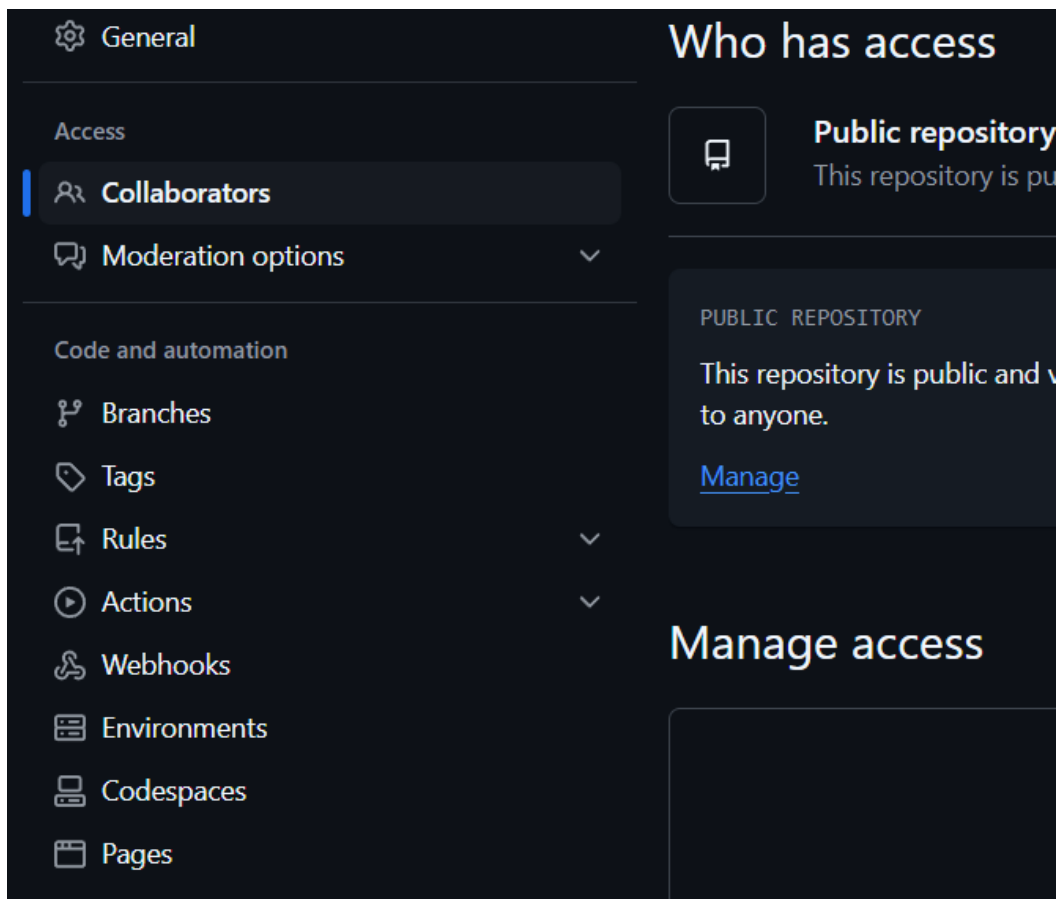
## Séptimo paso: agregar colaboradores al repositorio

Agregar colaboradores al repositorio nos ayuda a crear proyectos colaborativos, así revisar códigos y dar aprobación y/o correcciones a ramas de los colaboradores para agregarlas a la rama principal “main”

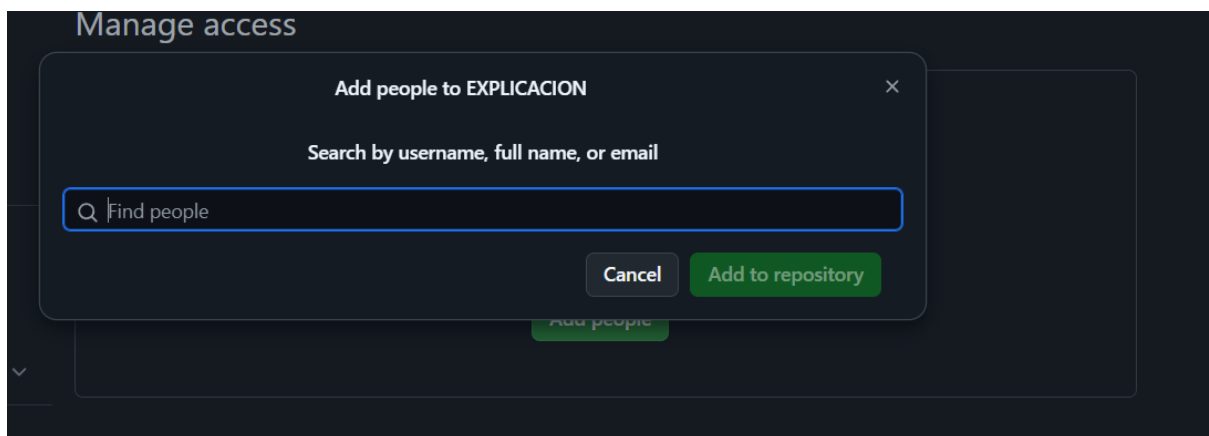
### 7.1 ingresar a settings en la parte superior



### 7.2 agregar colaboradores en la parte izquierda



7.3 clic a “Add people” e ingresar el nombre de usuario de la persona que desee agregar como colaborador



**Octavo paso:** subir la rama a GitHub

8.1 agregar código al proyecto ESTANDO EN LA NUEVA RAMA y usamos los códigos de las ramas anteriormente vistos para guardar los cambios



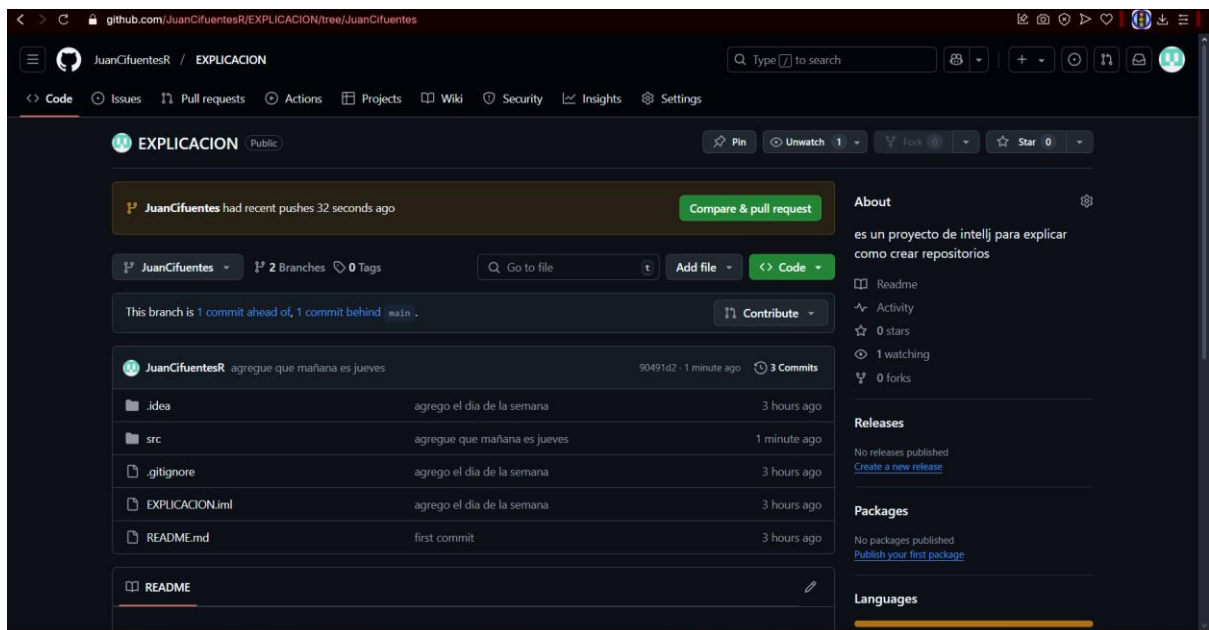
The screenshot shows an IDE with a project named 'EXPLICACION'. The 'Main.java' file is open, showing a Java class with a 'main' method. The terminal window displays the following commands and output:

```
PS C:\Users\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION> 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\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION> git add .
PS C:\Users\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION> git commit -m "agregue que mañana sera jueves"
[main a8c99c2] agregue que mañana sera jueves
1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION> git push origin JuanCifuentes
Everything up-to-date
PS C:\Users\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION>
```

8.2 verificar que los cambios estén en GitHub y hacer el pull request



8.3 clic en create pull request para confirmar

Markdown is supported Paste, drop, or click to add files

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

1 commit 1 file changed At 1 contributor

Commits on Feb 20, 2025

agregue que mañana es jueves  
JuanCifuentesR committed 2 minutes ago

Showing 1 changed file with 1 addition and 0 deletions.

Split Unified

```
src/Main.java
@@ -3,6 +3,7 @@ public static void main(String[] args) {
3      System.out.println("hello world");
4
5      System.out.println("hoy es miercoles");
6 +    System.out.println("mañana es jueves");
7
8 }
```

Open JuanCifuentesR wants to merge 1 commit into main from JuanCifuentes

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -0

JuanCifuentesR commented 1 minute ago  
No description provided.

agregue que mañana es jueves 90491d2

JuanCifuentesR commented now  
ok

No conflicts with base branch  
Merging can be performed automatically.

Merge pull request You can also merge this with the command line. [View command line instructions](#).

Preview Switch back to the classic merge experience Give feedback

Add a comment

Write Preview

Reviewers  
No reviews  
Still in progress? [Convert to draft](#)

Assignees  
No one—assign yourself

Labels  
None yet

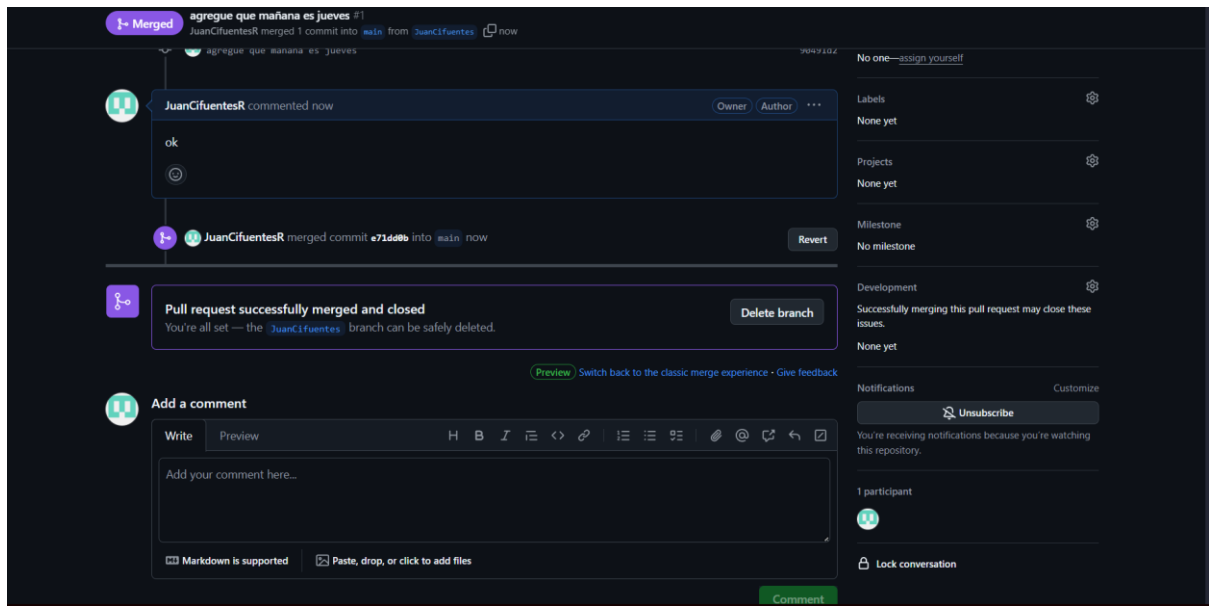
Projects  
None yet

Milestone  
No milestone

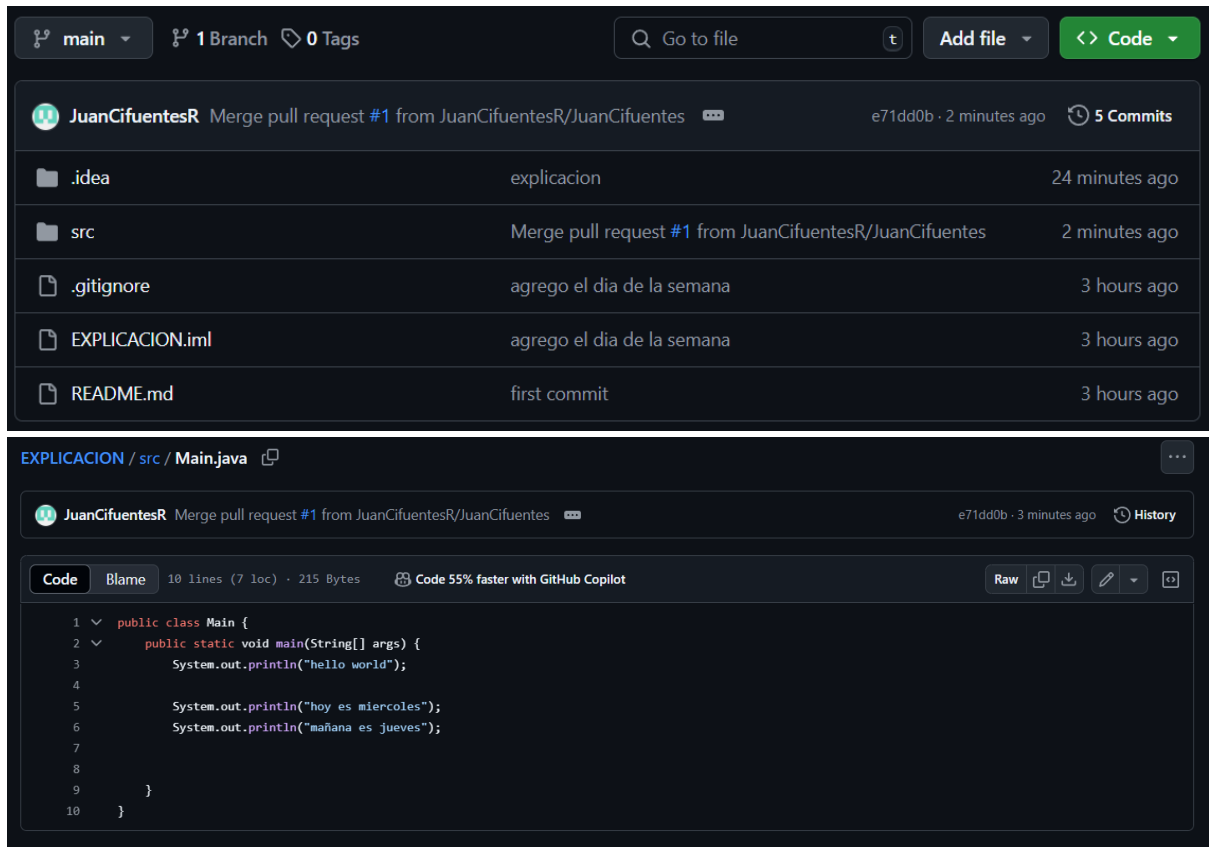
Development  
Successfully merging this pull request may close these issues.  
None yet

Notifications  
Unsubscribe

8.3 borrar la rama donde dice “Delete branch”



8.4 verificar que los cambios de la rama se hayan aplicado en la rama principal “main”



Noveno paso: eliminar la rama en el ordenador

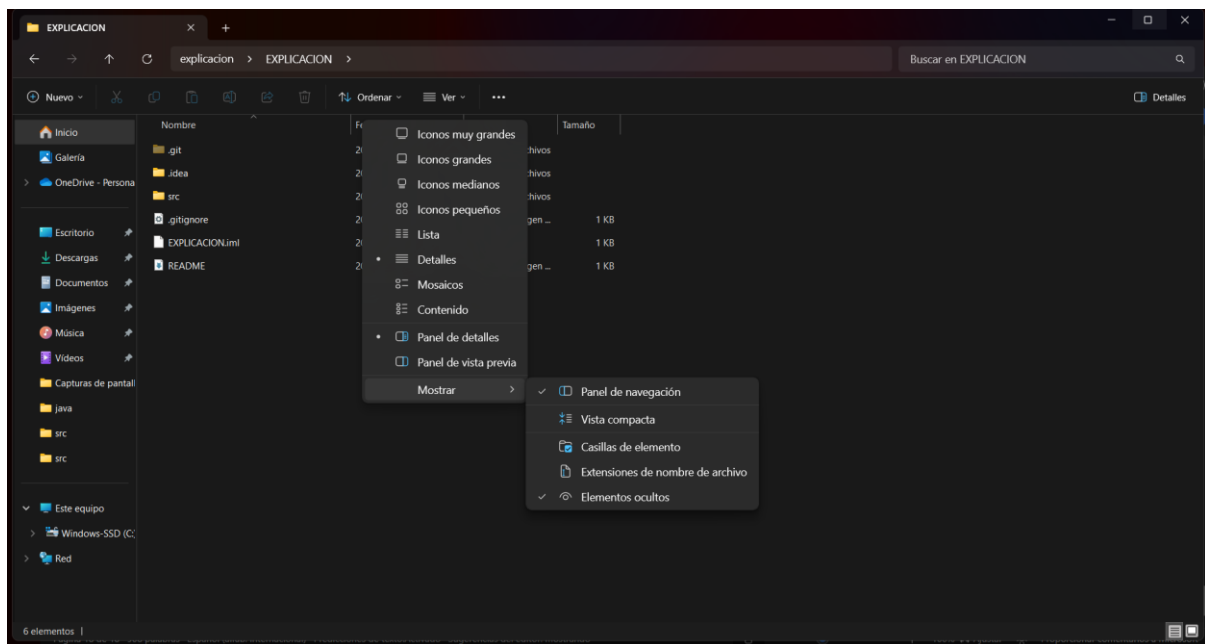
IntelliJ IDEA

Con el código `git branch -d "nombre de la rama"` puedes eliminar las ramas

```
PS C:\Users\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION> git branch -d JuanCifuentes
Deleted branch JuanCifuentes (was 90491d2).
PS C:\Users\Juan Diego\OneDrive\Escritorio\explicacion\EXPLICACION>
```

Proyecto clonado

Ver -> Mostrar -> Elementos ocultos y eliminamos la carpeta `.git`



En conclusión, con los anteriores pasos será capaz de relizar proyectos en IntelliJ IDEA y vincularlos en GitHub en un repositorio, junto con esto podrá realizar cambios en el código y subirlo a la plataforma mediante códigos Git en la Terminal de IntelliJ IDEA, tambien podra entender y manipular el concepto de “ramas” haciendo efectivo el trabajo colaborativo sin poner en riesgo la rama principal “main” ya que permite un sistema de colaboracion de todos los participantes para poder hacer merge de la rama de trabajo y la principal

**Decimo paso:** hacer reverse a los commit

10.1 créar una carpeta -> abrimos el git bash -> enlazamos con el repositorio -> creamos archivos de prueba con el siguiente comando

Touch “nombre del archivo.tipo del archivo” y verificamr con un `git status`

```
Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ touch mi_archivo.txt

Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

10.2 guardar cambio con los comandos anteriores -> git add . -> git commit -m "mensaje"  
-> git push origin

```
Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ touch mi_archivo2.txt

Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        mi_archivo.txt
        mi_archivo2.txt

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

Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ git add .

Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ gitt commit -c "agregue 2 archivos de texto"
bash: gitt: command not found

Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ git commit -c "agregue 2 archivos de texto"
fatal: could not lookup commit 'agregue 2 archivos de texto'

Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ git commit -m "agregue 2 archivos de texto"
[main 40ff039] agregue 2 archivos de texto
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 mi_archivo.txt
 create mode 100644 mi_archivo2.txt

Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ git push origin
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
```

10.3 hacer esto las veces que desee para luego hacer el git revert de la siguiente manera

Git relog: para ver las referencias del log y alli encontrar el ID del commit que deseamos borrar

```
Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ git relog
96b83ed (HEAD -> main, origin/main) HEAD@{0}: commit: agregue archivo 3-4
40ff039 HEAD@{1}: commit: agregue 2 archivos de texto
a86adb HEAD@{2}: Branch: renamed refs/heads/master to refs/heads/main
a86adb HEAD@{4}: commit (initial): first commit
```

## Git revert "ID del commit" y subir cambios con git push origin

```
Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ git revert 96b83ed
[main 3e8b058] Revertit git add README.md git commit -m "first commit" git b
chivo 3-4"
2 files changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 mi_archivo3.txt
delete mode 100644 mi_archivo4.txt

Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/github (main)
$ git push origin
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% (2/2), 349 bytes | 349.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/JuanCifuentesR/explicacion-reverse.git
96b83ed..3e8b058 main -> main
```

(al hacer el revert aparecera la siguiente ventana, donde se debera seguir los siguientes pasos)

[illegible]

Presionar esc -> escribir :q o :wq si se escribió algún comentario en el revert y enter para salir y poder guardar cambios

## Undecimo paso: asociar un archivo a un repositorio existente

Abirmos git bash en la carpeta donde tenemos el archivo y ponemos los siguientes codigos

Git init (para iniciar git en la carpeta)

```
Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/trabajo par subir
$ git init
Initialized empty Git repository in C:/Users/Juan Diego/OneDrive/Escritorio/trab
ajo par subir/.git/
```

Git remote add origin “enlace del repositorio” (para vincular al repositorio existente)

```
Juan Diego@Papupro MINGW64 ~/OneDrive/Escritorio/trabajo par subir (master)
$ git remote add origin https://github.com/JuanCifuentesR/explicacion-reverse
```

Por último hacemos el procedimiento debido para subir el archivo al repositorio

Git add . -> git commit-m -> git push origin

## REFERENCIAS

GitHub. (2025). *Acerca de GitHub y Git*. GitHub Docs. <https://docs.github.com/es/get-started/start-your-journey/about-github-and-git>

GitHub. (2025). *Clonar un repositorio*. GitHub Docs. <https://docs.github.com/es/repositories/creating-and-managing-repositories/cloning-a-repository>

GitHub. (2025). *Acerca de las ramas*. GitHub Docs. <https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches>

JetBrains. (s.f.). *Funciones de IntelliJ IDEA*. JetBrains. <https://www.jetbrains.com/es-es/idea/features/>

