

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`

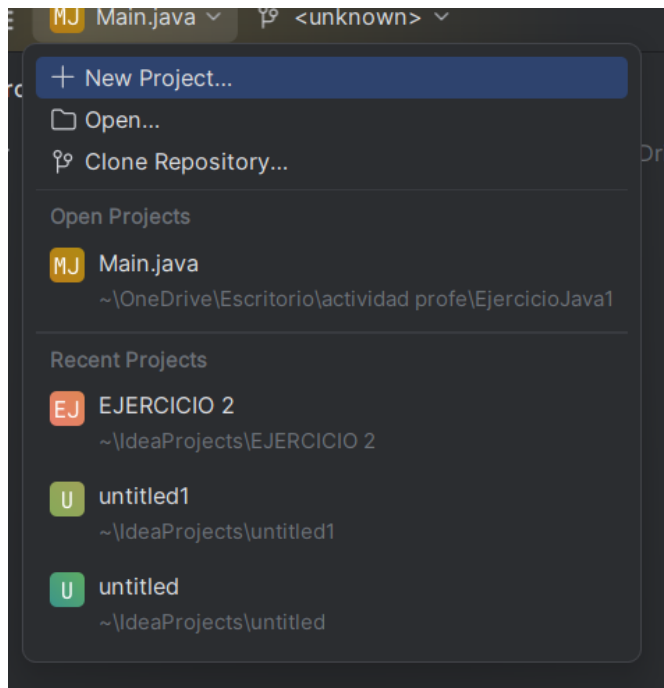
```
C:\Users\Juan Diego>Git --version
git version 2.45.1.windows.1
```

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

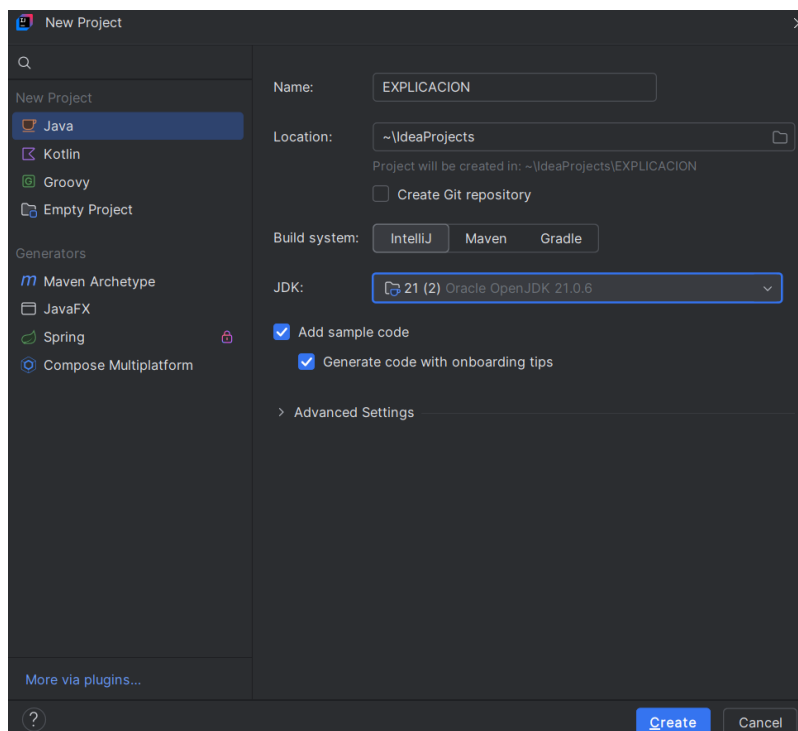
```
C:\Users\Juan Diego>java --version
java 22.0.2 2024-07-16
Java(TM) SE Runtime Environment (build 22.0.2+9-70)
Java HotSpot(TM) 64-Bit Server VM (build 22.0.2+9-70, mixed mode, sharing)
```

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

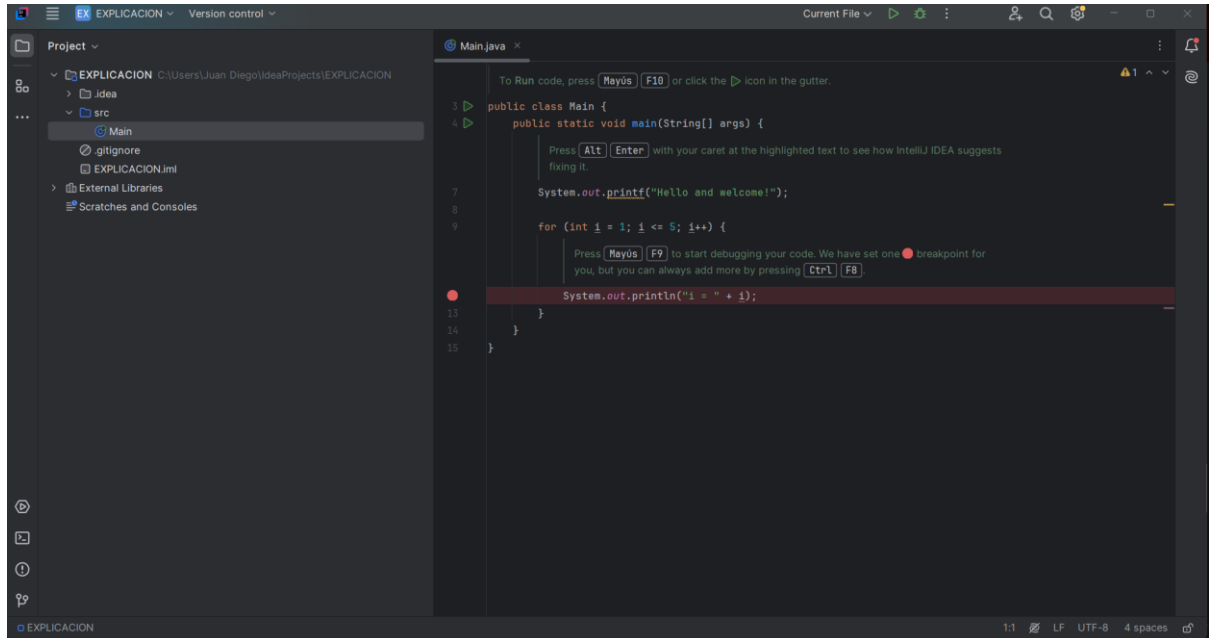
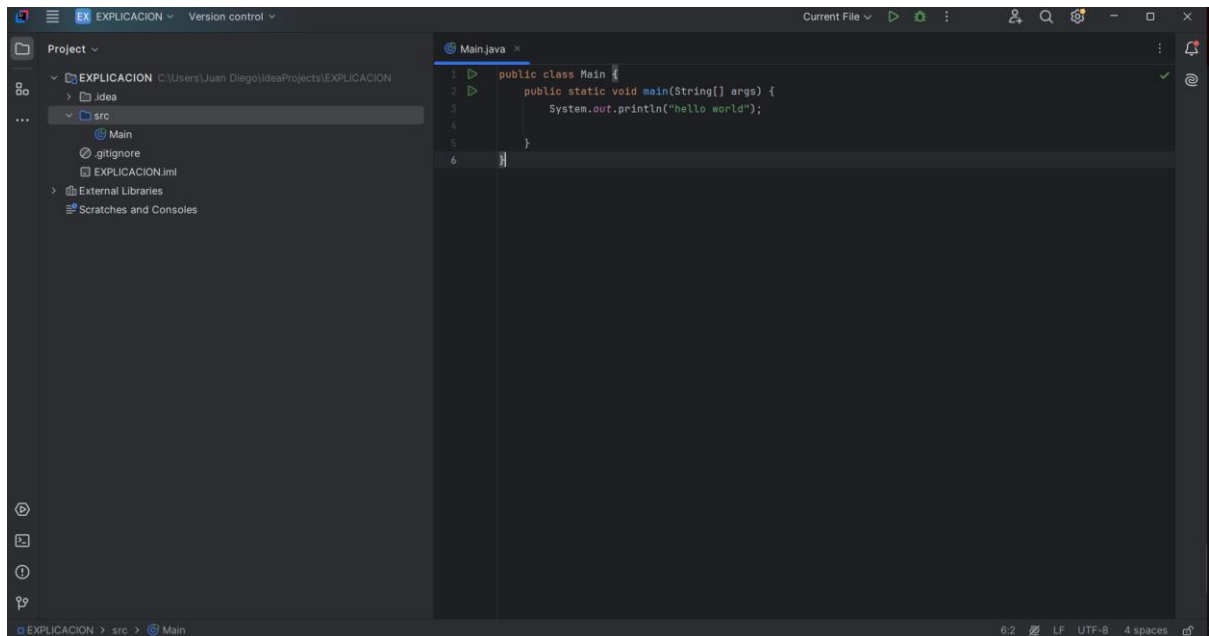
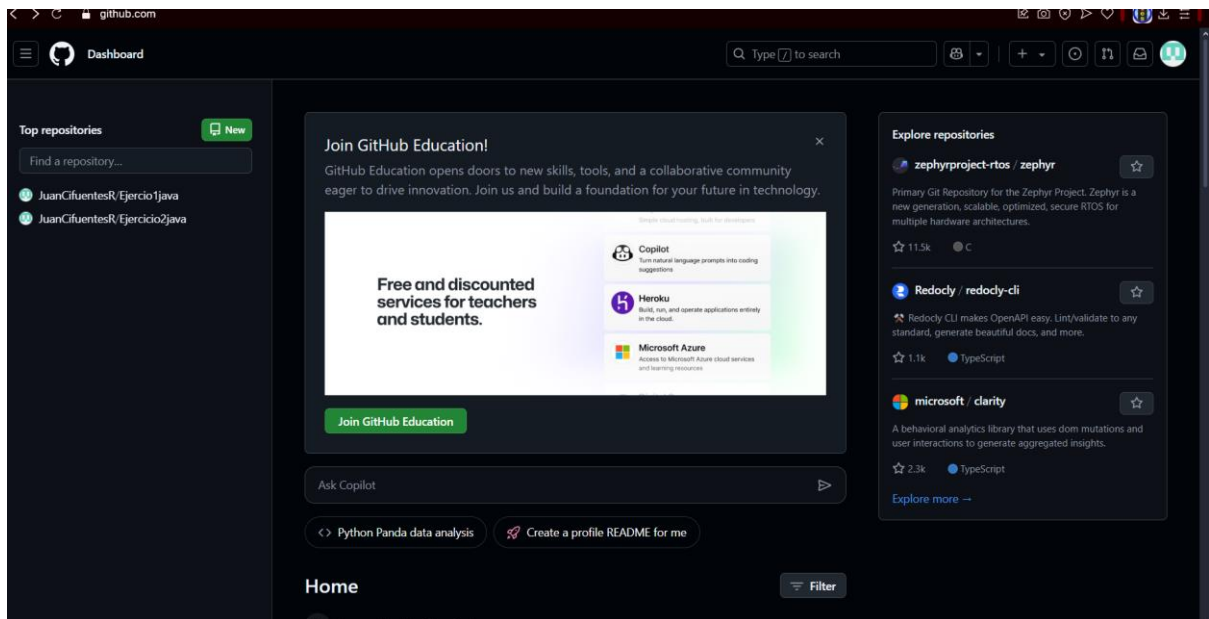


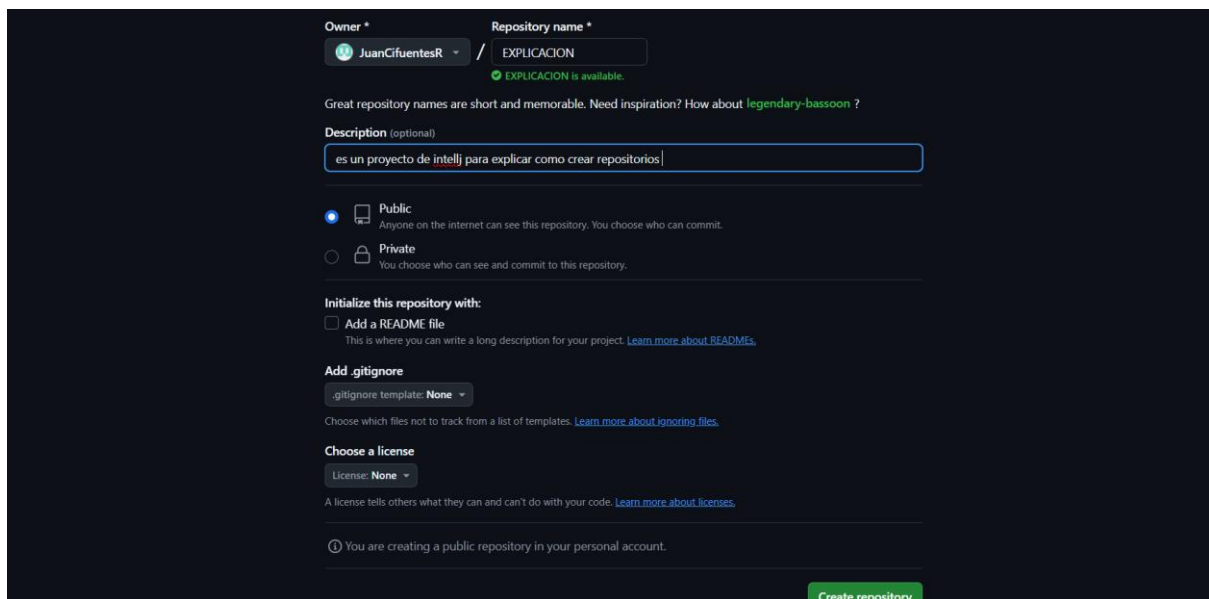
Imagen2



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



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

imagen1

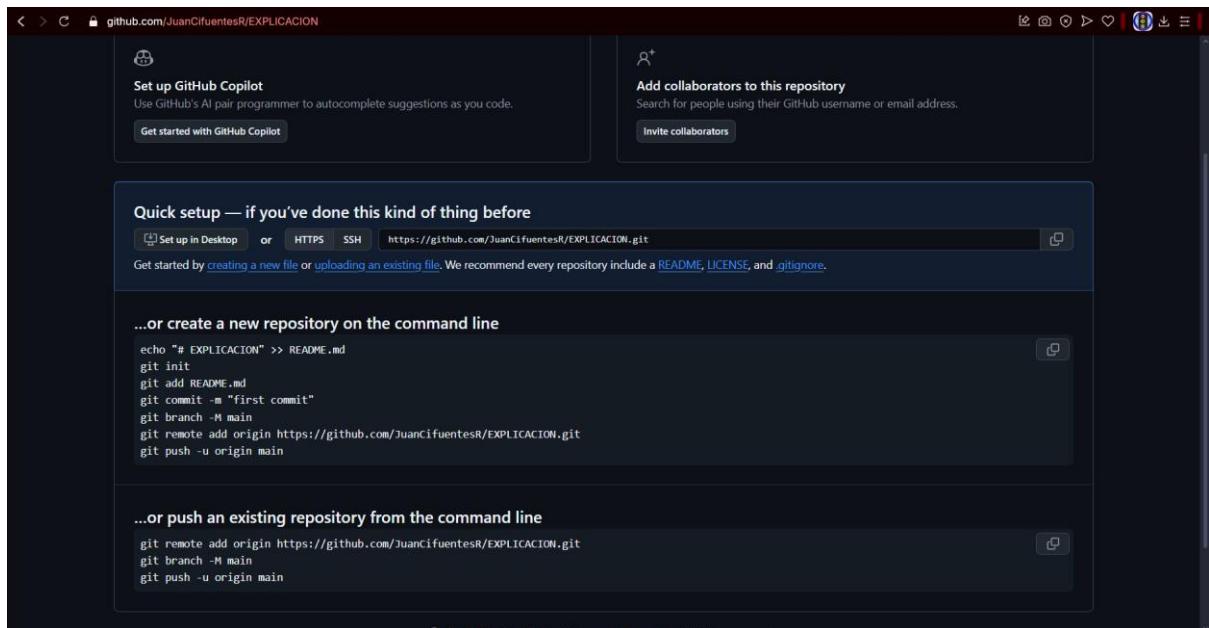


Imagen2

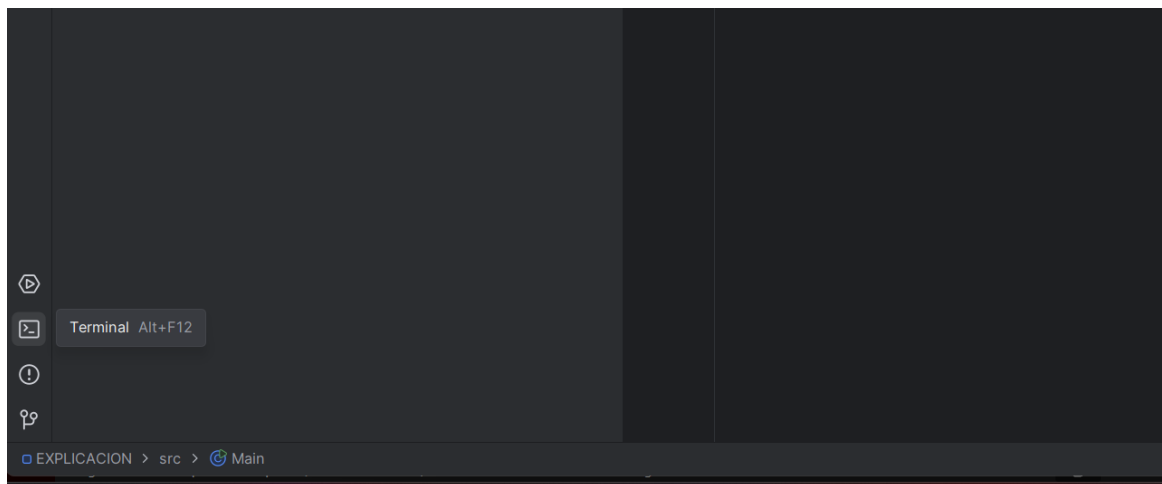
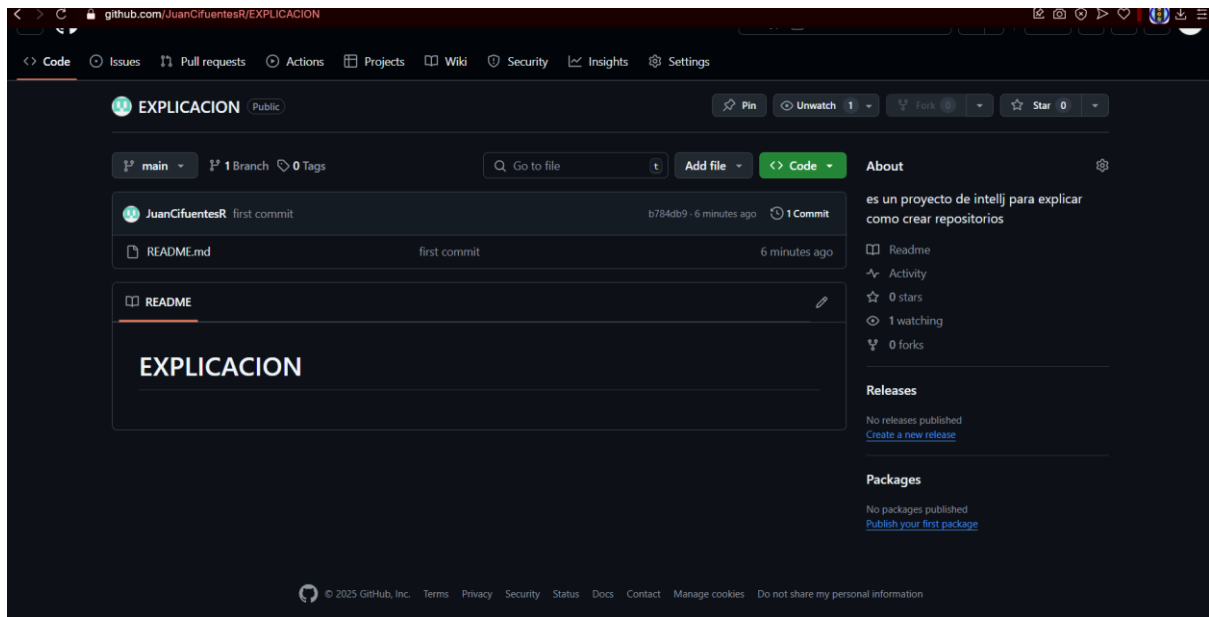


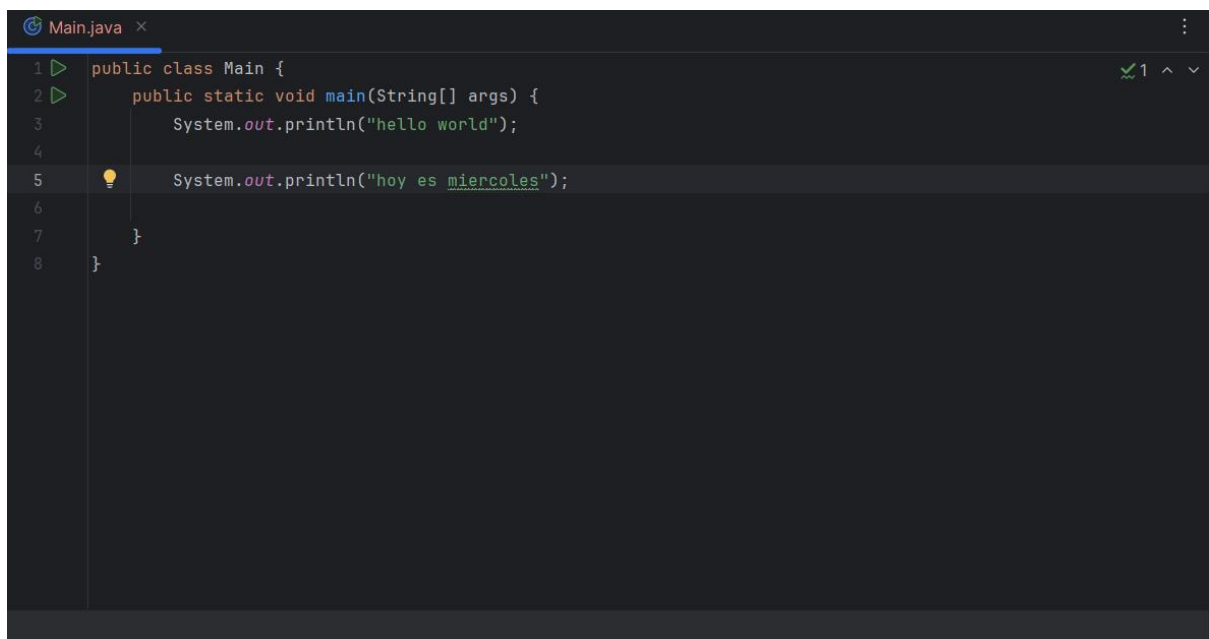
Imagen3



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



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”: sirve para poner comentarios de la novedad que hicimos en el código

```

PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git add .
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git commit -m "agrego el día de la semana"
[main 01c49ec] agrego el día 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

```

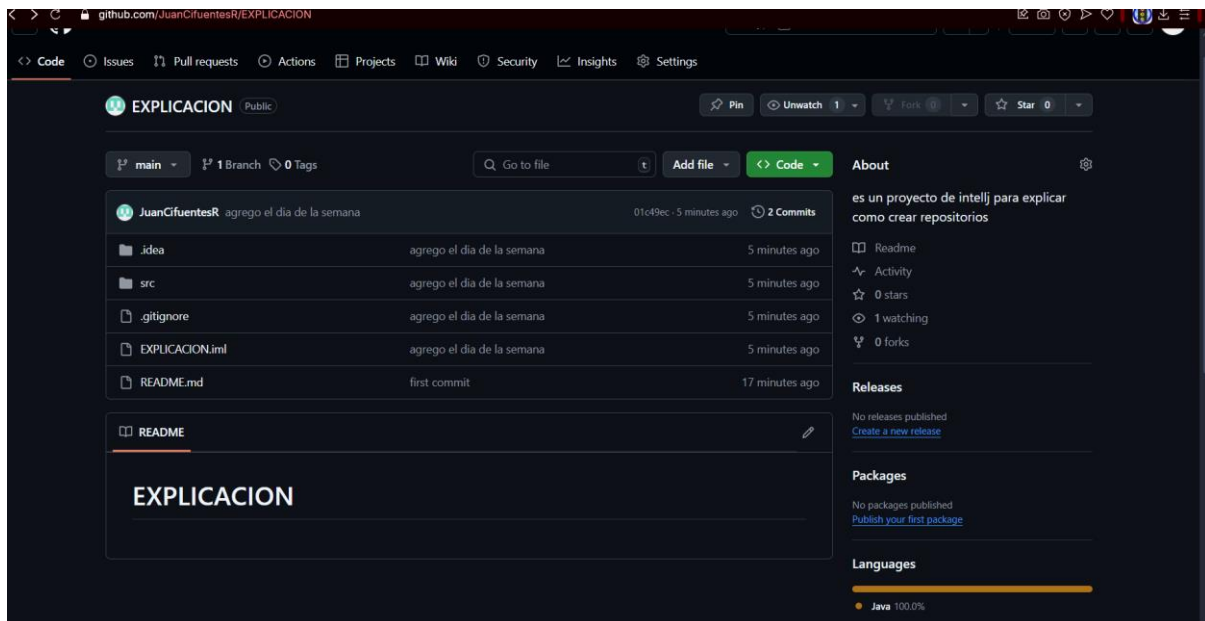
Git push: nos sirve para subir los cambios a GitHub

```

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)



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)

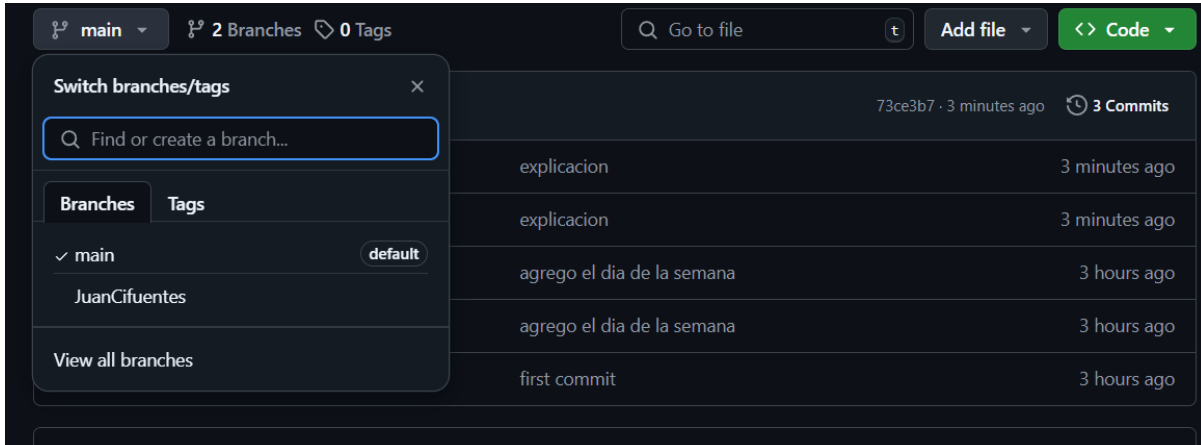
```
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git branch
* JuanCifuentes
  main
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION>
```

(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 aparecerá 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>
```



Commit Hash	Message	Time Ago
73ce3b7		3 minutes ago
	explicacion	3 minutes ago
	explicacion	3 minutes ago
	agrego el dia de la semana	3 hours ago
	agrego el dia de la semana	3 hours ago
	first commit	3 hours ago

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)

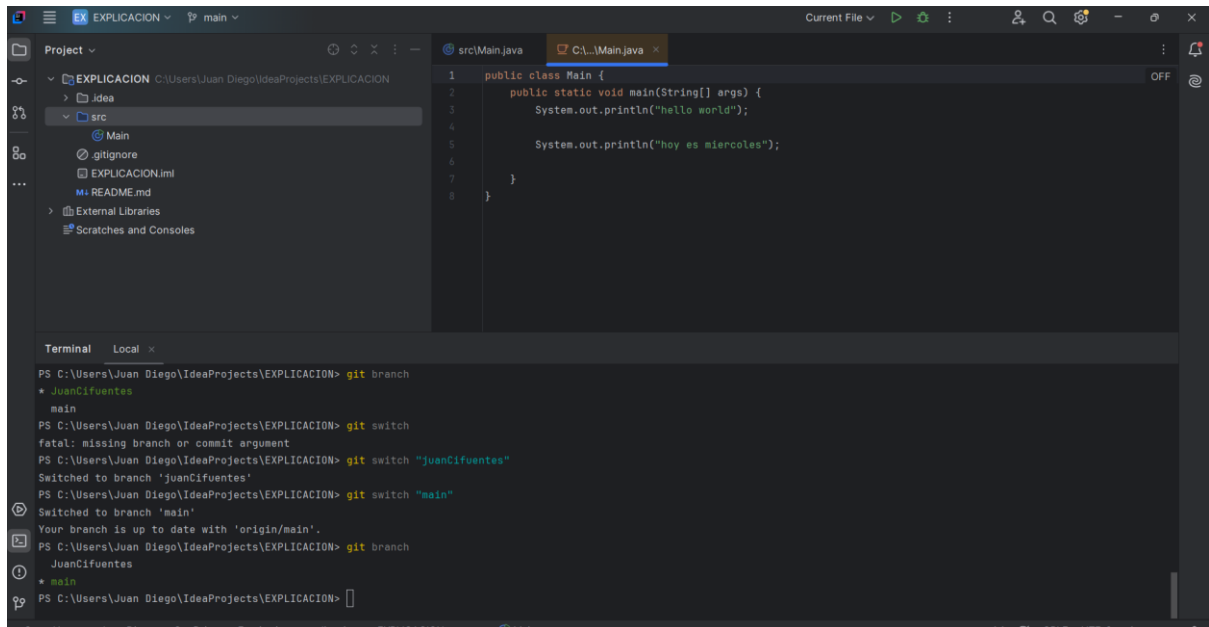
6.1 copiamos la url del repositorio que se desea clonar, en mi caso, mi mismo repositorio, para luego crear una carpeta vacía y allí dar clic derecho -> mostrar más opciones -> open git brash here y en la consola que se abre ponemos el siguiente codigo

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 abra el proyecto en IntelliJ IDEA

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



The screenshot shows an IDE with a project named 'EXPLICACION'. The file explorer on the left shows the project structure. The main editor displays a Java file 'src/Main.java' with the following code:

```
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
```

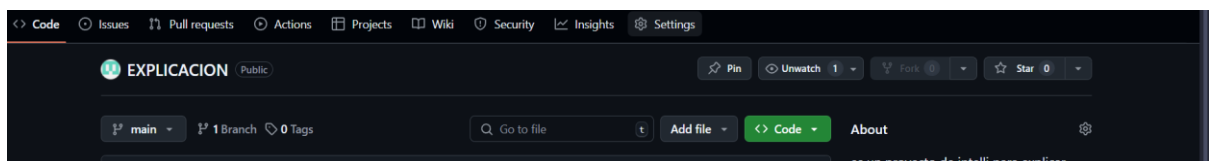
The terminal window at the bottom shows the following commands and output:

```
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git branch
* JuanCifuentes
main
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git switch
fatal: missing branch or commit argument
PS C:\Users\Juan Diego\IdeaProjects\EXPLICACION> git switch "JuanCifuentes"
Switched to branch 'JuanCifuentes'
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
* main
JuanCifuentes
```

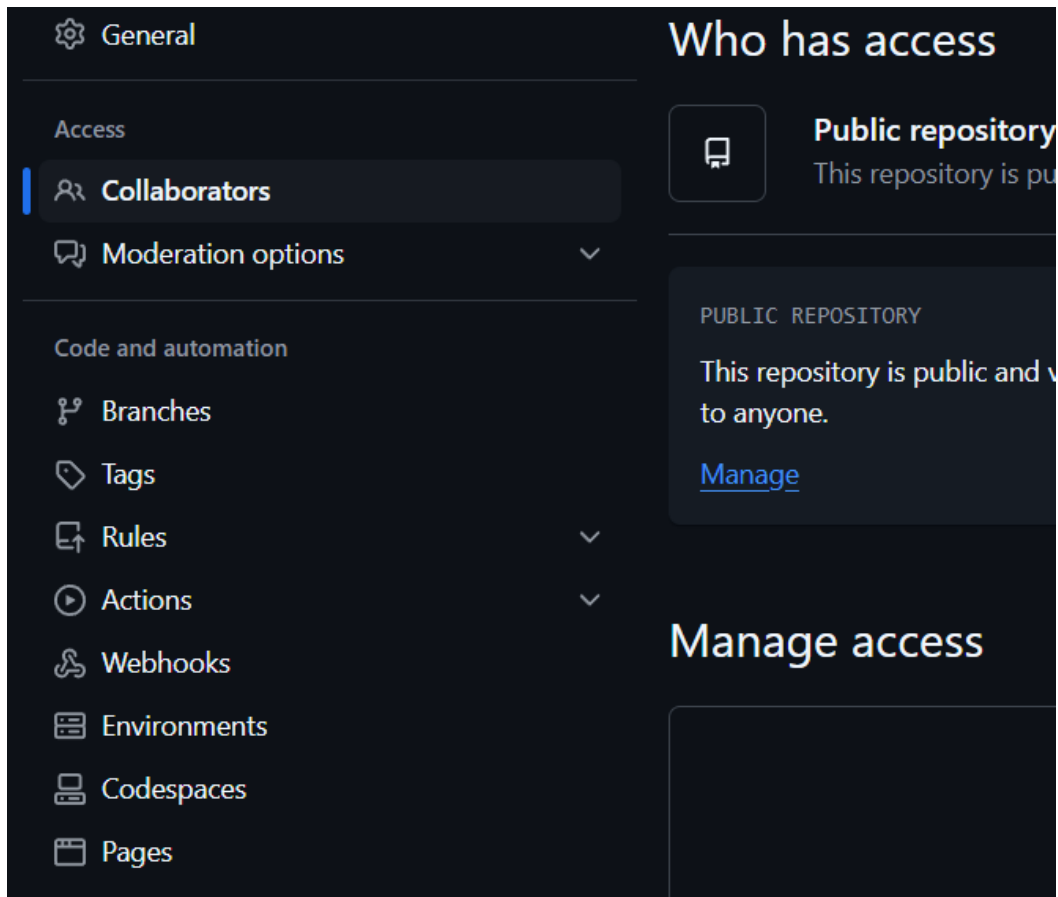
Séptimo paso: agregar colaboradores al repositorio

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”

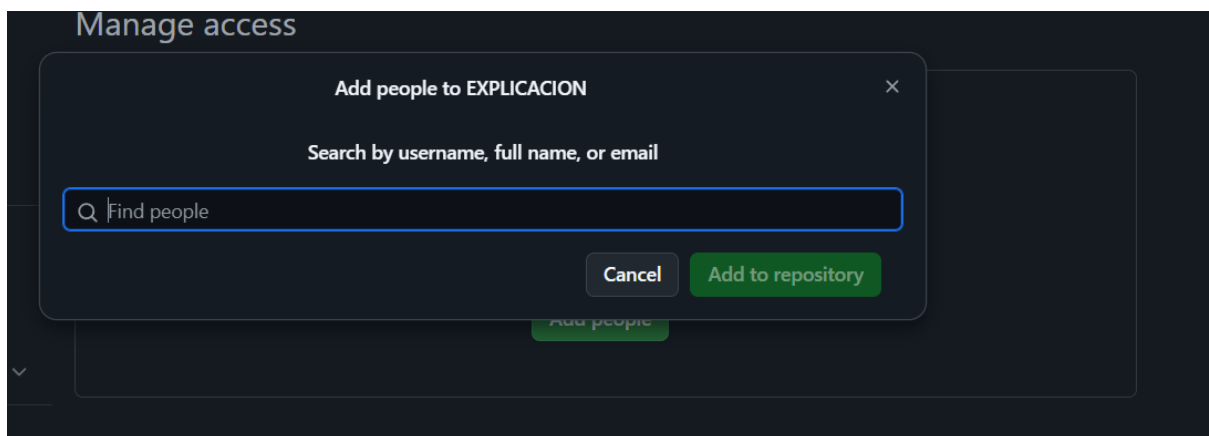
7.1 ingresar a settings en la parte superior



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

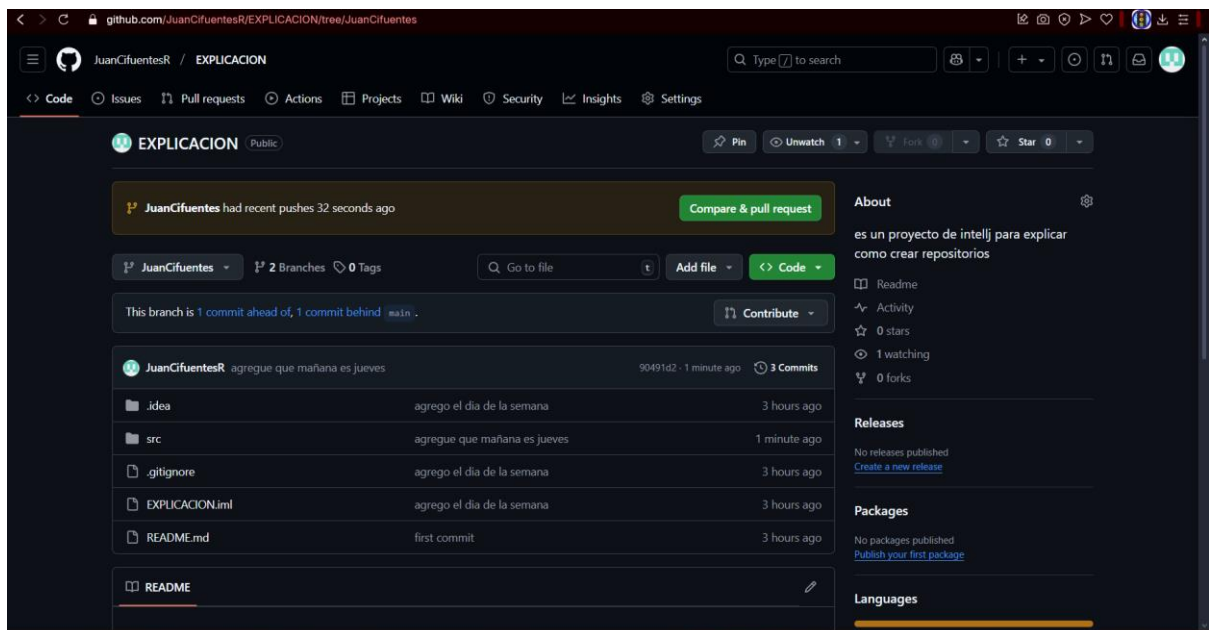
The screenshot shows an IDE with a project named 'EXPLICACION'. The 'Main.java' file is open, showing a Java class with a 'main' method. The terminal window displays the following commands and output:

```
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

The screenshot displays a GitHub pull request page. At the top, there's a section for adding files with a 'Create pull request' button. Below this, a commit summary shows '1 commit' and '1 file changed' by '1 contributor'. The commit message is 'agregue que mañana es jueves' by 'JuanCifuentesR' from 2 minutes ago. The diff view shows a change in 'src/Main.java', adding a new line of code: 'System.out.println("mañana es jueves");'. The right sidebar contains metadata like 'Milestone', 'Development', and 'Helpful resources'. The bottom section shows the pull request details, including a comment from 'JuanCifuentesR' and a 'Merge pull request' button. A green checkmark indicates 'No conflicts with base branch'.

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

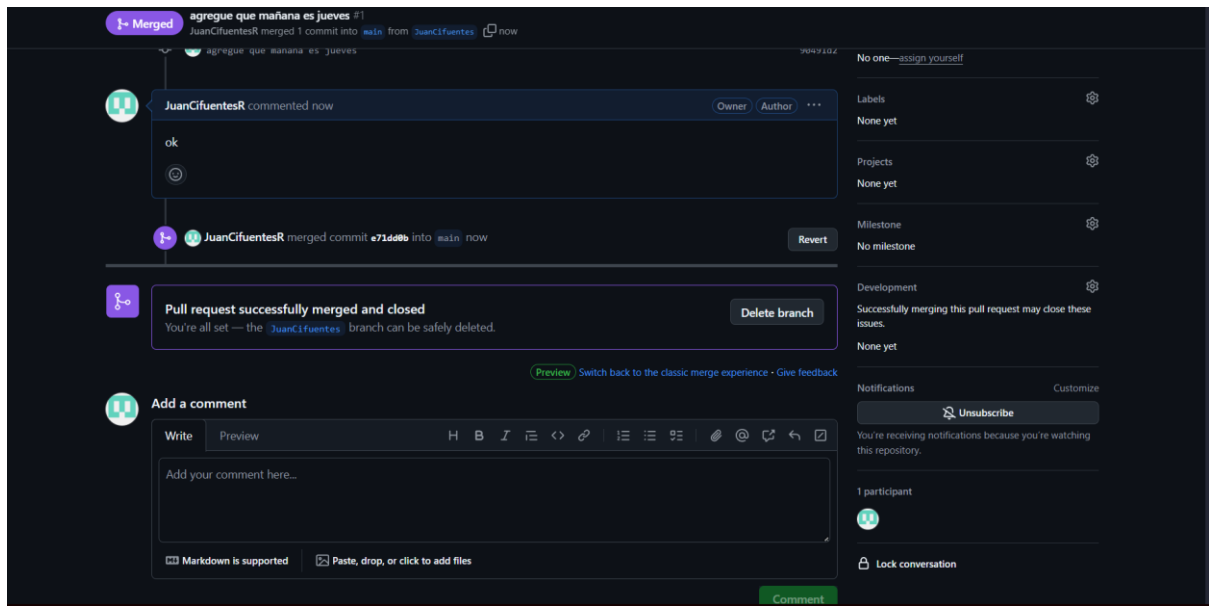
Projects
None yet

Milestone
No milestone

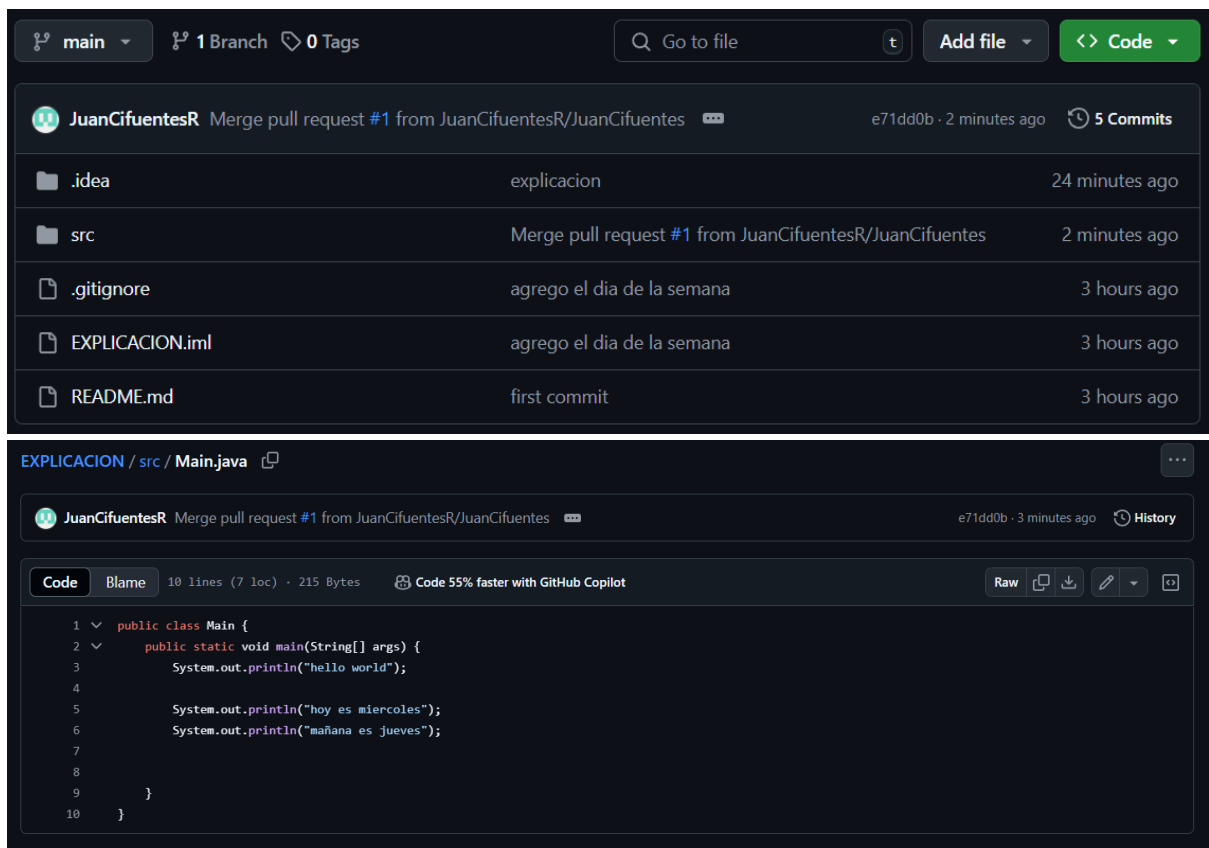
Development
Successfully merging this pull request may close these issues.
None yet

Notifications Customize
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

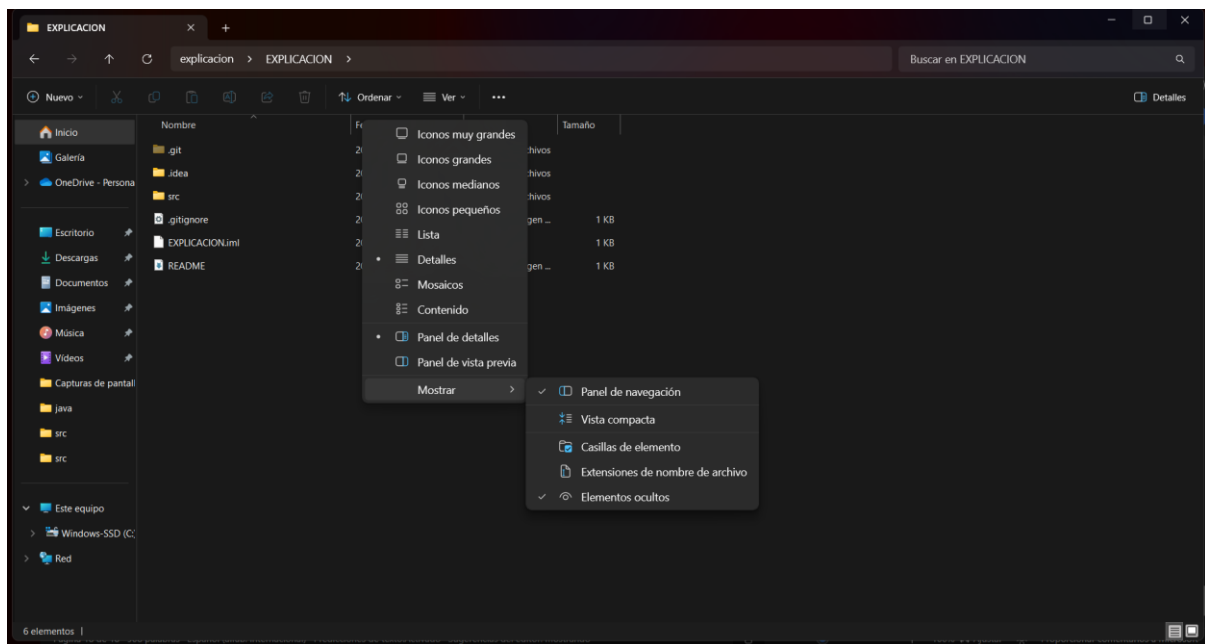
IntelliJ IDEA

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

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/>