

TAREA 2

Eres un desarrollador y quieres crear una aplicación de escritorio en Java llamada "MiAppJava". Para mantener un control adecuado de versiones, usarás Git. Seguirás un flujo de trabajo que incluirá ramas para nuevas funciones, commits organizados y gestión de versiones.

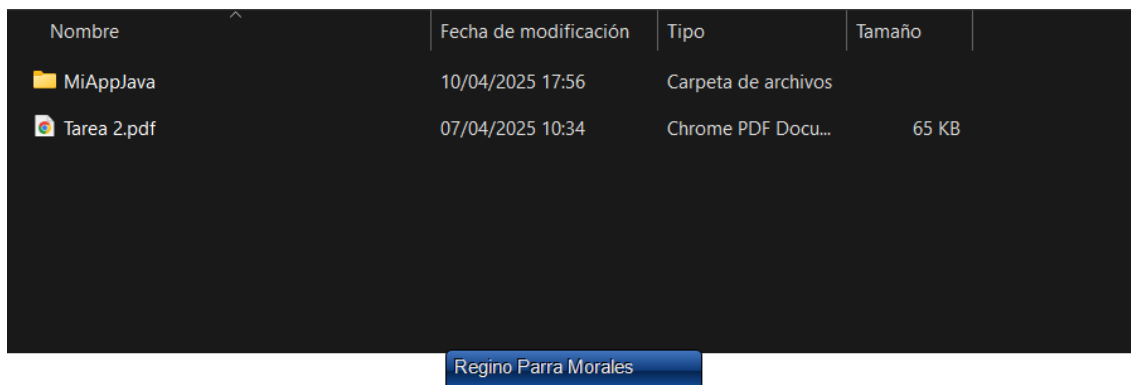
Control adecuado
de versiones de
Git

Índice

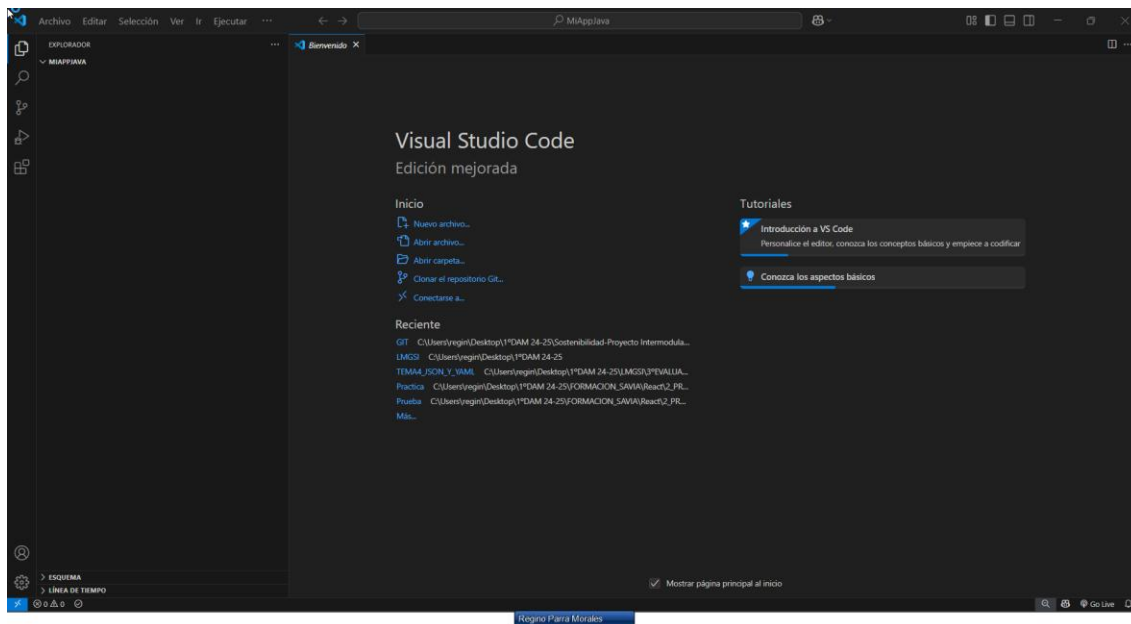
1. Crea un nuevo directorio para tu proyecto.	2
2. Inicializa un repositorio Git dentro del directorio.	3
3. Crea la estructura básica del proyecto.	4
4. Abre Main.java y agrega código.	6
5. Agregalo al área de preparación y realiza el primer commit.	6
6. Crea una nueva rama llamada modificacion1.	7
7. Cambiate a la nueva rama.	7
8. Crea una nueva clase que aporte una nueva funcionalidad.	8
9. Modifica Main.java para usar la nueva funcionalidad.	10
10. Agrega y confirma los cambios en la rama.	10
11. Cambia de vuelta a la rama principal (master).	11
12. Fusiona la rama modificación1 con master.	11
13. Elimina la rama modificacion1, ya que sus cambios fueron fusionados. 11	
14. Crea una etiqueta para marcar la versión 1.0 de la aplicación.	12
15. Verifica que la etiqueta ha sido creada.	12
16. Edita la última clase creada para hacer un commit con un error.	13
17. Agrega y confirma el commit con el error.	14
18. Revierte el último commit sin eliminarlo del historial.	14

1. Crea un nuevo directorio para tu proyecto.

Creamos la carpeta de MiAppJava

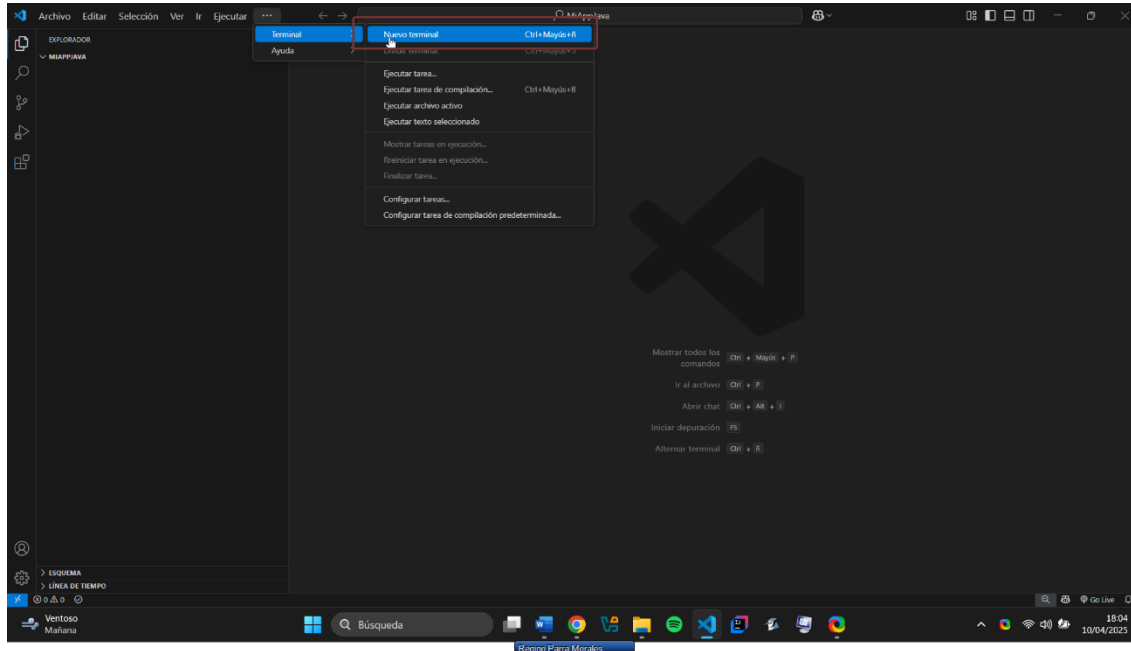


La metemos en Visual Studio Code

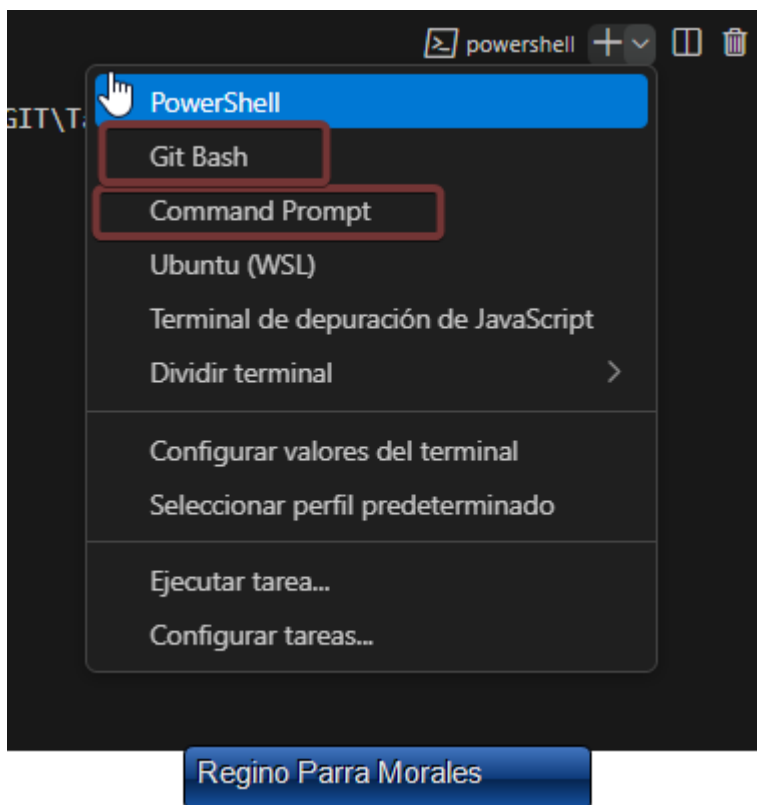


2. Inicializa un repositorio Git dentro del directorio.

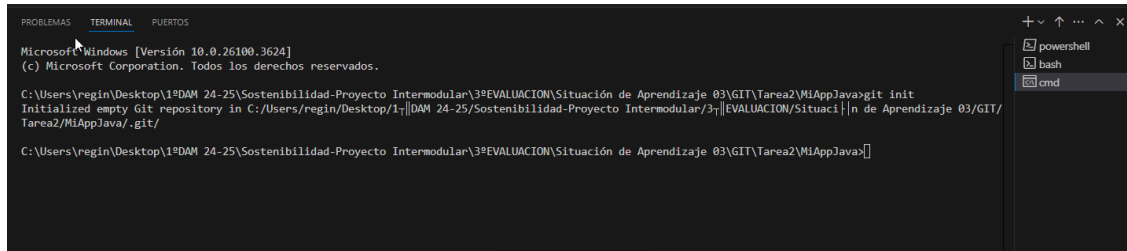
Nos vamos a la línea de comandos, en los 3 puntos arriba, terminal y nueva terminal



Podemos abrir la terminal o con git bash o con command prompt (CMD) en mi caso usare el CMD



En la línea de comandos escribimos **git init** y se nos creara una carpeta oculta, con este comando inicializamos git



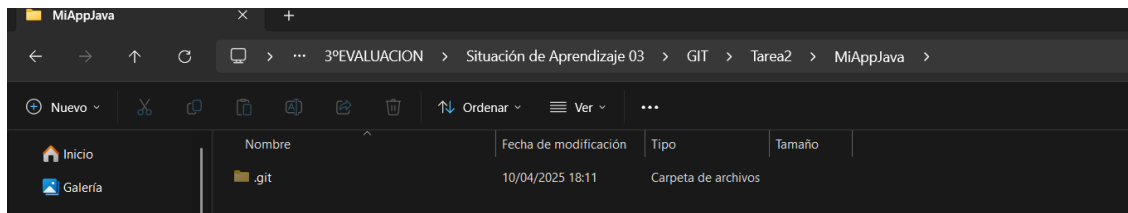
```
Microsoft Windows [Versión 10.0.26100.3624]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodal\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git init
Initialized empty Git repository in C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodal\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava\.git/

C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodal\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>
```

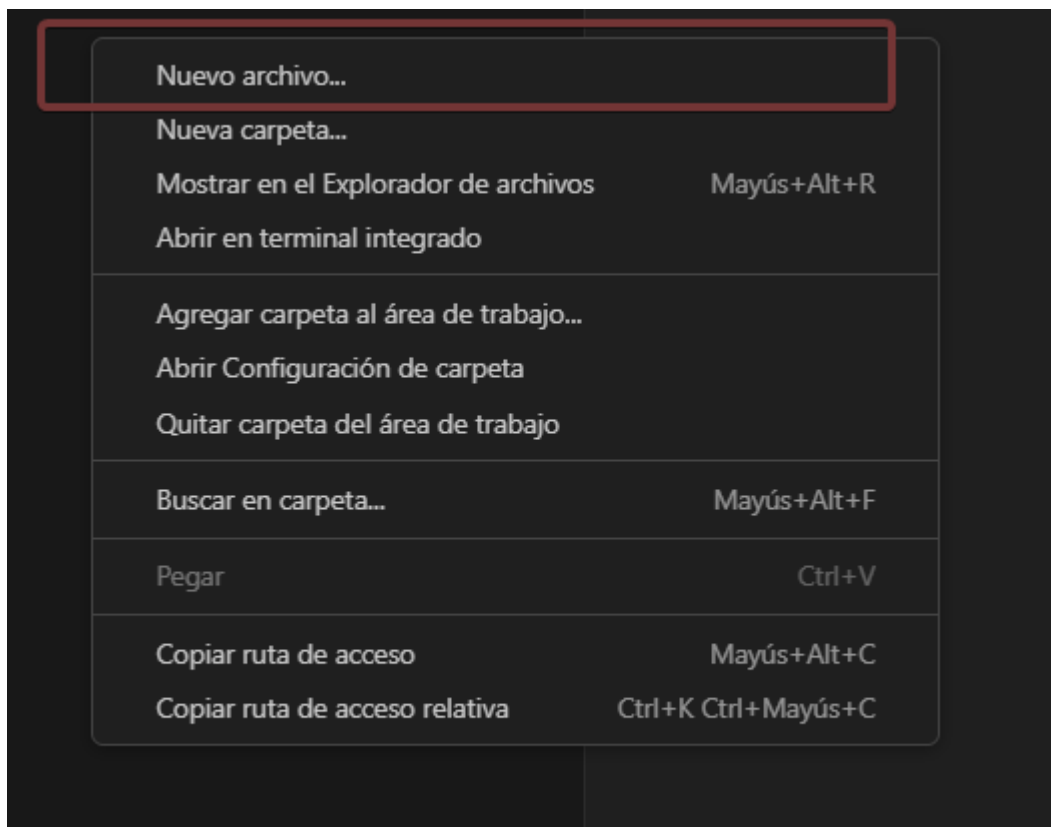
Regino Parra Morales

En la carpeta de MiAppJava, se podrá ver que se ha creado la carpeta



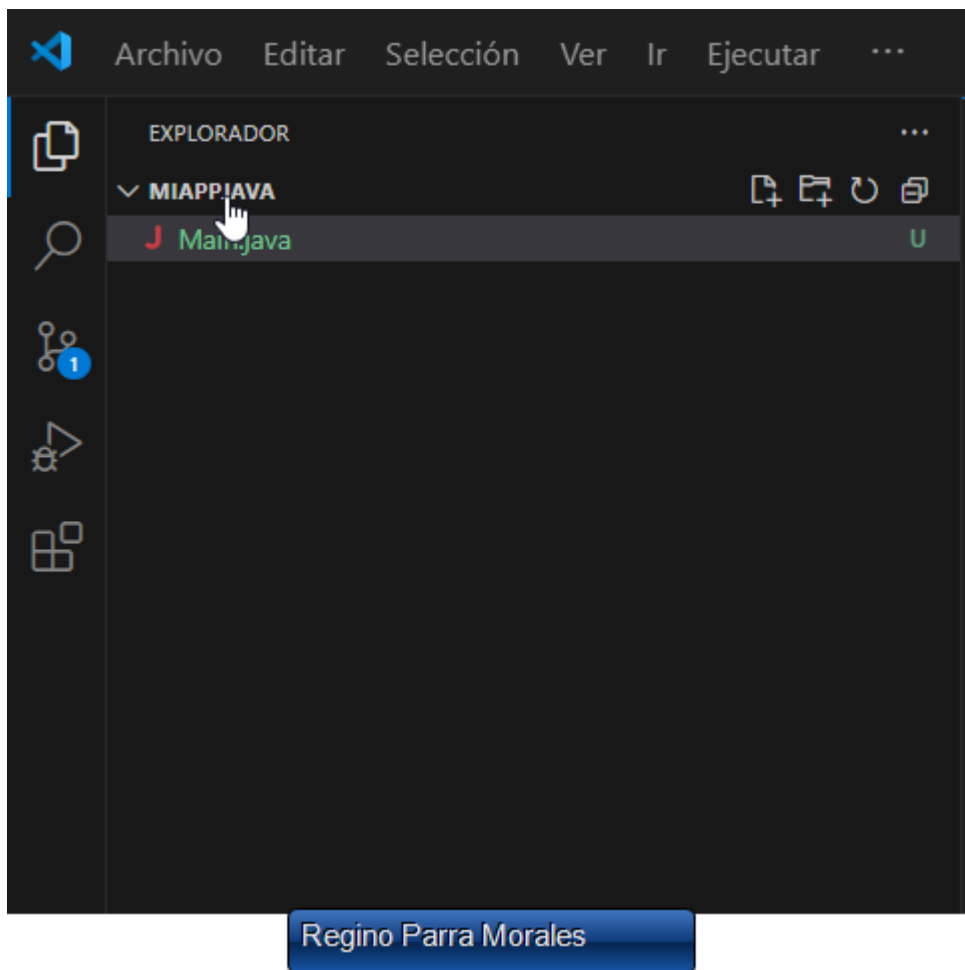
Regino Parra Morales

3. Crea la estructura básica del proyecto.



Regino Parra Morales

Creamos el archivo Main.java



4. Abre Main.java y agrega código.

Escribimos en el archivo Main.java, ese código :

```
public class Main {

    public static void main(String[] args) {

        System.out.println("Hola desde MiAppJava");

    }

}
```

Regino Parra Morales

5. Agregalo al área de preparación y realiza el primer commit.

Hacemos un : **git add .** , para agregar todo lo que este en la carpeta, en este caso el Main.java

```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git add .
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>
```

Regino Parra Morales

Hacemos un :

git commit -m "Primer commit: estructura básica y Main.java"

```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git add .
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git commit -m "Primer commit: estructura básica y Main.java"
[master (root-commit) 84c885e] Primer commit: estructura básica y Main.java
1 file changed, 5 insertions(+)
create mode 100644 Main.java
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>
```

Regino Parra Morales

6. Crea una nueva rama llamada modificacion1.

Creamos una rama:

git branch modificacion1

Y para ver la lista de ramas:

git branch

```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git branch modificacion1
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git branch
* master
  modificacion1
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>
```

Regino Parra Morales

7. Cambiate a la nueva rama.

Para cambiarse a la rama nueva creada ,escribimos el siguiente comando:

git checkout modificacion1

Y volvemos a escribir:

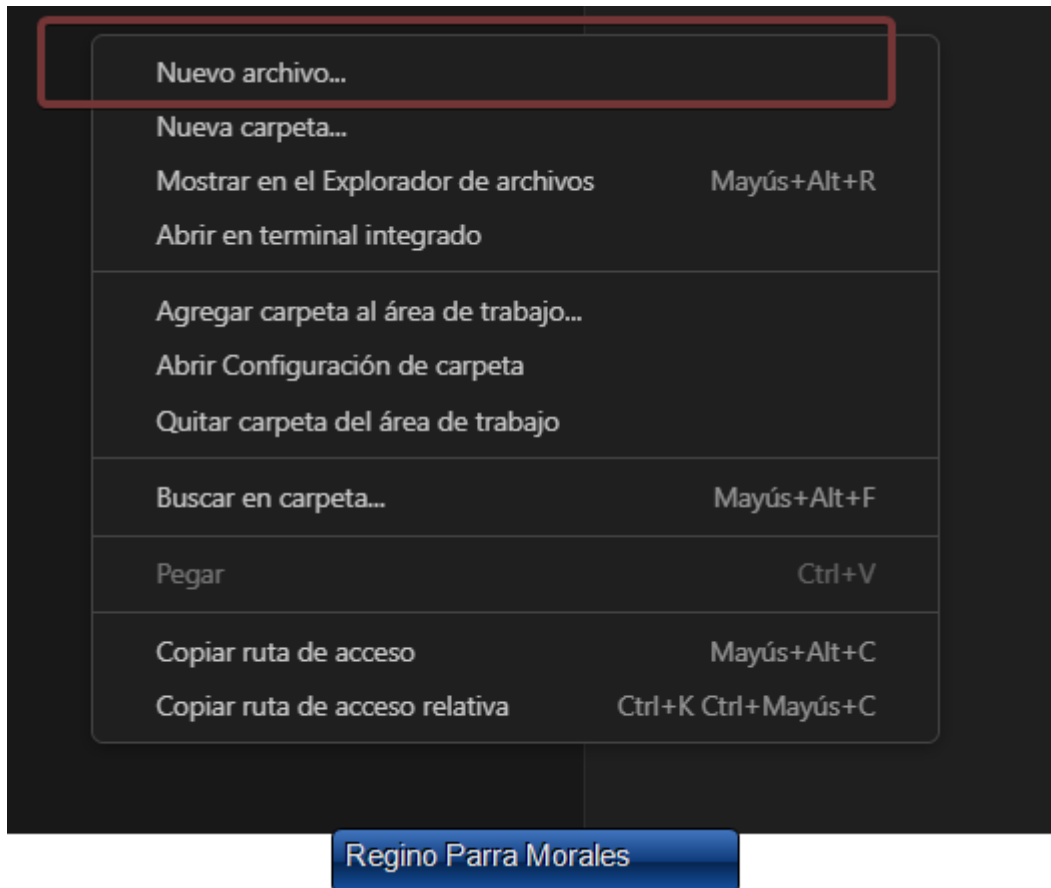
git branch

```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git checkout modificacion1
Switched to branch 'modificacion1'
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git branch
  master
* modificacion1
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>
```

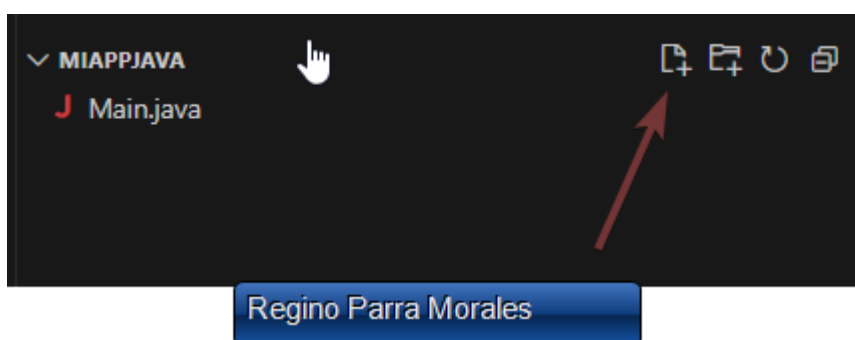
Regino Parra Morales

8. Crea una nueva clase que aporte una nueva funcionalidad.

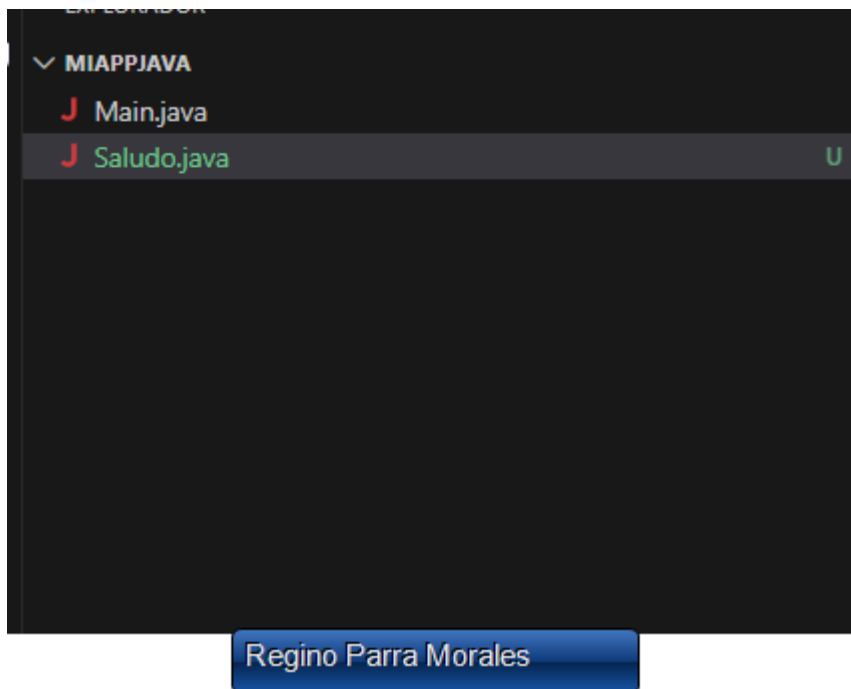
Volvemos a crear un archivo



O también a ese icono

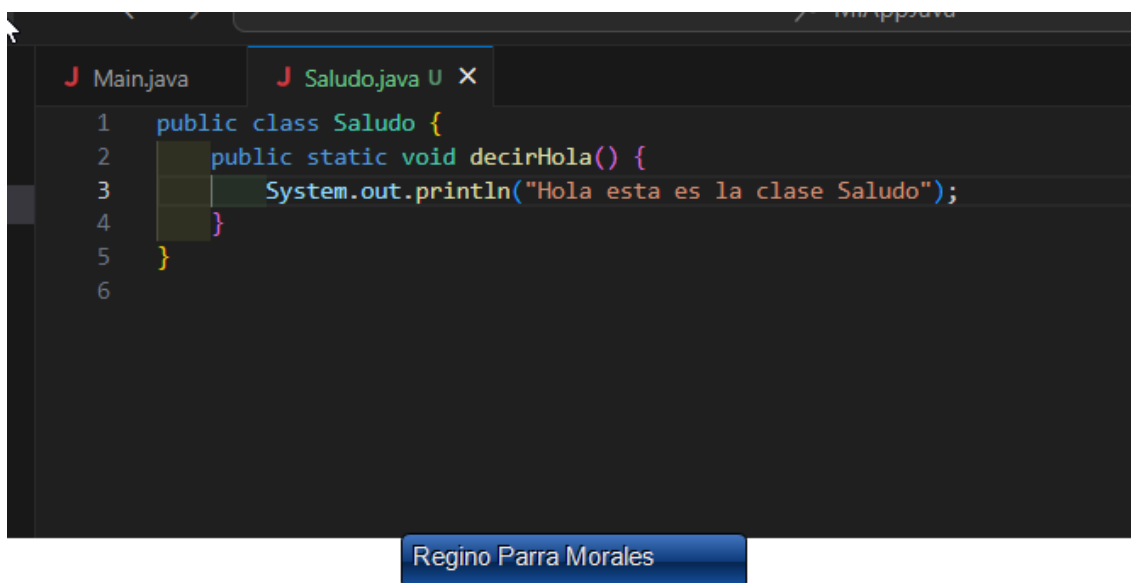


Archivo creado



Escribimos contenido en Saludo.java:

```
public class Saludo {  
    public static void decirHola() {  
        System.out.println("Hola desde la clase Saludo");  
    }  
}
```



9. Modifica Main.java para usar la nueva funcionalidad.

Modificamos con el siguiente código:

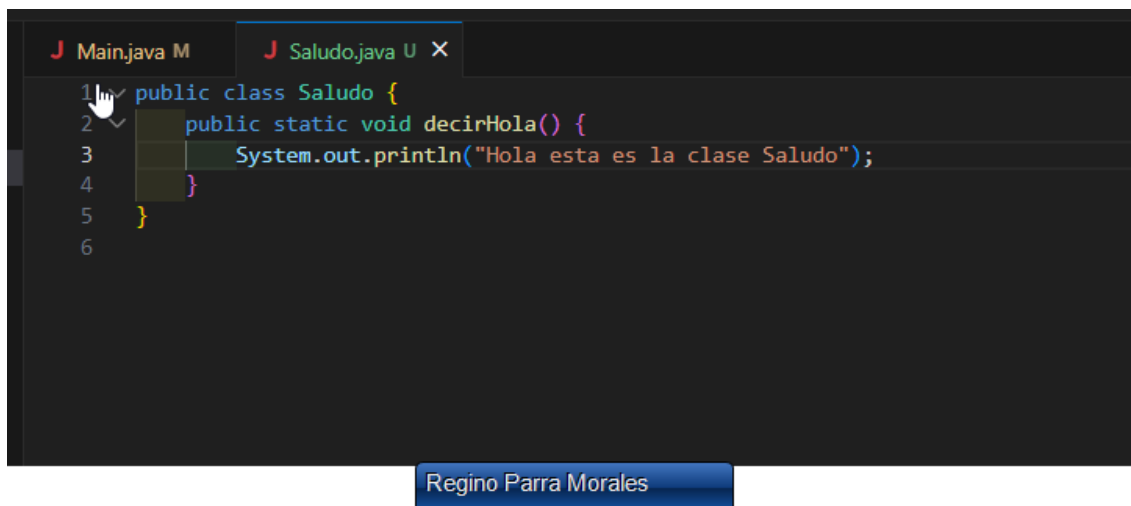
```
public class Main {

    public static void main(String[] args) {

        Saludo.decirHola();

    }

}
```

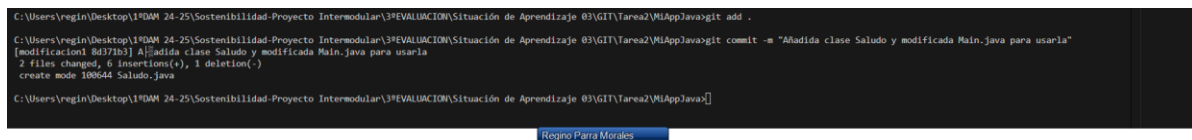


10. Agrega y confirma los cambios en la rama.

Agregamos con este código otro commit:

git add .

git commit -m "Añadida clase Saludo y modificada Main.java para usarla"



11. Cambia de vuelta a la rama principal (master).

Cambiamos de vuelta a la rama principal

git checkout master

Y verificamos que lo hicimos con :

git branch

```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git checkout master
Switched to branch 'master'

C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git branch
* master
  modificacion1

C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>
```

12. Fusiona la rama modificación1 con master.

Fusionamos las dos ramas , la creada y la principal con el siguiente comando:

git merge modificacion1

```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git merge modificacion1
Already up to date.
```

13. Elimina la rama modificacion1, ya que sus cambios fueron fusionados.

Escribimos el siguiente comando para eliminar la rama modificacion1:

git branch -D modificacion1

Y verificamos que lo hemos creado con:

git branch

```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git branch -D modificacion1
Deleted branch modificacion1 (was 8d371b3).

C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git branch
* master

C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>
```

14. Crea una etiqueta para marcar la versión 1.0 de la aplicación.

Para escribir una etiqueta escribimos:

git tag v1.0

```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git tag v1.0
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>
```

Regino Parra Morales

15. Verifica que la etiqueta ha sido creada.

Para verificar la etiqueta creada escribimos:

git tag

```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git tag v1.0
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>
```

Regino Parra Morales

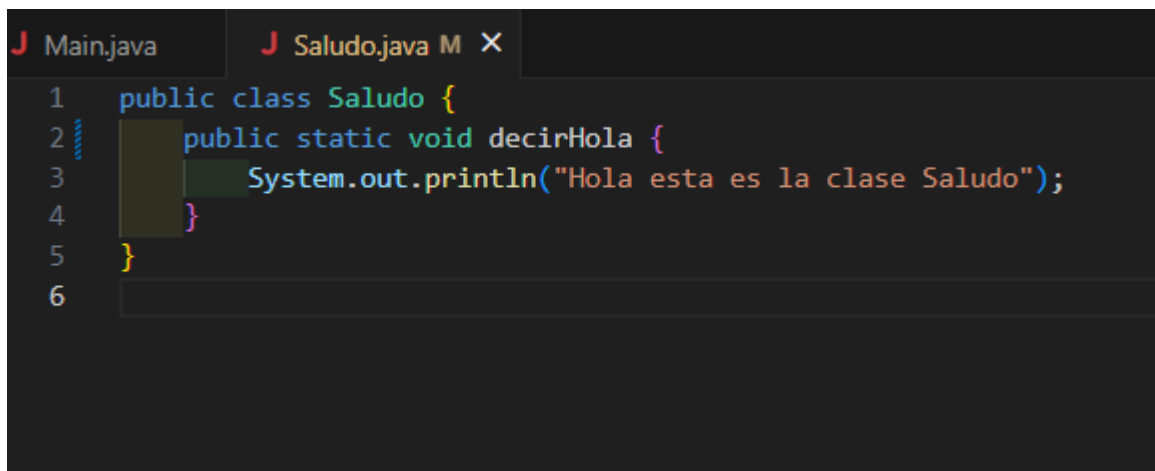
```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git tag
v1.0
```

Regino Parra Morales

16. Edita la última clase creada para hacer un commit con un error.

Introducimos un error en la clase creada ultima de Saludo.java quitamos el paréntesis de decirHola() quedaría asi:

```
public class Saludo {  
    public static void decirHola {  
        System.out.println("Hola esta es la clase Saludo");  
    }  
}
```



The screenshot shows an IDE with two tabs: 'Main.java' and 'Saludo.java M X'. The 'Saludo.java' tab is active, displaying the following code with line numbers 1 through 6 on the left. The code is:
1 public class Saludo {
2 public static void decirHola {
3 System.out.println("Hola esta es la clase Saludo");
4 }
5 }
6
A red squiggly line is under the opening curly brace of the 'decirHola' method on line 2, indicating a syntax error because the closing parenthesis is missing.

17. Agrega y confirma el commit con el error.

Hacemos un commit con el error cometido:

```
git add Saludo.java
```

```
git commit -m "Commit con error en Saludo.java"
```

```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git add Saludo.java
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git commit -m "Commit con error en Saludo.java"
[master 354abc3] Commit con error en Saludo.java
```

Regino Parra Morales

18. Revierte el último commit sin eliminarlo del historial.

```
git revert HEAD
```

Para revertir al ultimo commit me sale asi con los siguiente comandos de la captura:

```
git revert 354abc3
```

Y luego:

```
git log --oneline
```

```
354abc3 (HEAD -> master) Commit con error en Saludo.java
0ef1aaf (tag: v1.0) Añadida clase Saludo y modificada Main.java para usarla
8d371b3 Añadida clase Saludo y modificada Main.java para usarla
84c885e Primer commit: estructura básica y Main.java

C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git revert 354abc3
error: commit 354abc387cd79f718dd03c4a37ca1d4a023bcc2 is a merge but no -m option was given.
fatal: revert failed
```

Regino Parra Morales

```
C:\Users\regin\Desktop\1ºDAM 24-25\Sostenibilidad-Proyecto Intermodular\3ºEVALUACION\Situación de Aprendizaje 03\GIT\Tarea2\MiAppJava>git log --oneline
354abc3 (HEAD -> master) Commit con error en Saludo.java
0ef1aaf (tag: v1.0) Añadida clase Saludo y modificada Main.java para usarla
8d371b3 Añadida clase Saludo y modificada Main.java para usarla
84c885e Primer commit: estructura básica y Main.java
```

Regino Parra Morales

Como se ve me da un pequeño error pero sale la línea de comandos, así que ya está