

## Manual Instructivo

### Programación II

Profesor: William Alexander Matallana Porras

Estudiante: Tania Lorena Beltrán Lara

Facultad ingeniería de sistemas y computación 2022

Universidad de Cundinamarca extensión Chía

2025

## Tabla de contenido

INTRODUCCION .....	4
OBJETIVOS.....	5
DESCRIPCIÓN – SCREEN.....	6
¿CÓMO DESASOCIAR EL CORREO ELECTRONICO Y EL NOMBRE? .....	6
¿CÓMO ASOCIAR EL NOMBRE DE USUARIO Y EL CORREO ELECTRONICO? ....	6
<b>VERIFICACION CON GIT</b> CONFIG – -LIST .....	8
¿CÓMO CREAR UN REPOSITORIO? .....	9
¿COMO ELIMINAR UN REPOSITORIO? .....	16
¿COMO CREAR RAMAS? .....	18
¿COMO ELIMINAR LAS RAMAS? .....	20
¿COMO CREAR COMMIT EN LAS RAMAS? .....	21
¿COMO UTILIZAR GIT PULL ORIGIN RAMA?.....	23
COMANDOS .....	26
❖.....El comando <code>git log --oneline nombre rama</code>	26
❖.....El comando <code>git Branch -r</code>	26
❖.....El <code>Git merge+ (--Rebase)</code>	26

❖ .....	Git log
27	
❖ .....	git reflog
28	
❖ .....	git Branch -d (Nombre de la Rama)
29	
Git fetch - -all .....	29
Git reverse .....	31
¿COMO CLONAR UN ARCHIVO? .....	33
CONCLUSIONES.....	35
BIBLIOGRAFIA.....	36

## INTRODUCCION

IntelliJ IDEA es uno de los entornos de desarrollo integrado (IDE) más populares y potentes para programadores, especialmente en el desarrollo con Java y otros lenguajes como Kotlin, Python y JavaScript. Su eficiencia se debe en gran parte a su amplio conjunto de comandos y atajos de teclado, que permiten optimizar el flujo de trabajo, mejorar la productividad y reducir el tiempo dedicado a tareas repetitivas.

## OBJETIVOS

El presente manual tiene como finalidad proporcionar una guía clara y detallada sobre el uso de **IntelliJ IDEA**, que IDEA es una de las herramientas más populares para el desarrollo en **Java**, **Kotlin**, y muchos otros lenguajes, gracias a su inteligencia, facilidad de uso y una amplia gama de funcionalidades que optimizan la productividad del programador. Permitiendo a los usuarios aprender cómo se maneja y sus conceptos básicos.

## DESCRIPCIÓN – SCREEN

### ¿CÓMO DESASOCIAR EL CORREO ELECTRONICO Y EL NOMBRE?

**Paso 1:** `git config --global --unset user.email` El siguiente comando eliminará la dirección de correo electrónico previamente configurada a nivel global.

**Paso 2:** `git config --global --unset user.name` El siguiente comando eliminará el nombre de usuario configurado globalmente en Git.

```
PS C:\Users\Casa\IdeaProjects\PrimerEntregable> git config --global --unset user.email
PS C:\Users\Casa\IdeaProjects\PrimerEntregable> git config --global --unset user.name
git: 'congig' is not a git command. See 'git --help'.

The most similar command is
    config
PS C:\Users\Casa\IdeaProjects\PrimerEntregable>
```

### ¿CÓMO ASOCIAR EL NOMBRE DE USUARIO Y EL CORREO ELECTRONICO?

**Paso 1:** `Git config --global user.email` este comando establece el correo electrónico que se asociará a los commits en Git.

**Paso 2:** `Git config --global user.name` este comando asigna un nombre de usuario que se usará en los commits.

```
PS C:\Users\Casa\IdeaProjects\PrimerEntregable> git config --global user.email "taniabeltran2020@gmail.com"
PS C:\Users\Casa\IdeaProjects\PrimerEntregable> git config --global user.name "TANI1864"
PS C:\Users\Casa\IdeaProjects\PrimerEntregable> git config --global --list
user.name=TANI1864
user.emai=taniabeltran2020@gmail.com
user.email=taniabeltran2020@gmail.com
PS C:\Users\Casa\IdeaProjects\PrimerEntregable> 
```

## VERIFICACION CON GIT CONFIG --LIST

muestra la configuración actual de Git, incluyendo valores globales, locales (por repositorio) y del sistema, incluyendo usuario, correo, editor predeterminado, configuración de credenciales.

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
```

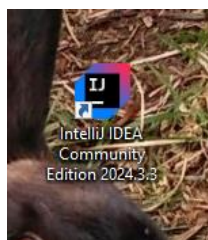
```
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=TANI1864
user.emai=taniabeltran2020@gmail.com
user.email=taniabeltran2020@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/TANI1864/PrimerTrabajoTania.git
:
```



## ¿CÓMO CREAR UN REPOSITORIO?

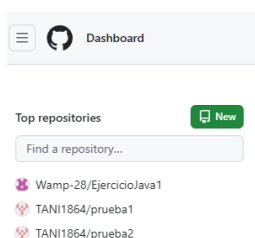
### Paso 1: Abrimos el IntelliJ IDEA community

IntelliJ IDEA ofrece compatibilidad con una amplia variedad de marcos de trabajo tanto para el desarrollo del lado del servidor como del front-end. Además, permite la integración con servidores de aplicaciones, herramientas de gestión de bases de datos y herramientas de creación de perfiles para la optimización del rendimiento del código.

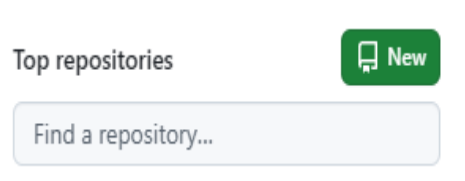


### Paso 2: Abrimos el GitHub

GitHub es una plataforma de desarrollo que permite a los desarrolladores crear, almacenar, administrar y compartir su código de manera eficiente. Basada en Git, proporciona un sistema de control de versiones distribuido, junto con herramientas para la gestión de accesos, el seguimiento de errores, la solicitud de nuevas funciones, la administración de tareas, la integración continua y la documentación colaborativa a través de wikis para cada proyecto.



**Paso 3:** Para Crear un nuevo repositorio damos Clic en la opción New



**Paso 4:** En la opción Repository name le damos un nombre a nuestro proyecto, en la opción de descripción es algo opcional

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


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


Owner \*      Repository name \*

 TANI1864 /

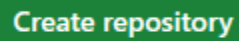
✓ PrimerTrabajoTania is available.

- Se debe tener en cuenta que este en la opción Public

☒  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.

- Dar clic



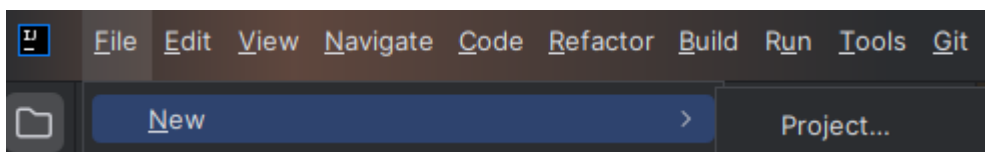
**Paso 5:** Al dar el clic el Create Repository no va a llevar aquí, Las líneas de código que nos aparecen las vamos a copiar y vamos a pegarlas en el IntelliJ IDEA community.

```
Quick setup — if you've done this kind of thing before
[Set up in Desktop] or [HTTPS] [SSH] https://github.com/TANI1864/PruebaTrabajoEscrito.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.
```

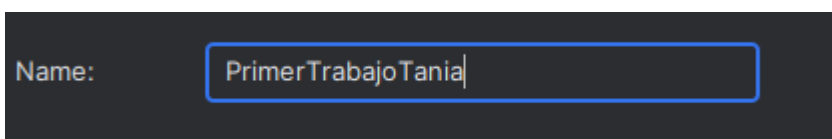
---

```
...or create a new repository on the command line
echo "# PruebaTrabajoEscrito" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/TANI1864/PruebaTrabajoEscrito.git
git push -u origin main
```

**Paso 6:** Vamos al IntelliJ y creamos un nuevo proyecto



**Paso 7:** Le damos un Nombre



### Paso 8: Damos clic en el terminal



- Pegamos la línea de código del GitHub al IntelliJ Idea

```
PS C:\Users\Casa\IdeaProjects\PruebaTrabajoEscrito> echo "# PruebaTrabajoEscrito" >> README.md
PS C:\Users\Casa\IdeaProjects\PruebaTrabajoEscrito> git init
Initialized empty Git repository in C:/Users/Casa/IdeaProjects/PruebaTrabajoEscrito/.git/
PS C:\Users\Casa\IdeaProjects\PruebaTrabajoEscrito> git add README.md
PS C:\Users\Casa\IdeaProjects\PruebaTrabajoEscrito> git commit -m "primer commit"
[master (root-commit) 784b46d] primer commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\Users\Casa\IdeaProjects\PruebaTrabajoEscrito> git branch -M main
PS C:\Users\Casa\IdeaProjects\PruebaTrabajoEscrito> git remote add origin https://github.com/TANI1864/PruebaTrabajoEscrito.git
PS C:\Users\Casa\IdeaProjects\PruebaTrabajoEscrito> git push -u origin main
```

**Paso 9:** Ponemos en comando `git status` que muestra el estado del directorio de trabajo y del área de preparación, permitiendo identificar qué cambios han sido preparados, cuáles aún no lo han sido y qué archivos no están siendo rastreados por Git. Sin embargo, el resultado del estado no proporciona información sobre el historial del proyecto.

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> 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/
        PrimerTrabajoTania.iml
        src/

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

**Paso 10:** Escribimos el comando `git add`, que añade al área de preparación (staging área) todos los cambios realizados en el directorio actual y sus subdirectorios, incluyendo archivos nuevos, modificados o eliminados, para su inclusión en el próximo commit."

```
nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/Main.java', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> |
```

**Paso 11:** Escribimos el `git commit` se usa para guardar una instantánea de esos cambios en el historial del repositorio. [https://github.com/mokuzaru/git\\_tutorial](https://github.com/mokuzaru/git_tutorial)

```
commit
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git commit -m "Hola chiquistriquis"
[main 986bf12] Hola chiquistriquis
 7 files changed, 78 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 PrimerTrabajoTania.iml
 create mode 100644 src/Main.java
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania>
```

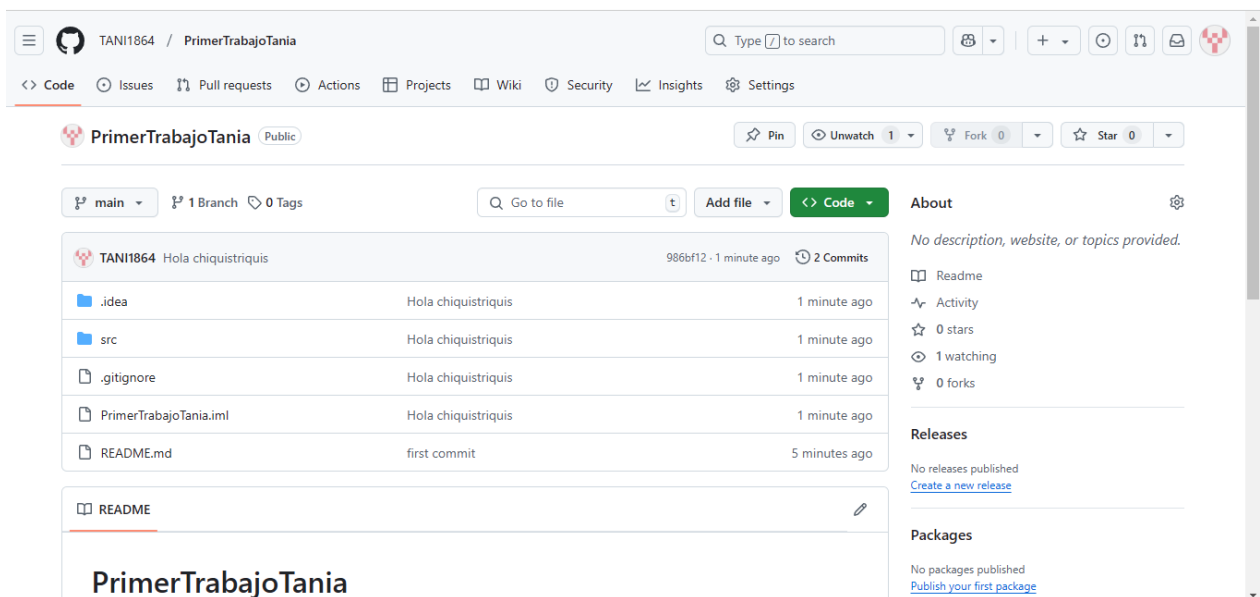
**Paso 12:** Escribimos el comando `git push origin main` permite subir los commits desde tu rama (Branch) local en tu repositorio git local al repositorio remoto.

```

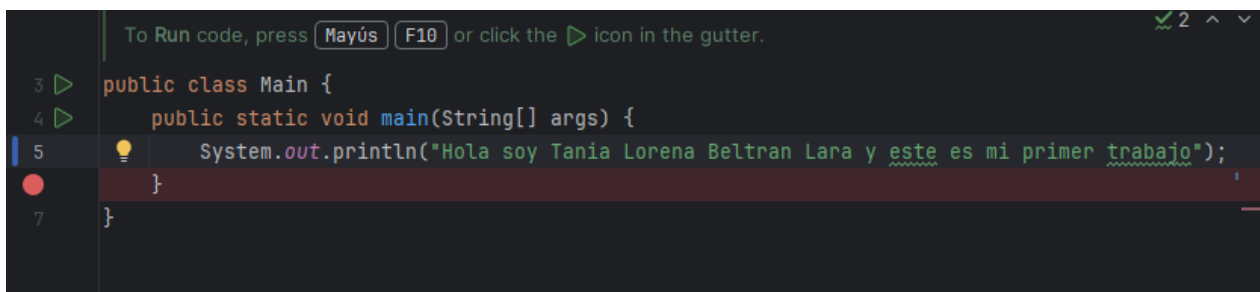
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git push origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (11/11), 2.00 KiB | 1.00 MiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/TANI1864/PrimerTrabajoTania.git
 400b95b..986bf12  main -> main
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania>

```

**Paso 13:** Vamos al repositorio y refrescamos la página y nos debe aparecer que el repositorio ya está creado.



**Paso 14:** Escribimos un mensaje en la pantalla de código en IntelliJ.



**Paso 15:** Escribimos el comando en la terminal `git init` que se utiliza para inicializar un nuevo repositorio de Git en un directorio.

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git init
Reinitialized existing Git repository in C:/Users/Casa/IdeaProjects/PrimerTrabajoTania/.git/
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania>
```

**Paso 16:** Volvemos a escribir todos los comandos anteriores

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git init
Reinitialized existing Git repository in C:/Users/Casa/IdeaProjects/PrimerTrabajoTania/.git/
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> 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\Casa\IdeaProjects\PrimerTrabajoTania> git commit -m "Hola nenes"
[main c6ab4c9] Hola nenes
1 file changed, 1 insertion(+), 9 deletions(-)
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 408 bytes | 408.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/TANI1864/PrimerTrabajoTania.git
986bf12..c6ab4c9 main -> main
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania>
```

**Paso 17:** Vamos al git Hub y refrescamos la pagina

Commit **c6ab4c9** Preview Give feedback

TANI1864 committed 1 minute ago

Hola nenes

main 1 parent 986bf12 co

Filter files...

src

Main.java

1 file changed +1 -9 lines changed

src/Main.java

```

@@ -2,14 +2,6 @@
2 2 // click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.
3 3 public class Main {
4 4     public static void main(String[] args) {
5 5         //TIP Press <shortcut actionId="ShowIntentionActions"/> with your care
6 6         // to see how IntelliJ IDEA suggests fixing it.
7 7         System.out.printf("Hello and welcome!");
8 8
9 9         for (int i = 1; i <= 5; i++) {
10 10             //TIP Press <shortcut actionId="Debug"/> to start debugging your code. We have set one <icon
src="AllIcons.Debugger.Db_set_breakpoint"/> breakpoint
11 11             // for you, but you can always add more by pressing <shortcut actionId="ToggleLineBreakpoint"/>.
12 12             System.out.println("i = " + i);
13 13         }

```

Search within code

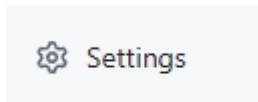
Customizable line height

The default line height has been increased to improve accessibility. You can choose to enable compact line height from the view settings menu.

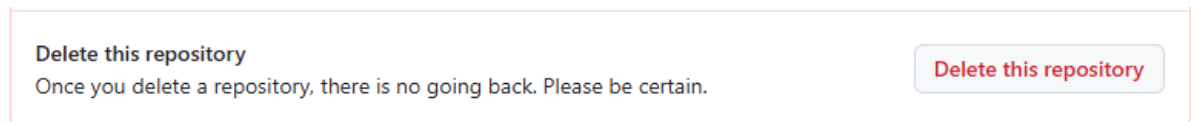
Enable compact line height

## ¿COMO ELIMINAR UN REPOSITORIO?

**PASO 1:** Vamos a la opción setting



**PASO 2:** Nos vamos hasta el final de la página y damos clic en Delete this Repository

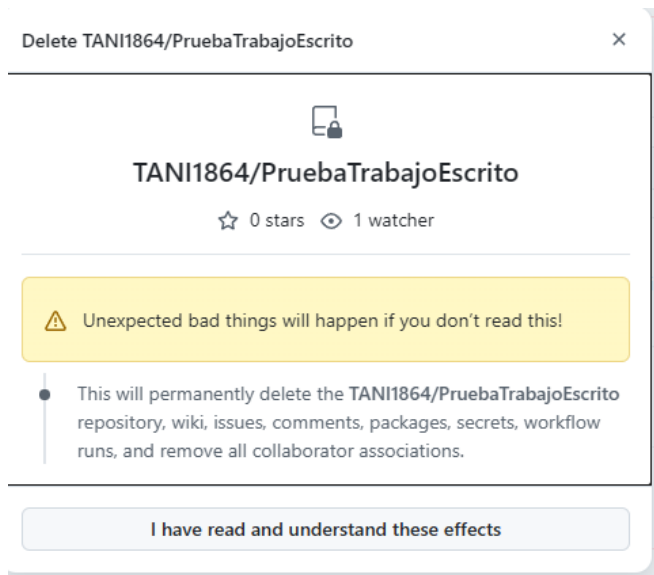


**Paso 3:** Damos clic en I want to delete this Repository

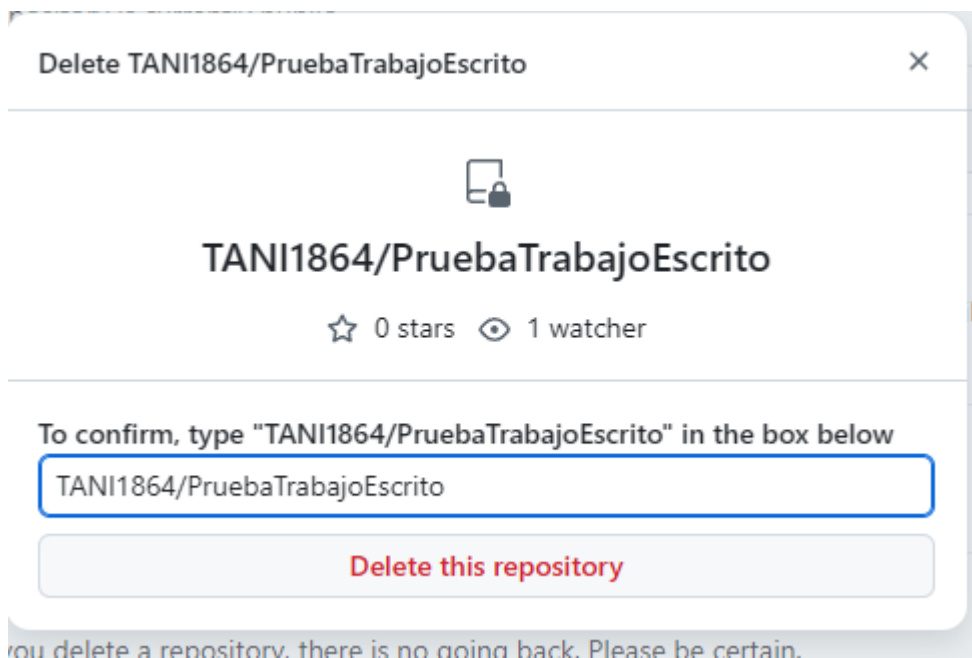


**Paso 4:** Damos clic en I have read and understad these effects





**PASO 5:** Copiamos y pegamos el nombre y damos clic.



## ¿COMO CREAR RAMAS?

**Paso 1:** Escribimos en la terminal el comando `git Branch` es una bifurcación del estado del código que crea una nueva línea de desarrollo en paralelo a otras ramas (Git Branch) existentes. Esto permite incorporar nuevas funcionalidades de manera ordenada y precisa, sin afectar la rama principal del proyecto. (Verificar Rama).

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git branch
* main
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> |
```

**Paso 2:** Escribimos el `git switch -c` que, para cambiar entre ramas de manera más sencilla y directa, de igual manera para crear las ramas que el antiguo comando `git checkout`, simplificando el proceso y evitando confusiones. (Crea las Ramas).

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git switch -c LorisB
Switched to a new branch 'LorisB'
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> |
```

**Paso 3:** Escribimos el comando `git push origin` enviarán la rama actual a la contraparte remota. Si se utilizan las configuraciones originales de Git, Git Push supone que la rama actual es la que se enviará y que la remota es la de origen. (Publicar

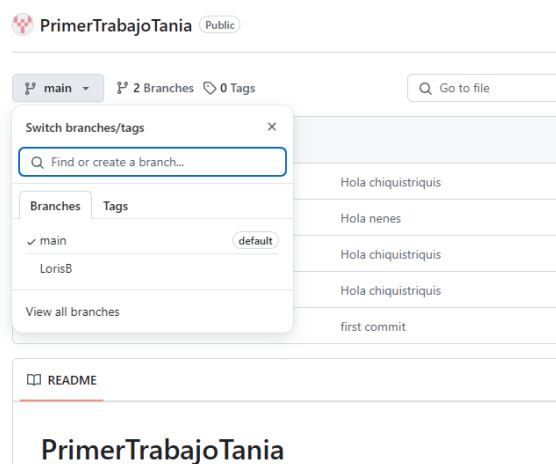
Ramas).

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git push origin LorisB
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'LorisB' on GitHub by visiting:
remote:   https://github.com/TANI1864/PrimerTrabajoTania/pull/new/LorisB
remote:
To https://github.com/TANI1864/PrimerTrabajoTania.git
 * [new branch]      LorisB -> LorisB
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania>
```

**Paso 4:** Se escribe el comando `git pull` main se utiliza para actualizar un repositorio local con los cambios más recientes de un repositorio remoto.

```
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Wamp-28/EjercicioJava1.git
 2d23f47..3cc6bbf  TaniaB -> TaniaB
PS C:\Users\Casa\IdeaProjects\Ejercicio\EjercicioJava1> git branch -D TaniaB
error: cannot delete branch 'TaniaB' used by worktree at 'C:/Users/Casa/IdeaProjects/Ejercicio/EjercicioJava1'
PS C:\Users\Casa\IdeaProjects\Ejercicio\EjercicioJava1> git pull origin main
remote: Enumerating objects: 198, done.
remote: Counting objects: 100% (76/76), done.
remote: Compressing objects: 100% (39/39), done.
```

**Paso 5:** Vamos al git Hub y refrescamos la página y nos debe aparecer nuestra rama.



## ¿COMO ELIMINAR LAS RAMAS?

**Paso 1:** Colocamos el comando `git Branch -D`

- **Elimina una rama local** que ya no es necesaria.
- **Forza la eliminación**, incluso si la rama tiene cambios sin fusionar (a diferencia de `git Branch -d`, que solo permite eliminar ramas ya fusionadas).
- **Limpia el entorno de trabajo** y evita acumulación de ramas innecesarias.

```
PS C:\Users\Casa\IdeaProjects\PruebaTrabajoEscrito> git switch main
A      .gitignore
A      .idea/.gitignore
A      .idea/misc.xml
A      .idea/modules.xml
A      .idea/vcs.xml
A      PruebaTrabajoEscrito.iml
A      src/Main.java
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Casa\IdeaProjects\PruebaTrabajoEscrito> git branch -D TanisB
Deleted branch TanisB (was 784b46d).
```

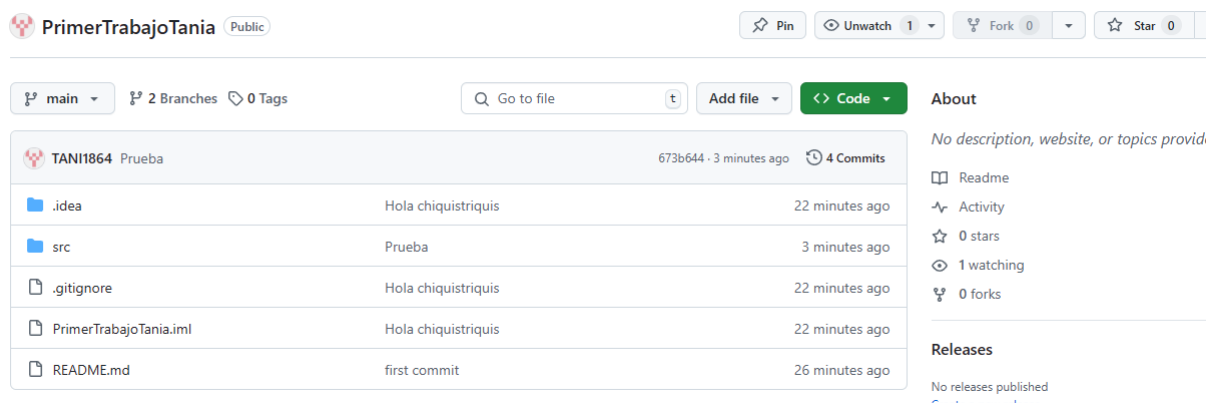
## ¿COMO CREAR COMMIT EN LAS RAMAS?

**Paso 1:** Empezamos poniendo nuestros comandos básicos, pero debemos tener en cuenta que estemos parados en la rama que necesitamos, para verificar usamos el comando git Branch.

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git status
On branch LorisB
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git branch
LorisB
* main
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> 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\Casa\IdeaProjects\PrimerTrabajoTania> git commint -c "Prueba"
git: 'commint' is not a git command. See 'git --help'.

The most similar command is
    commit
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git commit -m "Prueba"
[main 673b644] Prueba
 1 file changed, 1 insertion(+)
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 391 bytes | 391.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
```

**Paso 2:** Vamos al repositorio y refrescamos



The screenshot shows the GitHub interface for the repository 'PrimerTrabajoTania'. At the top, there are navigation buttons: Pin, Unwatch (1), Fork (0), and Star (0). Below this, the repository is shown with a dropdown menu for 'main' and '2 Branches'. A search bar and 'Add file' button are also visible. The main content area displays a list of files and their commit history:

File	Commit	Time
.idea	Hola chiquistruquis	22 minutes ago
src	Prueba	3 minutes ago
.gitignore	Hola chiquistruquis	22 minutes ago
PrimerTrabajoTania.iml	Hola chiquistruquis	22 minutes ago
README.md	first commit	26 minutes ago


On the right side, there is an 'About' section with links to Readme, Activity, Stars (0), Watching (1), and Forks (0). Below that is a 'Releases' section stating 'No releases published' with a link to 'Create a new release'.

**Paso 3:** Hacemos lo mismo con la segunda rama.

```

1 file changed, 1 insertion(+)
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git switch LorisB
Already on 'LorisB'
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git branch
* LorisB
  main
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git add .
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git commit -m "prueba"
On branch LorisB
nothing to commit, working tree clean
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git push origin LorisB
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/TANI1864/PrimerTrabajoTania.git
   c6ab4c9..673b644  LorisB -> LorisB
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania>


```


**PrimerTrabajoTania**
Public
Pin
Unwatch
1

🔗 LorisB
🔗 2 Branches
🔗 0 Tags

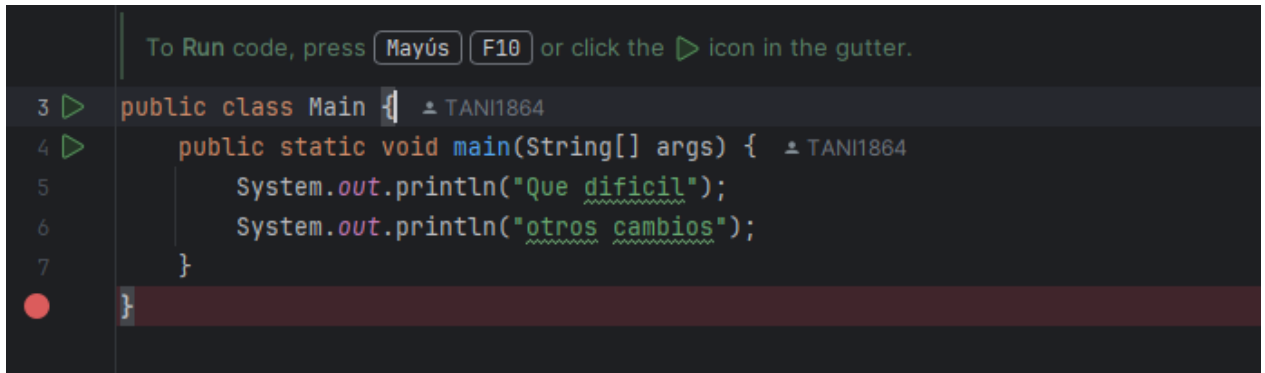
t
Add file
<> Code

This branch is up to date with main .
🔗 Contribute


 <b>TANI1864</b> Prueba	673b644 · 11 minutes ago	🕒 4 Commits
📁 .idea	Hola chiquistruis	30 minutes ago
📁 src	Prueba	11 minutes ago
📄 .gitignore	Hola chiquistruis	30 minutes ago
📄 PrimerTrabajoTania.iml	Hola chiquistruis	30 minutes ago
📄 README.md	first commit	34 minutes ago





## ¿COMO UTILIZAR GIT PULL ORIGIN RAMA?

**Paso 1:** Escribimos en la pantalla de código unas líneas



```

To Run code, press Mayús F10 or click the  icon in the gutter.

3  public class Main {  TANI1864
4      public static void main(String[] args) {  TANI1864
5         System.out.println("Que dificil");
6         System.out.println("otros cambios");
7     }
    }
  
```

**Paso 2:** Escribimos en la terminal el (git status, git add., git commint -m, git push originRama), Para realizar los cambios y subirlos.

```

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

PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git add .
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git commit -m "Cambiar cambios"
1 file changed, 2 insertions(+), 2 deletions(-)
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git push origin LorisTB
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 375 bytes | 375.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/TANI1864/PrimerTrabajoTania.git
  
```

**Paso 3:** Para cambiarnos de rama vamos a escribir **git switch** (Nombre de la rama) y con el **git Branch** verificamos que estemos en la rama correcta.

```

To https://github.com/TANI1864/PrimerTrabajoTania.git
673b644..c505691 LorisTB -> LorisTB
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git branch
LorisTB
* main

```

**Paso 4:** Después vamos a escribir un `git pull origin (Nombre de la rama)` se usa para descargar los cambios de una rama remota y fusionarlos con tu rama actual.

```

To https://github.com/TANI1864/PrimerTrabajoTania.git
673b644..c505691 LorisTB -> LorisTB
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git branch
LorisTB
* main
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git pull origin LorisTB
From https://github.com/TANI1864/PrimerTrabajoTania
* branch          LorisTB      -> FETCH_HEAD
Updating 673b644..c505691
Fast-forward
 src/Main.java | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> 

```



Paso 5: Refrescamos la página del repositorio

LorisTB3 Branches0 Tags

Go to file

Add file

Code

This branch is up to date with main.

Contribute

TANI1864Cambiar cambiosc505691 · 50 minutes ago5 Commits

.idea	Hola chiquistruquis	3 hours ago
src	Cambiar cambios	50 minutes ago
.gitignore	Hola chiquistruquis	3 hours ago
PrimerTrabajoTania.iml	Hola chiquistruquis	3 hours ago
README.md	first commit	3 hours ago

README

PrimerTrabajoTania

main3 Branches0 Tags

Go to file

Add file

Code

TANI1864Cambiar cambiosc505691 · 51 minutes ago5 Commits

.idea	Hola chiquistruquis	3 hours ago
src	Cambiar cambios	51 minutes ago
.gitignore	Hola chiquistruquis	3 hours ago
PrimerTrabajoTania.iml	Hola chiquistruquis	3 hours ago
README.md	first commit	3 hours ago

## COMANDOS

- ❖ El comando `git log --oneline nombre rama` muestra el historial de commits de forma **compacta y resumida**.

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git log --oneline "LorisTB"
673b644 (HEAD -> LorisTB, origin/main, origin/LorisTB, origin/LorisB, origin/HEAD, main) Prueba
c6ab4c9 Hola nenes
986bf12 Hola chiquistruquis
400b95b first commit
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania>
```

- ❖ El comando `git Branch -r` muestra todas las ramas remotas del repositorio, es decir, las ramas que existen en el servidor (GitHub, GitLab, Bitbucket, etc.), pero no necesariamente en tu repositorio local.

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git branch -r
origin/HEAD -> origin/main
origin/LorisB
origin/LorisTB
origin/main
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania>
```

- ❖ El `Git merge+ (--Rebase)` combina ramas con un commit de fusión

Escribimos `git merge <nombre rama>` y si estás en main y quieres fusionar LorisTB

- **Git merge** se usa para combinar cambios de una rama en otra. Se ejecuta desde la rama en la que quieres recibir los cambios.

- El comando `git checkout` se usa para Cambiar de rama en un repositorio y Restaurar archivos o commits anteriores.

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git merge feature
merge: feature - not something we can merge
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git merge Rama3
Updating c505691..07f2658
Fast-forward
 src/Main.java | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)
```

- ❖ `Git log` se usa para **ver el historial de commits** en un repositorio de Git.

Te muestra información sobre cada commit, incluyendo:

**Hash del commit** (identificador único), **Autor del commit**, **Fecha y**

**hora**, **Mensaje del commit**

```

PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git log
commit c5056916526097c8a8f992425ea1d21342064189 (HEAD -> main, origin/main, origin/LorisTB, origin/HEAD, LorisTB)
Author: TANI1864 <taniabeltran2020@gmail.com>
Date:   Fri Feb 21 00:27:10 2025 -0500

    Cambiar cambios

commit 673b644e712259c31900f5d992ddb80276ad6e80 (origin/LorisB)
Author: TANI1864 <taniabeltran2020@gmail.com>
Date:   Thu Feb 20 22:31:01 2025 -0500

    Prueba

commit c6ab4c9f2dfd72e3dc3c7e48a9c4eef70b4911a3
Author: TANI1864 <taniabeltran2020@gmail.com>
Date:   Thu Feb 20 22:17:59 2025 -0500

    Hola nenes

commit 986bf125f2b763410db87a8731734d91f0ac8f4a
Author: TANI1864 <taniabeltran2020@gmail.com>
Date:   Thu Feb 20 22:11:34 2025 -0500

```

❖ **git reflog** se usa para **ver el historial de referencias de HEAD y otras**

**ramas**, incluyendo cambios de posición en los commits, reversiones y fusiones.

Es muy útil cuando necesitas **recuperar commits perdidos**, ya que muestra movimientos de HEAD, incluso si un commit ya no está en una rama visible con git log.

```

PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git reflog
c505691 (HEAD -> main, origin/main, origin/LorisTB, origin/HEAD, LorisTB) HEAD@{0}: checkout: moving from Rama3 to main
07f2658 (origin/Rama3, Rama3) HEAD@{1}: commit: Se agrego algo nuevo
c505691 (HEAD -> main, origin/main, origin/LorisTB, origin/HEAD, LorisTB) HEAD@{2}: checkout: moving from main to Rama3
c505691 (HEAD -> main, origin/main, origin/LorisTB, origin/HEAD, LorisTB) HEAD@{3}: pull origin LorisTB: Fast-forward
673b644 (origin/LorisB) HEAD@{4}: checkout: moving from LorisTB to main
c505691 (HEAD -> main, origin/main, origin/LorisTB, origin/HEAD, LorisTB) HEAD@{5}: commit: Cambiar cambios
673b644 (origin/LorisB) HEAD@{6}: checkout: moving from main to LorisTB
673b644 (origin/LorisB) HEAD@{7}: checkout: moving from main to main
673b644 (origin/LorisB) HEAD@{8}: checkout: moving from LorisTB to main
673b644 (origin/LorisB) HEAD@{9}: checkout: moving from LorisB to LorisTB
673b644 (origin/LorisB) HEAD@{10}: checkout: moving from LorisB to LorisB
673b644 (origin/LorisB) HEAD@{11}: checkout: moving from LorisB to LorisB
673b644 (origin/LorisB) HEAD@{12}: pull origin main: Fast-forward
c6ab4c9 HEAD@{13}: checkout: moving from main to LorisB
673b644 (origin/LorisB) HEAD@{14}: checkout: moving from LorisB to main
c6ab4c9 HEAD@{15}: checkout: moving from main to LorisB
673b644 (origin/LorisB) HEAD@{16}: commit: Prueba
c6ab4c9 HEAD@{17}: checkout: moving from LorisB to main
c6ab4c9 HEAD@{18}: checkout: moving from main to LorisB
c6ab4c9 HEAD@{19}: commit: Hola nenes
986bf12 HEAD@{20}: commit: Hola chiquistriquis

```

❖ `git Branch-d` (Nombre de la Rama) se usa para **eliminar una rama**

**local** en Git, pero solo si ya ha sido fusionada con la rama actual.

**Paso 1:** No debemos estar parados en la rama que queremos eliminar

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git branch
LorisB
LorisTB
Rama3
* Rama4
main
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git switch Rama3
Switched to branch 'Rama3'
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git branch
LorisB
LorisTB
* Rama3
Rama4
main
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git branch -d Rama4
Deleted branch Rama4 (was c505691).
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania>
```

## Git fetch--all

Paso 1: Antes de hacer fetch, puedes revisar si tienes cambios sin confirmar:

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

You are currently reverting commit 4406e4c.
  (all conflicts fixed: run "git revert --continue")
  (use "git revert --skip" to skip this patch)
  (use "git restore --staged <file>..." to unstage)
    modified:   src/Main.java
```

Paso 2: Después escribimos el `git fetch--all` Esto descargará los cambios de **todas** las ramas de **todos** los remotos configurados en tu repositorio.

Si solo quieres actualizar un remoto específico (origin):

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git fetch --all
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git fetch origin
```

**Paso 3:** Ver qué cambios se descargaron

Después de ejecutar `git fetch --all`, puedes ver los commits nuevos que existen en las ramas remotas:

Escribimos el comando `git log --oneline --graph --all --decorate` Este comando en Git **muestra el historial de commits** de manera visual y simplificada, con información sobre ramas, etiquetas y relaciones entre commits. Es útil para entender la estructura del historial del repositorio y ver cómo las ramas se han fusionado o separado.

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git log --oneline --graph --all --decorate
* 9fc212a (HEAD -> main) Guardando cambios antes del revert
* 4406e4c (origin/main, origin/HEAD) Hola buenas tardes
| * 91db254 (origin/Rama3) Update Main.java
|/
* 07f2658 (Rama3) Se agrego algo nuevo
| * 1ec3a7b (origin/LorisTB) Update Main.java
|...skipping...
* 9fc212a (HEAD -> main) Guardando cambios antes del revert
* 4406e4c (origin/main, origin/HEAD) Hola buenas tardes
| * 91db254 (origin/Rama3) Update Main.java
|/
* 07f2658 (Rama3) Se agrego algo nuevo
| * 1ec3a7b (origin/LorisTB) Update Main.java
|/
* c505691 (origin/Rama4, LorisTB, LorisB) Cambiar cambios
* 673b644 (origin/LorisB) Prueba
* c6ab4c9 Hola nenes
* 986bf12 Hola chiquistriquis
|...skipping...
* 4406e4c (origin/main, origin/HEAD) Hola buenas tardes
| * 91db254 (origin/Rama3) Update Main.java
|/
```

**Paso 4:** Si quieres comparar tu código con la última versión remota escribimos el comando `git diff HEAD origin/main` Este comando **compara tu código local (HEAD) con el código más**

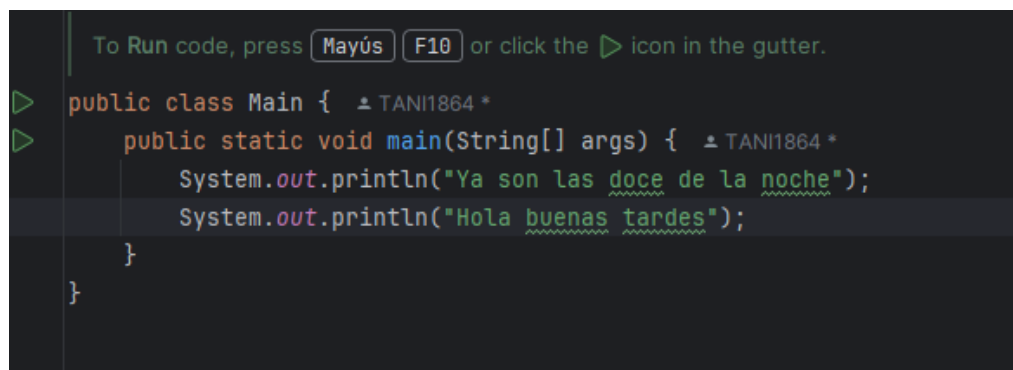
reciente en la rama remota (origin/main). Es útil para ver qué cambios tienes en tu repositorio local antes de hacer un git pull o un git merge.

```
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git diff HEAD origin/main
diff --git a/src/Main.java b/src/Main.java
index 7d98902..cd7aeac 100644
--- a/src/Main.java
+++ b/src/Main.java
@@ -3,6 +3,6 @@
 public class Main {
     public static void main(String[] args) {
         System.out.println("Ya son las doce de la noche");
-
+        System.out.println("Hola buenas tardes");
     }
 }
\ No newline at end of file
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania>
```

## Git reverse

Deshace un commit **creando un nuevo commit** que revierte los cambios del commit original.

**Paso 1:** Eliminamos la línea de código en la pantalla



The screenshot shows an IDE window with a Java file. The code is as follows:

```
To Run code, press Mayús F10 or click the ▶ icon in the gutter.
public class Main {
    public static void main(String[] args) {
        System.out.println("Ya son las doce de la noche");
        System.out.println("Hola buenas tardes");
    }
}
```

The line `System.out.println("Hola buenas tardes");` is highlighted with a mouse cursor, indicating it is about to be deleted.

**Paso 2:** En la terminal escribimos el comando `git log --oneline` para verificar todos los commit

**Paso 3:** Al encontrarlo escribimos el comando git revert con el código del commint, guardamos los cambios

**Paso 4:** Después escribimos el comando git HEAD es un **puntero especial** que indica en qué commit y rama estás trabajando actualmente.

Cuando ejecutas git log --oneline, el commit en el que está HEAD es el más reciente de tu rama. Y nos aparecerá de nueva la línea de código

```






no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git log --oneline
4406e4c (HEAD -> main, origin/main, origin/HEAD) Hola buenas tardes
673b644 (origin/LorisB) Prueba
c6ab4c9 Hola nenes
400b95b first commit
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git revert 4406e4c
error: Your local changes to the following files would be overwritten by merge:
  src/Main.java
Please commit your changes or stash them before you merge.
Aborting
fatal: revert failed
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git add .
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git commit -m "Guardando cambios antes del revert"
[main 9fc212a] Guardando cambios antes del revert
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git revert 4406e4c
Auto-merging src/Main.java
CONFLICT (content): Merge conflict in src/Main.java
error: could not revert 4406e4c... Hola buenas tardes
hint: "git add/rm <paths>", then run
hint: "git revert --continue".
hint: You can instead skip this commit with "git revert --skip".
hint: run "git revert --abort".
hint: Disable this message with "git config set advice.mergeConflict false"
PS C:\Users\Casa\IdeaProjects\PrimerTrabajoTania> git revert HEAD

```




## ¿COMO CLONAR UN ARCHIVO?

### Paso 1: Crear una carpeta en el WINDOWS

1.  WINDOWS (C:)
2.  Usuarios
3.  Casa
4.  IdeaProjects
5.  Tania Bel

En la siguiente carpeta damos clic derecho y damos en la opción open git  
Bash here

6.  Open Git Bash here


**Paso 2:** Abrimos una consola y escribimos en comando **git clone** se usa para **copiar un repositorio remoto a tu computadora**. Básicamente, descarga todo el contenido del repositorio (código, historial de commits, ramas, etc.) y lo convierte en una copia local en la que puedes trabajar.







```

MINGW64:/e/Bakup de informacion en tiempo real no Borrar/Escritorio/Tani...
Casa@DESKTOP-SOTEEB6 MINGW64 /e/Bakup de informacion en tiempo real no Borrar/Escritorio/Tania Bel
$ git clone https://github.com/TANI1864/prueba1
Cloning into 'prueba1'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 13 (delta 0), reused 13 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (13/13), done.
Casa@DESKTOP-SOTEEB6 MINGW64 /e/Bakup de informacion en tiempo real no Borrar/Escritorio/Tania Bel
$

```

**Paso 3:** Verificamos en la carpeta que hemos creado y nos debe aparecer la carpeta que clonamos, con archivos dentro.

 prueba1

Nombre	Fecha de modificación	tipo	tamaño
 .git	21/02/2025 12:04 a.m.	Carpeta de archivos	
 .idea	21/02/2025 12:04 a.m.	Carpeta de archivos	
 src	21/02/2025 12:04 a.m.	Carpeta de archivos	
 .gitignore	21/02/2025 12:02 a.m.	Documento de te...	1 KB
 clonar.iml	21/02/2025 12:02 a.m.	Archivo IML	1 KB
 README	21/02/2025 12:02 a.m.	Archivo de origen ...	1 KB

## CONCLUSIONES

El uso conjunto de IntelliJ IDEA y GitHub representa una solución eficaz y robusta para el desarrollo de software, ya que permite a los programadores administrar su código de manera estructurada y colaborativa. IntelliJ ofrece un entorno de desarrollo intuitivo con funciones avanzadas de integración con Git, lo que simplifica el control de versiones, mejora el trabajo en equipo y agiliza la automatización de tareas.

## BIBLIOGRAFIA

*IntelliJ IDEA overview | IntelliJ IDEA.* (s. f.). IntelliJ IDEA

Help. <https://www.jetbrains.com/help/idea/discover-intellij-idea.html#IntelliJ-IDEA-editions>

*Acerca de GitHub y Git - documentación de GitHub.* (s. f.). GitHub

Docs. <https://docs.github.com/es/get-started/start-your-journey/about-github-and-git>

Perez, C. (2021, 23 abril). *El comando git push explicado.*

freeCodeCamp.org. <https://www.freecodecamp.org/espanol/news/el-comando-git-push-explicado/>