

PRIMER TALLER DE LINEA 3

ALUMNO:

MIGUEL ANGEL MEJIA SOSA

PROFESOR:

WILLIAM ALEXANDER MATALLANA PORRAS

UNIVERSIDAD DE CUNDINAMARCA  
INGENIERIA DE SISTEMAS  
EXTENSION CHIA  
AÑO 2025

Git config – -list-----	2
Git config – -global user.email-----	2
Git config – -global user.name-----	3
Git config - - global - -unset user.email-----	4
Git config - - global - -unset user.name-----	5
Git init-----	6
Git clone-----	7
Git status-----	8
Git add .-----	8
Git commit -m “”-----	9
Git push origin rama-----	10
Git pull origin rama-----	11
Git branch -D rama-----	12
Git swtich -c rama-----	13
Git switch rama-----	13
Git fetch - -all -----	14
Git Branch -r-----	14
Git remote -v-----	15
Git log-----	16
Git reflog-----	16
Git log –oneline-----	17
Git merge-----	17
Git revert-----	18
Git push origin - - delete rama-----	18
Conclusiones// referencia-----	19

## **INTRODUCCION**

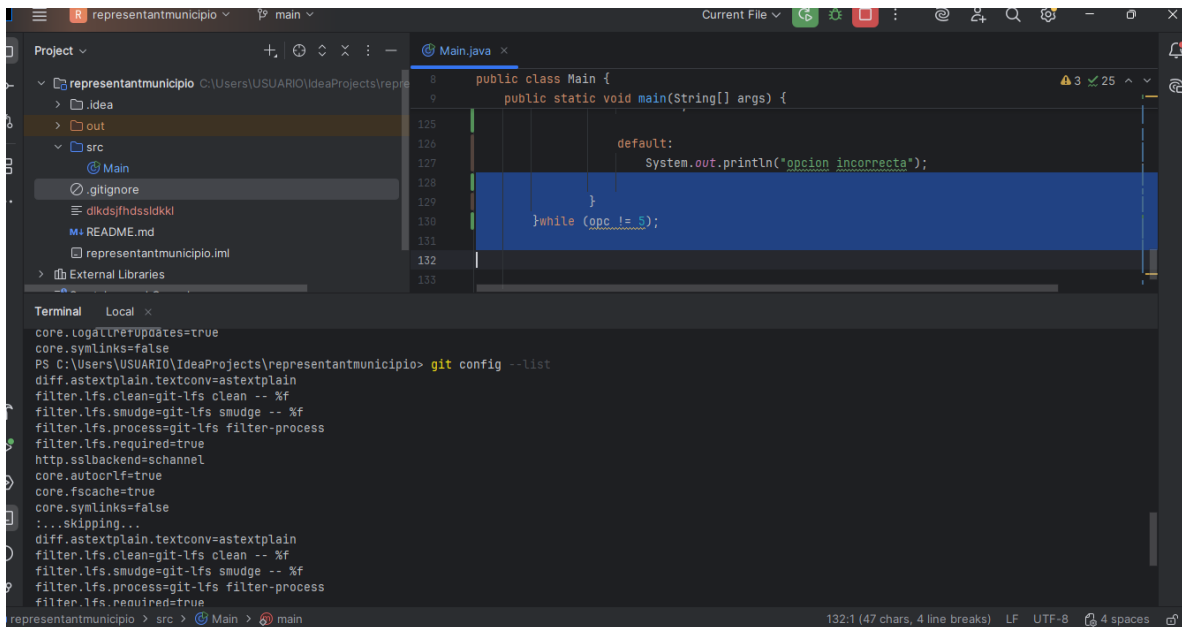
A continuación encontrara una lista de comandos de git fundamentales los cuales nos explican la importancia y el para que utilizarlos y también una ilustración de que es lo normal que puede salir a continuación encontrara la oportunidad de cambiar la forma de compartir y hacer trabajos en grupo

### **Objetivo :**

Mejorar el conocimiento básicos de compartir e interactuar con código en vivo

## Git config – -list

El comando git config se ha utilizado en casi todos los capítulos del libro. En Configurando Git por primera vez lo usamos para especificar nuestro nombre, dirección de correo electrónico y editor de preferencia antes incluso de comenzar a utilizar Git



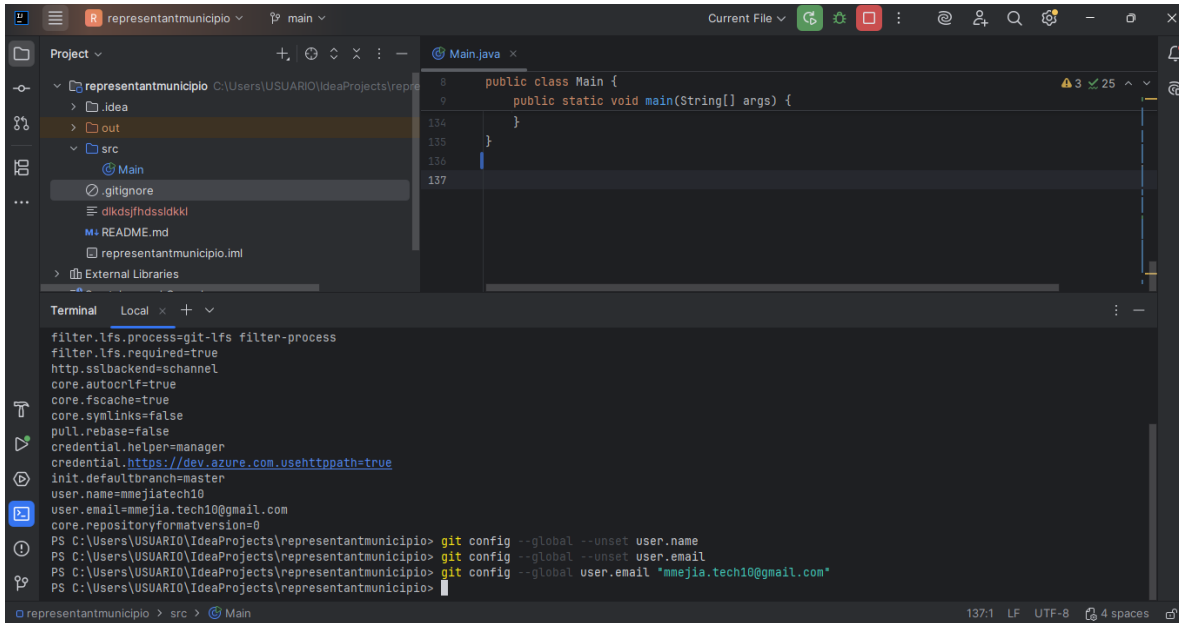
The screenshot shows an IDE with a project named 'representantmunicipio'. The 'Project' view on the left shows the file structure: 'representantmunicipio' (containing '.idea', 'out', 'src', 'Main', '.gitignore', 'dtkdsjfhdsldkkl', 'README.md', and 'representantmunicipio.iml') and 'External Libraries'. The 'Main.java' file is open in the editor, showing a Java class with a 'main' method. The terminal window at the bottom displays the output of the 'git config --list' command, showing various Git configuration settings.

```
public class Main {
    public static void main(String[] args) {
        default:
            System.out.println("opcion incorrecta");
    }
}while (opc != 5);
```

```
core.logallrefupdates=true
core.symlinks=false
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> 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=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
...skipping...
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
```

## Git config – -global user.email

Este comando es muy útil para verificar que la información de usuario y correo electrónico esté correctamente configurada. En caso de que quieras ver más detalles de alguna configuración en particular, puedes utilizar el comando "git config", donde es el nombre de la configuración que deseas ver.



The screenshot shows an IDE window with a project named 'representantmunicipio'. The left sidebar shows the project structure with files like '.idea', 'out', 'src', 'Main', '.gitignore', 'dtkdsjfhdsldkkl', 'README.md', and 'representantmunicipio.iml'. The main editor shows a Java file 'Main.java' with the following code:

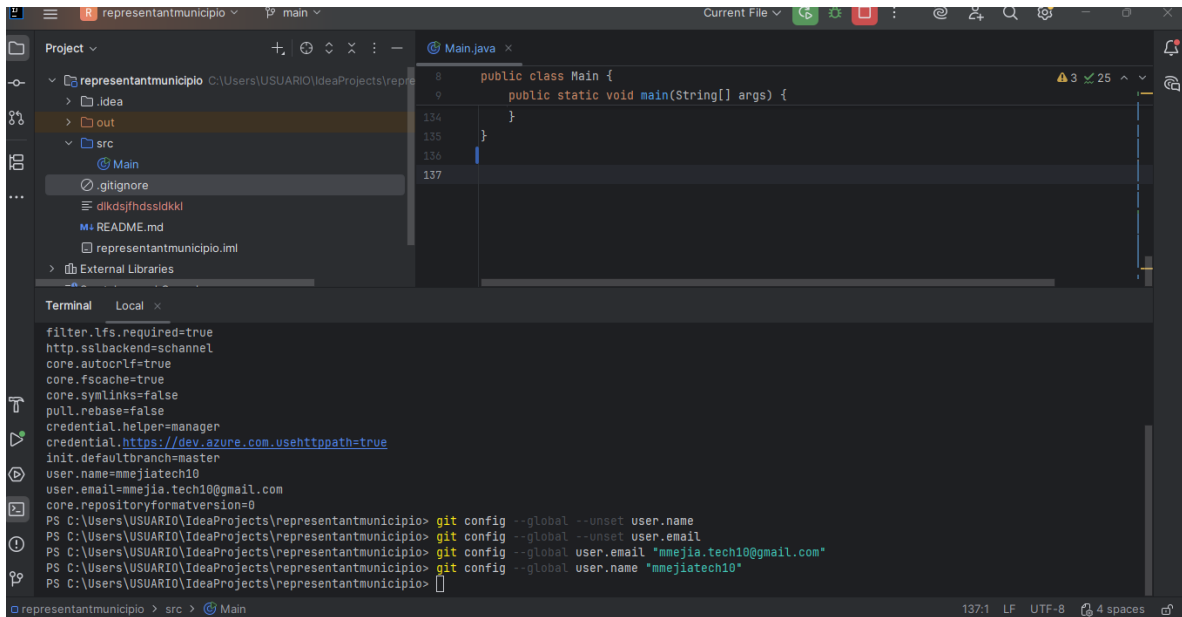
```
8 public class Main {
9     public static void main(String[] args) {
134     }
135 }
136
137
```

The bottom terminal window shows the output of the following commands:

```
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
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.name=mmejia.tech10
user.email=mmejia.tech10@gmail.com
core.repositoryformatversion=0
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git config --global --unset user.name
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git config --global --unset user.email
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git config --global user.email "mmejia.tech10@gmail.com"
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git config – -global user.name

El comando Git config se usa para configurar la configuración de usuario de Git para el proyecto o globalmente. Al iniciar el proyecto, asegúrese de configurar correctamente los valores de user.name y user. email. El comando "git config" también se puede usar para configurar esta configuración globalmente.



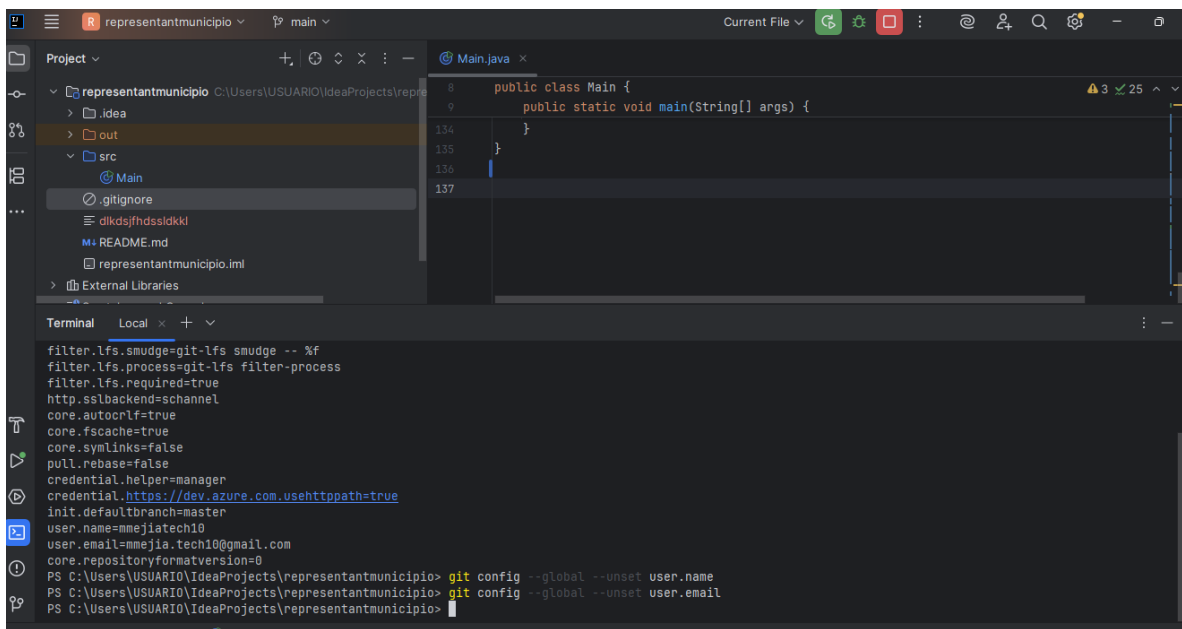
The screenshot shows an IDE with a project named 'representantmunicipio'. The 'Main.java' file is open, showing a simple Java class with a 'main' method. The terminal window at the bottom displays the output of several 'git config' commands:

```
filter.lfs.required=true
http.sslbackend=schannel
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.name=mmejiatech10
user.email=mmejia.tech10@gmail.com
core.repositoryformatversion=0
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git config --global --unset user.name
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git config --global --unset user.email
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git config --global user.email "mmejia.tech10@gmail.com"
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git config --global user.name "mmejiatech10"
```

## Git config - - global - -unset user.email

### Borrar email

Configuración por primera vez Ahora que tienes Git en tu sistema, vas a querer hacer algunas cosas para personalizar tu entorno de Git. Es neceso hacer estas cosas solamente una vez en tu computadora, y se mantendrán entre actualizaciones. También puedes cambiarlas en cualquier momento volviendo a ejecutar los comandos correspondientes.

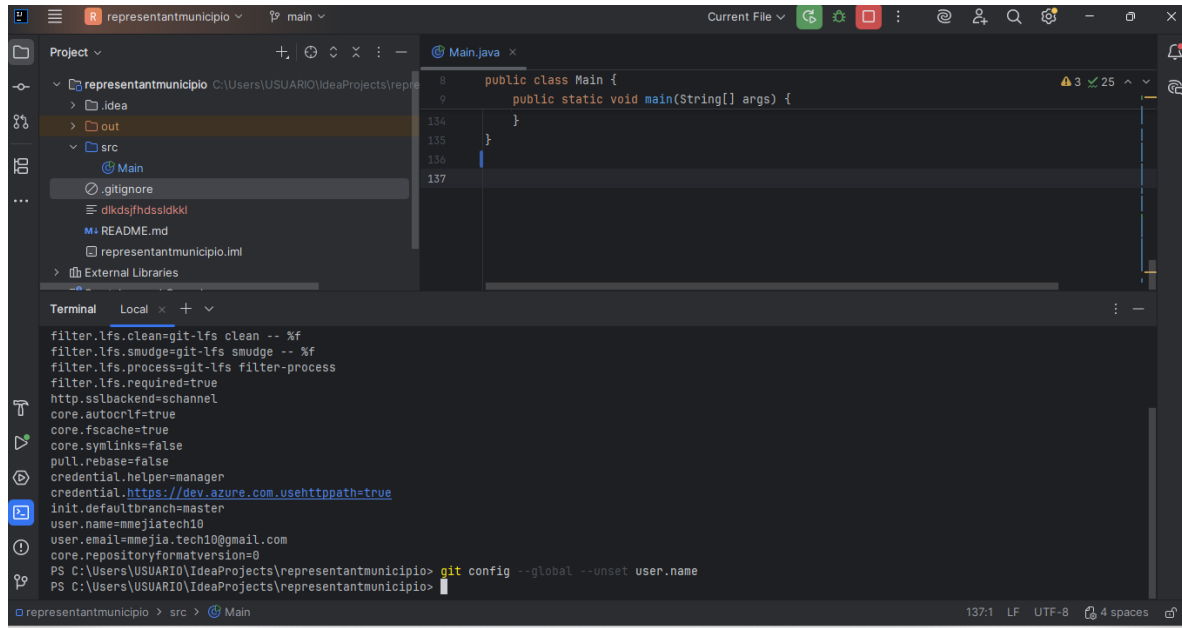


The screenshot shows the same IDE as before, but the terminal window now displays the output of the 'git config' commands, including the 'unset' commands:

```
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
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.name=mmejiatech10
user.email=mmejia.tech10@gmail.com
core.repositoryformatversion=0
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git config --global --unset user.name
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git config --global --unset user.email
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git config - - global - -unset user.name

Borrar nombre



The screenshot shows an IDE with a project named 'representantmunicipio'. The left sidebar shows the project structure with folders 'idea', 'out', 'src', and files '.gitignore', 'README.md', and 'representantmunicipio.iml'. The main editor shows a Java file 'Main.java' with the following code:

```
8 public class Main {
9     public static void main(String[] args) {
134     }
135 }
136
137
```

The bottom terminal window shows the output of the command 'git config --global --unset user.name'.

```
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=schannel
core.autocrlf=true
core.fsmonitor=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=mmejia.tech10
user.email=mmejia.tech10@gmail.com
core.repositoryformatversion=0
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git config --global --unset user.name
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git init

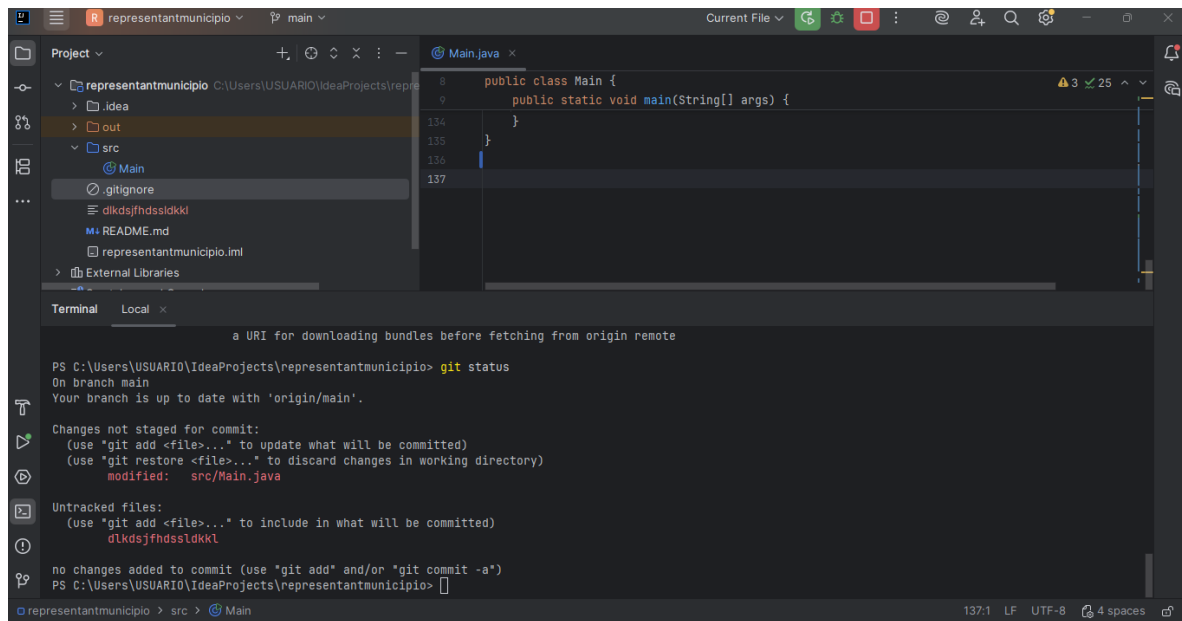
inicializa un nuevo repositorio de Git en un directorio de proyecto existente o crea un nuevo directorio con el repositorio. Esto se logra creando una carpeta oculta .git que contiene los metadatos, la estructura y el historial de control de versiones necesarios para rastrear los cambios en el proyecto

## Git clone

git clone crea una copia local de un repositorio remoto, descargando todo su historial y permitiéndote trabajar con los archivos en tu máquina local. Este proceso crea un nuevo directorio con los archivos del proyecto y configura automáticamente una conexión llamada "origin" al repositorio original, facilitando la interacción futura.

## Git status

git status muestra el estado actual de tu proyecto en el repositorio Git, informándote sobre qué archivos han sido modificados, cuáles están listos para ser confirmados (staged), cuáles no se han añadido aún al área de preparación (staged), y cuáles no están siendo rastreados por Git. Este comando es esencial para entender los cambios pendientes antes de realizar un commit.



The screenshot shows an IDE with a project named 'representantmunicipio'. The terminal window displays the output of the 'git status' command. The output indicates that the branch 'main' is up to date with 'origin/main'. It lists changes not staged for commit, including a modified file 'src/Main.java'. It also lists untracked files, including 'dLkdsjfhdsldkkl'. The IDE interface shows the project structure on the left, the source code of 'Main.java' in the center, and the terminal output at the bottom.

```
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> 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

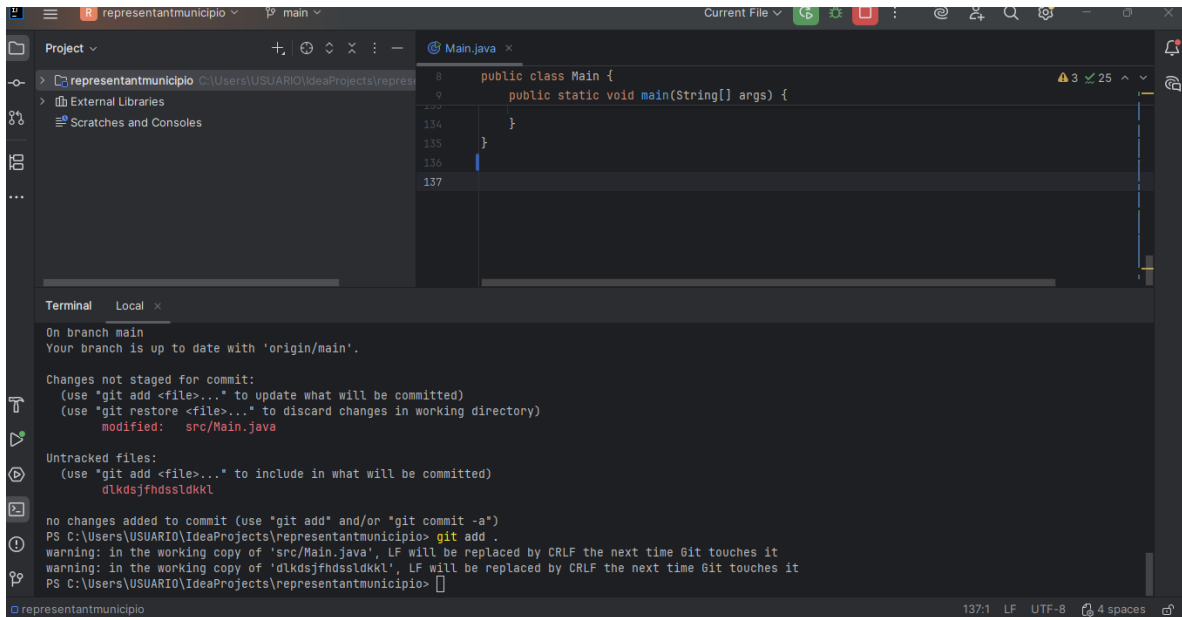
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        dLkdsjfhdsldkkl

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## git add .

en Git prepara para la próxima confirmación (commit) todos los cambios en el directorio actual y sus subdirectorios. En otras palabras, mueve las modificaciones, adiciones y eliminaciones de archivos desde tu directorio de trabajo al "área de ensayo" o "staging area", que es el siguiente paso antes de guardar esos cambios de forma permanente en el historial de tu proyecto





The screenshot shows the IntelliJ IDEA interface. The top pane displays the 'Main.java' file with the following code:

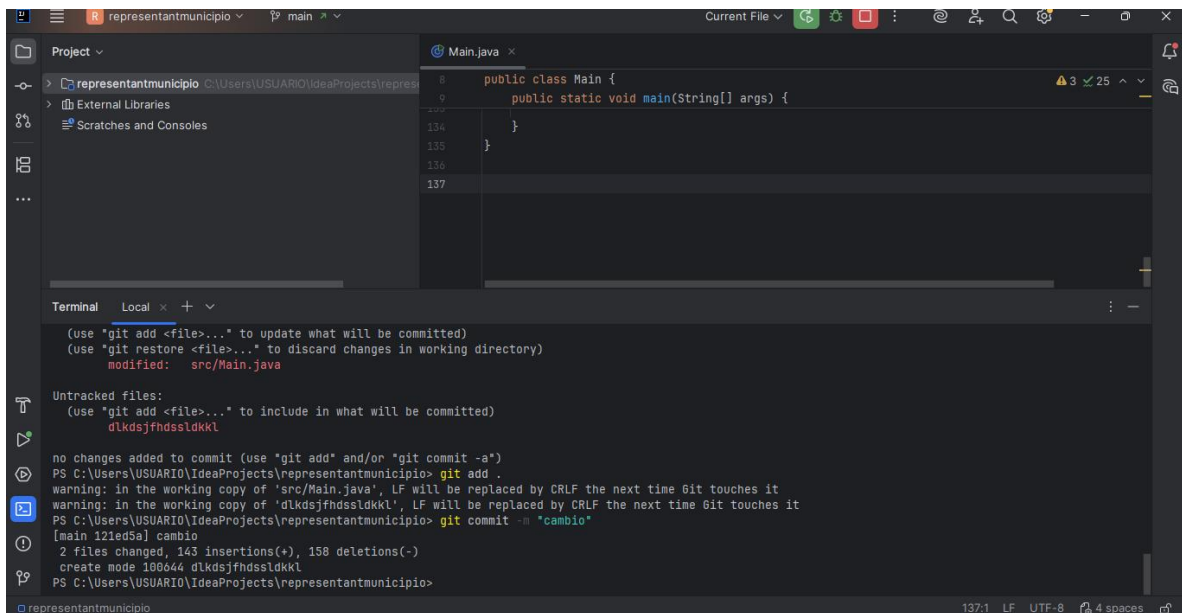
```
public class Main {  
    public static void main(String[] args) {  
    }  
}
```

The bottom pane shows the 'Terminal' window with the following output:

```
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  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
        dlkdsjfhdsldkkl  
  
no changes added to commit (use "git add" and/or "git commit -a")  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git add .  
warning: in the working copy of 'src/Main.java', LF will be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'dlkdsjfhdsldkkl', LF will be replaced by CRLF the next time Git touches it  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git commit -m ""

registra los archivos dentro del área de preparación como una nueva confirmación en el historial de actualización del repositorio. git push/pull – Sincroniza los cambios desde un repositorio local a un repositorio remoto



The screenshot shows the IntelliJ IDEA interface after a git commit. The top pane displays the 'Main.java' file with the same code as before:

```
public class Main {  
    public static void main(String[] args) {  
    }  
}
```

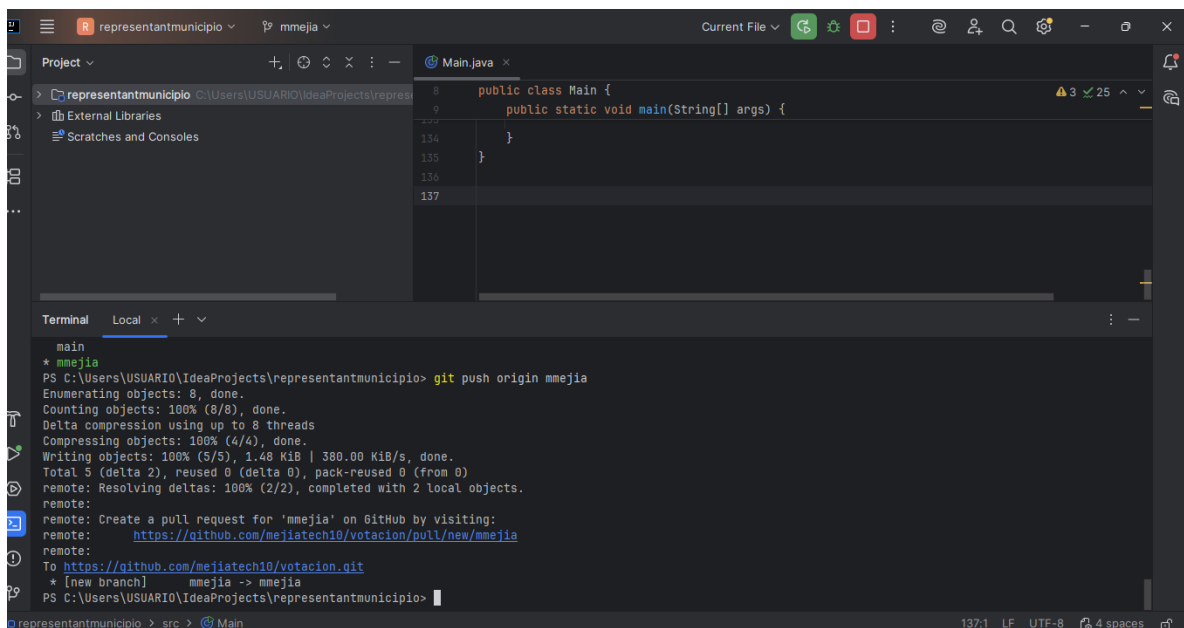
The bottom pane shows the 'Terminal' window with the following output:

```
(use "git add <file>..." to update what will be committed)  
(use "git restore <file>..." to discard changes in working directory)  
        modified:   src/Main.java  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
        dlkdsjfhdsldkkl  
  
no changes added to commit (use "git add" and/or "git commit -a")  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git add .  
warning: in the working copy of 'src/Main.java', LF will be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'dlkdsjfhdsldkkl', LF will be replaced by CRLF the next time Git touches it  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git commit -m "cambio"  
[main 121ed5a] cambio  
 2 files changed, 143 insertions(+), 158 deletions(-)  
 create mode 100644 dlkdsjfhdsldkkl  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git push origin rama

El comando `git push origin rama` sirve para subir los cambios de una rama local a un repositorio remoto llamado origin.

- `git push`: Es el comando que envía commits desde tu repositorio local al remoto.
- `origin`: Es el nombre por defecto del repositorio remoto desde el que clonaste tu proyecto (como GitHub).
- `rama`: Es el nombre de la rama que quieres subir.

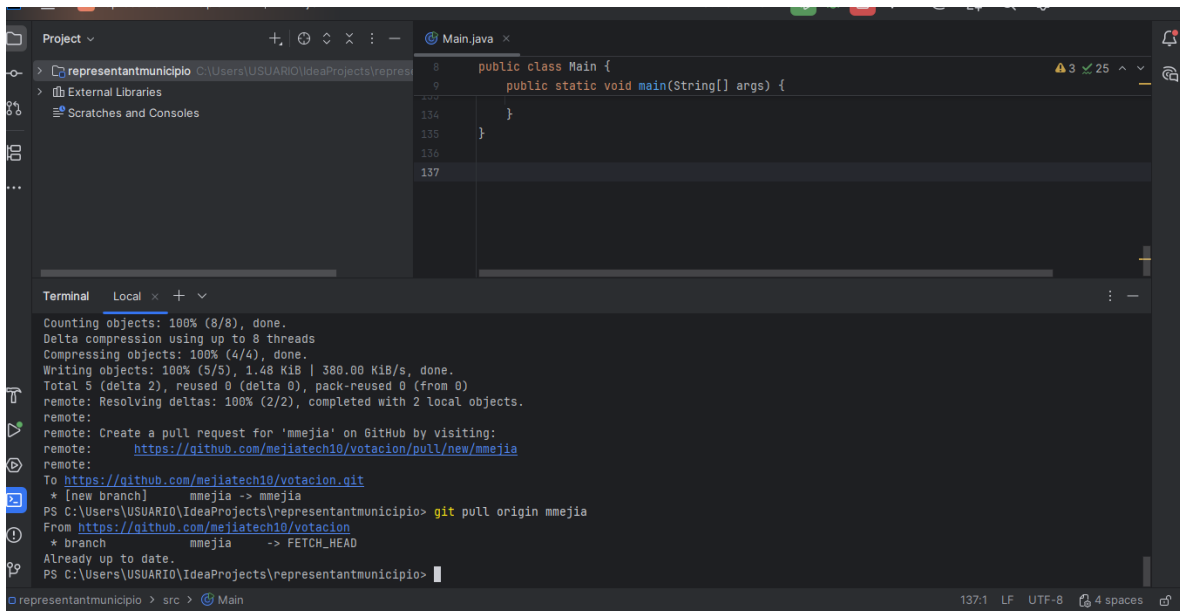


The screenshot shows an IDE with a project named 'representantmunicipio'. The 'Main.java' file is open, showing a simple Java class with a `main` method. Below the editor, the terminal window displays the output of the `git push origin mmejia` command. The output indicates that the push was successful, with 5 objects written and 2 local objects completed. It also provides a link to create a pull request on GitHub.

```
main
* mmejia
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git push origin mmejia
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.48 KiB | 380.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'mmejia' on GitHub by visiting:
remote:   https://github.com/mejiatech10/votacion/pull/new/mmejia
remote:
To https://github.com/mejiatech10/votacion.git
 * [new branch]      mmejia -> mmejia
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git pull origin rama

El comando `git pull origin rama` extrae los cambios del repositorio remoto llamado origin de la rama especificada (rama) y los fusiona automáticamente con tu rama local actual, sincronizando tu copia local del proyecto con las actualizaciones remotas. En esencia, actualiza tu repositorio local con el contenido más reciente disponible en el repositorio remoto en la rama indicada.



The screenshot shows an IDE with a project named 'representantmunicipio'. The main editor displays a Java file 'Main.java' with the following code:

```
8 public class Main {
9     public static void main(String[] args) {
10
11     }
12 }
```

The terminal window shows the output of a 'git pull' command:

```
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.48 KiB | 380.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'mmejia' on GitHub by visiting:
remote:   https://github.com/mejiatech10/votacion/pull/new/mmejia
remote:
To https://github.com/mejiatech10/votacion.git
 * [new branch]      mmejia -> mmejia
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git pull origin mmejia
From https://github.com/mejiatech10/votacion
 * branch            mmejia      -> FETCH_HEAD
Already up to date.
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git branch -D rama

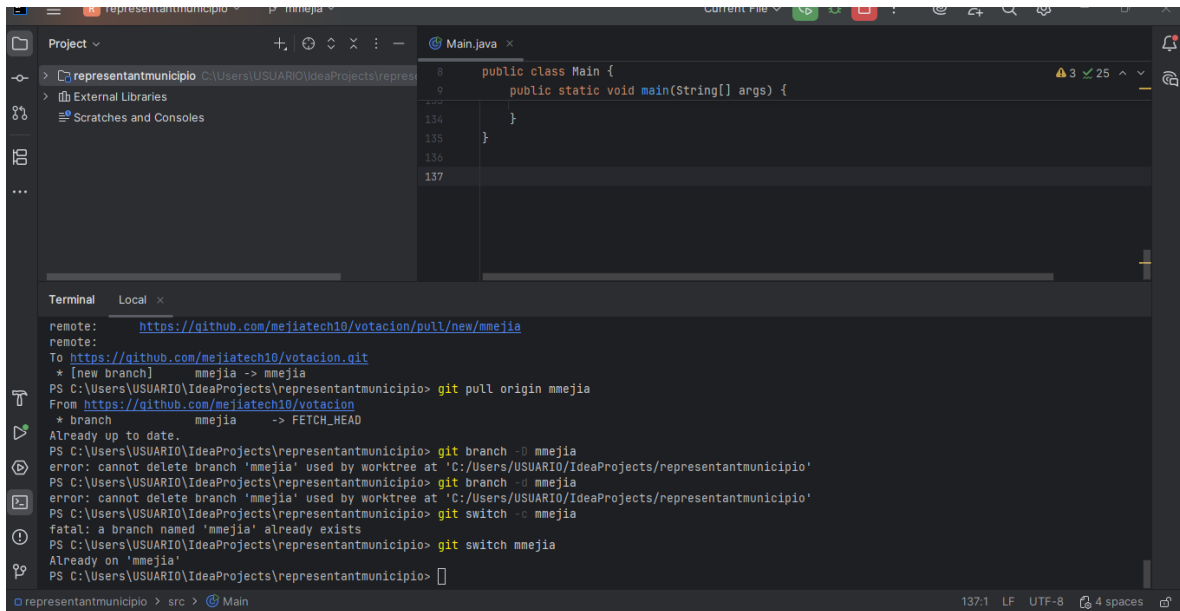
El comando `git branch` permite crear una rama nueva. Si quieres empezar a trabajar en una nueva función, puedes crear una rama nueva a partir de la rama `main` con `git branch new_branch`. Una vez creada, puedes usar `git checkout new_branch` para cambiar a esa rama.

## Git switch -c rama

En versiones recientes de Git, se creó un comando `git switch` cuyo único propósito es crear copias locales de ramas remotas y cambiar entre ramas locales

## Git switch rama

El comando `git switch rama` sirve para cambiar la rama HEAD actual a la especificada, actualizando el árbol de trabajo para que coincida con ella. Introducido en Git v2.23, simplifica el proceso de gestión de ramas en comparación con el antiguo comando `git checkout`, que tenía múltiples funciones. `git switch` se centra únicamente en el cambio de ramas y ofrece mayor seguridad y claridad al evitar operaciones que podrían llevar a la pérdida de datos



The screenshot shows an IDE with a project named 'representantmunicipio'. The main editor displays a Java file 'Main.java' with the following code:

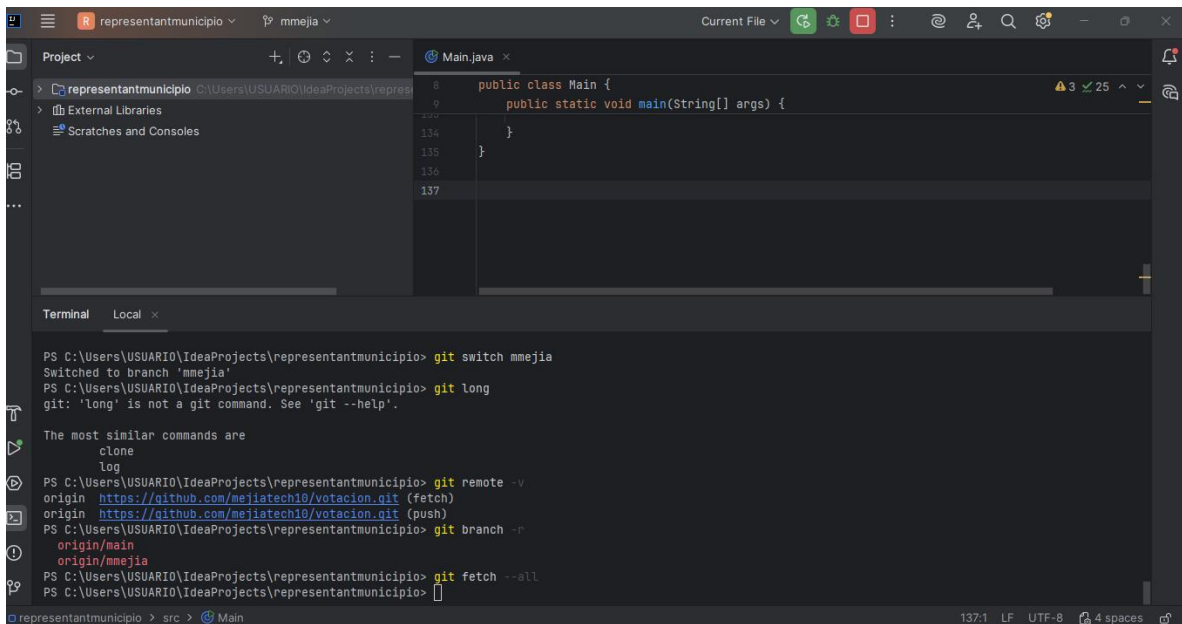
```
public class Main {  
    public static void main(String[] args) {  
  
    }  
}
```

The terminal window shows the following commands and output:

```
remote: https://github.com/mejiatech10/votacion/pull/new/mmejia  
remote:  
To https://github.com/mejiatech10/votacion.git  
* [new branch] mmejia -> mmejia  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git pull origin mmejia  
From https://github.com/mejiatech10/votacion  
* branch mmejia -> FETCH_HEAD  
Already up to date.  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git branch -d mmejia  
error: cannot delete branch 'mmejia' used by worktree at 'C:/Users/USUARIO/IdeaProjects/representantmunicipio'  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git branch -d mmejia  
error: cannot delete branch 'mmejia' used by worktree at 'C:/Users/USUARIO/IdeaProjects/representantmunicipio'  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git switch -c mmejia  
fatal: a branch named 'mmejia' already exists  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git switch mmejia  
Already on 'mmejia'  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git fetch -all

El comando `git fetch` comunica con un repositorio remoto y obtiene toda la información que se encuentra en ese repositorio que no está en el tuyo actual y la almacena en tu base de datos local. En primer lugar, observamos este comando en Traer y Combinar Remotos y seguimos viendo ejemplos de su uso en Ramas Remota



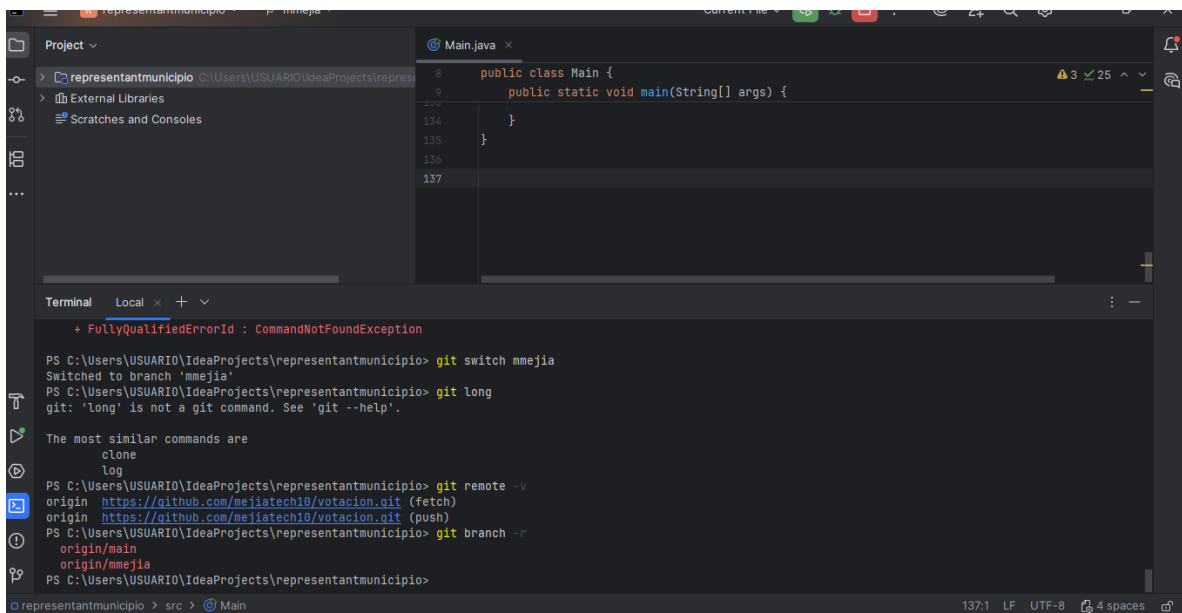
The screenshot shows an IDE with a project named 'representantmunicipio'. The 'Main.java' file is open, showing a simple Java class with a 'main' method. The terminal window is active, displaying the following commands and output:

```
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git switch mmejia
Switched to branch 'mmejia'
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git long
git: 'long' is not a git command. See 'git --help'.

The most similar commands are
    clone
    log
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git remote -v
origin  https://github.com/mejiatech10/votacion.git (fetch)
origin  https://github.com/mejiatech10/votacion.git (push)
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git branch -r
origin/main
origin/mmejia
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git fetch --all
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git Branch -r

El uso de las ramas de desarrollo de Git es una excelente manera de trabajar con una aplicación mientras rastreas sus versiones. En general, una rama de desarrollo (“Git Branch”) es una bifurcación del estado del código que crea un nuevo camino para la evolución del mismo



The screenshot shows the same IDE as before, but the terminal window now displays an error message:

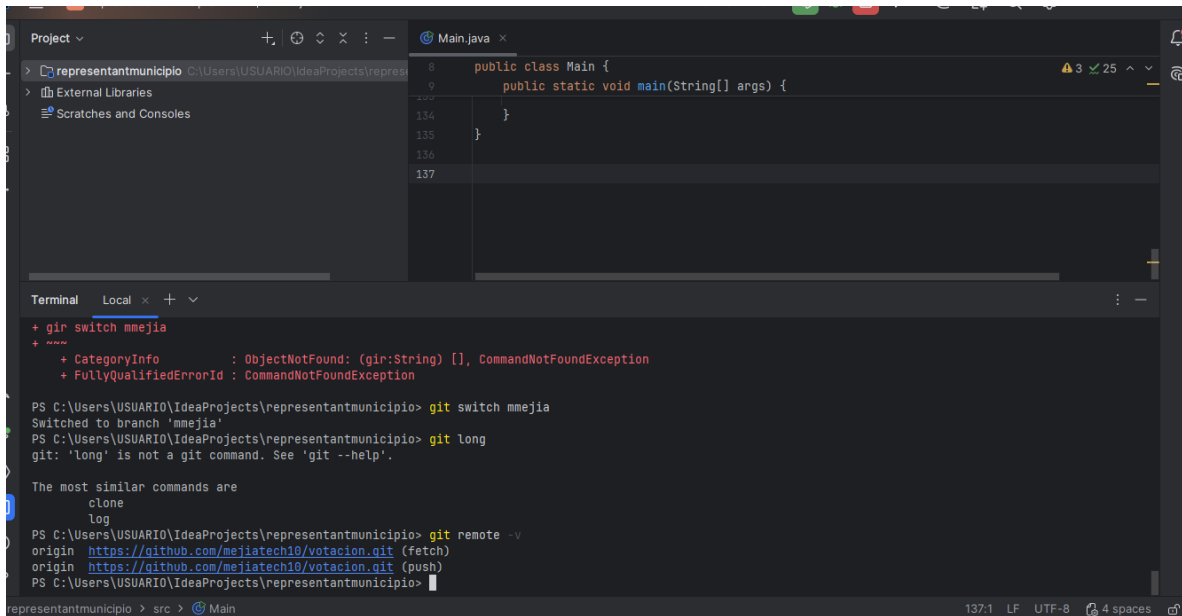
```
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git switch mmejia
Switched to branch 'mmejia'
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git long
git: 'long' is not a git command. See 'git --help'.

The most similar commands are
    clone
    log
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git remote -v
origin  https://github.com/mejiatech10/votacion.git (fetch)
origin  https://github.com/mejiatech10/votacion.git (push)
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git branch -r
origin/main
origin/mmejia
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git remote -v

Es un comando útil para administrar conexiones remotas. En lugar de pasar la URL completa a los comandos fetch, pull y push, permite usar un atajo más significativo



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

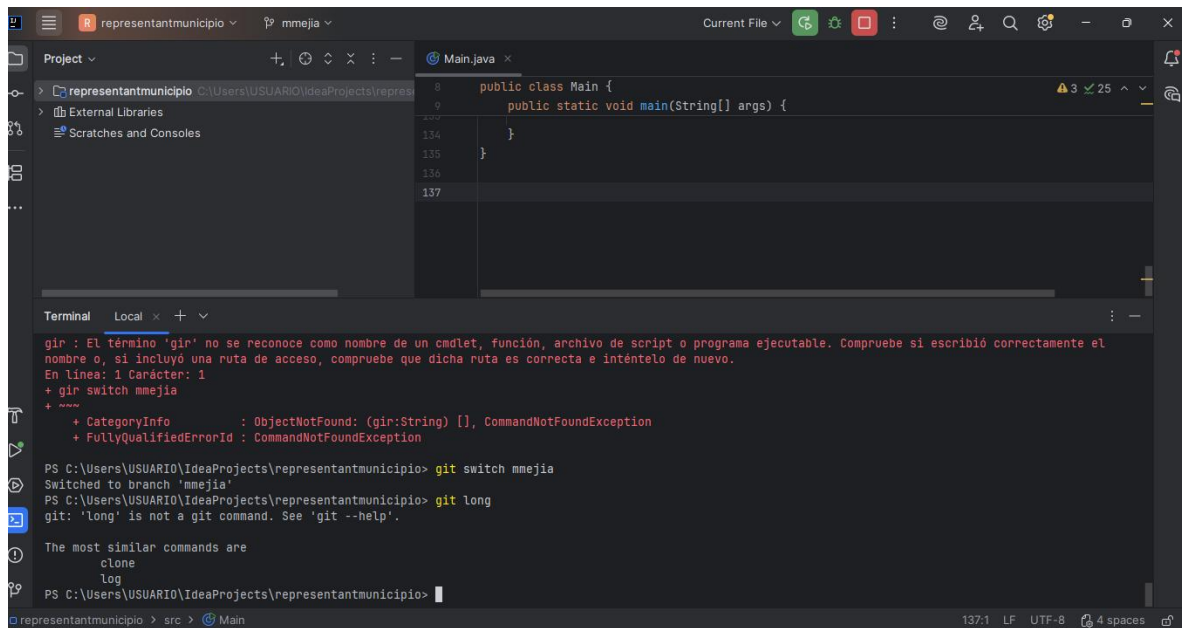
```
+ gir switch mmejia
+ CategoryInfo          : ObjectNotFound: (gir:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git switch mmejia
Switched to branch 'mmejia'
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git long
git: 'long' is not a git command. See 'git --help'.

The most similar commands are
    clone
    log
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git remote -v
origin  https://github.com/mejiatech10/votacion.git (fetch)
origin  https://github.com/mejiatech10/votacion.git (push)
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git log

git log muestra el historial de confirmaciones (commits) de un repositorio Git, permitiendo listar, filtrar y buscar cambios específicos en el proyecto. De forma predeterminada, muestra los commits en orden cronológico inverso, incluyendo el hash único del commit, el autor, la fecha y el mensaje de cada cambio. Este comando es esencial para entender la evolución de un proyecto y rastrear modificaciones realizadas



The screenshot shows an IDE window with a project named 'representantmunicipio'. The main editor displays a Java file 'Main.java' with the following code:

```
public class Main {  
    public static void main(String[] args) {  
    }  
}
```

The terminal window shows the following commands and output:

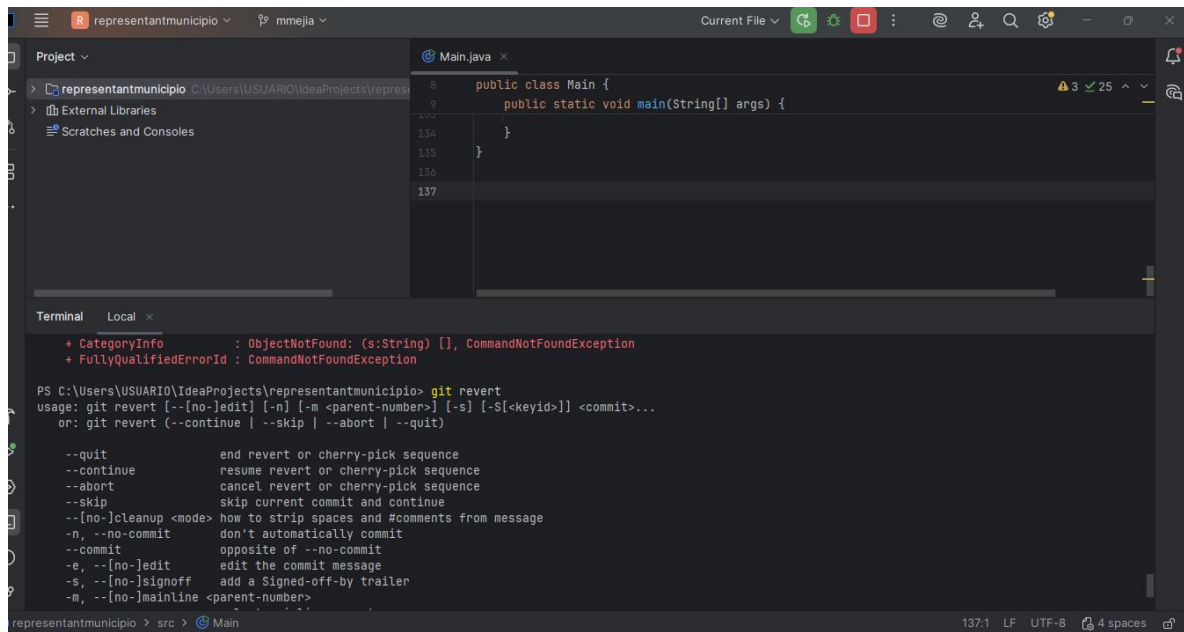
```
gir : El término 'gir' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e inténtelo de nuevo.  
En línea: 1 Carácter: 1  
+ gir switch mmejia  
+ ~~~~~  
+ CategoryInfo          : ObjectNotFound: (gir:String) [], CommandNotFoundException  
+ FullyQualifiedErrorId : CommandNotFoundException  
  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git switch mmejia  
Switched to branch 'mmejia'  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git long  
git: 'long' is not a git command. See 'git --help'.  
  
The most similar commands are  
clone  
log  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

## Git merge

El comando git merge en Git se utiliza para integrar los cambios de una rama en otra, creando un nuevo commit que une el historial de desarrollo de ambas ramas. Es la forma principal de Git para recomponer líneas de desarrollo bifurcadas, combinando el trabajo realizado en una rama independiente (la rama fuente) en la rama actual (la rama destino)

## Git revert

El comando git revert en Git sirve para deshacer los cambios de una confirmación (commit) específica creando una nueva confirmación que introduce el efecto contrario a los cambios anteriores. A diferencia de git reset, git revert no elimina la confirmación original del historial, sino que preserva el registro histórico y crea una nueva entrada que anula los cambios de la confirmación objetivo



The screenshot shows an IDE window with a project named 'representantmunicipio'. The main editor displays a Java file 'Main.java' with the following code:

```
8 public class Main {
9     public static void main(String[] args) {
10
11     }
12 }
```

The terminal window shows the command 'git revert' and its usage:

```
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git revert
usage: git revert [--no]edit [-n] [-m <parent-number>] [-s] [-S[<keyid>]] <commit>...
or: git revert [--continue | --skip | --abort | --quit]

--quit          end revert or cherry-pick sequence
--continue      resume revert or cherry-pick sequence
--abort         cancel revert or cherry-pick sequence
--skip         skip current commit and continue
--[no]cleanup <mode> how to strip spaces and #comments from message
-n, --no-commit don't automatically commit
--commit       opposite of --no-commit
-e, --[no]edit  edit the commit message
-s, --[no]signoff add a Signed-off-by trailer
-m, --[no]mainline <parent-number>
```

## Git push origin - - delete rama

El comando `git push origin --delete <rama>` se usa para eliminar una rama específica en el repositorio remoto. Desencadena un envío al servidor remoto con la instrucción de borrar la rama que se haya especificado, pero la rama local en tu máquina no se verá afectada a menos que también la borres por separad



The screenshot shows an IDE with a project named 'representantmunicipio'. The file explorer on the left shows the project structure: 'out' (production, representantmunicipio), 'src' (Main), and 'README.md'. The editor displays 'Main.java' with the following code:

```
8 public class Main {
9     public static void main(String[] args) {
10
11     }
12 }
```

The terminal window shows the following commands and output:

```
--[no-]reference use the 'reference' format to refer to commits
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git merge
fatal: No remote for the current branch.
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git branch
main
* mmejia
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git add .
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git switch main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git branch
* main
mmejia
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git add .
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> 
```

The screenshot shows the same IDE with the 'representantmunicipio' project. The file explorer and editor are the same as in the first screenshot. The terminal window shows the following commands and output:

```
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git switch main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git branch
* main
mmejia
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git add .
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git commit -m
error: switch 'm' requires a value
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git commit -m "cambio"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> 
```

A small window titled 'Herramienta Recortes' is visible in the bottom right corner of the terminal area.

The screenshot shows an IDE window titled 'representantmunicipio' with a 'main' branch selected. The 'Project' view on the left shows a directory structure with 'src/Main.java' selected. The editor displays the following Java code:

```
6  
7  
8 public class Main {  
9     public static void main(String[] args) {  
10  
11         int op1, op2, op3, op4;  
12  
13  
14         int vot1a = 0;  
15         int vot2a = 0;  
16         int vot3a = 0;  
17         int vot4a = 0;  
18     }  
19 }
```

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

```
error: switch 'm' requires a value  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git commit -m "cambio"  
On branch main  
Your branch is ahead of 'origin/main' by 1 commit.  
(use 'git push' to publish your local commits)  
  
nothing to commit, working tree clean  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git push origin mmejia  
Everything up-to-date  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> 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\USUARIO\IdeaProjects\representantmunicipio> git commit -m "cambio"  
[main 59ead58] cambio  
1 file changed, 2 insertions(+)  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio> git push origin mmejia  
Everything up-to-date  
PS C:\Users\USUARIO\IdeaProjects\representantmunicipio>
```

The status bar at the bottom indicates the file is 'Main.java' in the 'src' directory, with a 'main' branch selected. The encoding is 'UTF-8' and the line length is '13:1'.

## CONCLUSIONES

**Aprendiendo comandos es la forma mas corta de acortar errores en un futuro o saber como resolverlos con lo cual con este taller se aprendio desde crear hasta eliminar elementos que son necesarios para una buenas interacción**

## REFERENCIA BIBLIOGRAFIA

<https://git-scm.com/>

<https://es.wikipedia.org/wiki/Wikipedia>