



**Argentina Programa 4.0**

**Desarrollador Java Inicial**

**Comisión N°128**

**Guía de Ejercicios – Clase 2**

**Docentes:**

**Ing. Demicheri, Soledad**

**Frossasco, Ezequiel**

**Autor:**

**Arias, Matías**

**Año 2023**



## Consigna – Git y GitHub

1. Si aún no lo hizo, cree una cuenta en Github (o Gitlab si prefiere), e instale y configure el cliente de GIT que va a utilizar (colocando su nombre y dirección de correo electrónico), por ejemplo GitHub Desktop.
2. Cree un repositorio remoto nuevo, sin ningún archivo y clonelo localmente.
3. Localmente, agregue un archivo txt, con un texto cualquiera, haga su primer commit y verifique que NO esté en el repositorio remoto. ¿Por qué no está?
4. Ahora haga el push correspondiente y verifique que si esté
5. Modifique el archivo desde la interfaz web de Github, verifique que dicho cambio NO está aplicado localmente ¿Por qué no está?
6. Ahora ejecute el comando correspondiente para traer los cambios y verifique que son visibles.
7. Vuelva a modificar el archivo desde la interfaz web de Github, y antes de hacer el pull, modifique algo DISTINTO en su copia local y haga un commit de dichos cambios.
8. Ahora pruebe hacer un push de sus cambios. ¿Qué sucedió? ¿cómo puede proseguir?
9. Arregle el problema

## Resolución

Para la resolución de la consigna utilizaré GitHub y Git Bash, para mostrar otra propuesta de resolución sin el uso de GitHub Desktop.

1. Cuenta de GitHub: [MatiasArias \(Matias Arias\) \(github.com\)](https://github.com/MatiasArias)

Configuraciones:

---

```
git config --global user.name "Matias Arias"
git config --global user.email "matiasarias384@gmail.com"
```

---

En el caso de querer hacer otras configuraciones:

---

```
git config -global -e
```

---

Abre un archivo con todas las configuraciones.



2. Repositorio remoto:

Link: [MatiasArias/AP40-Clase2: Desarrollador Java Inicial | Guía de Ejercicios - Clase 2 \(github.com\)](https://github.com/MatiasArias/AP40-Clase2)

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner: **MatiasArias** / Repository name: **AP40-Clase2** ✓

Great repository names are short and memorable. Need inspiration? How about **curly-funicular**?

Description (optional):  
**Desarrollador Java Inicial | Guía de Ejercicios - Clase 2**

☒ **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.

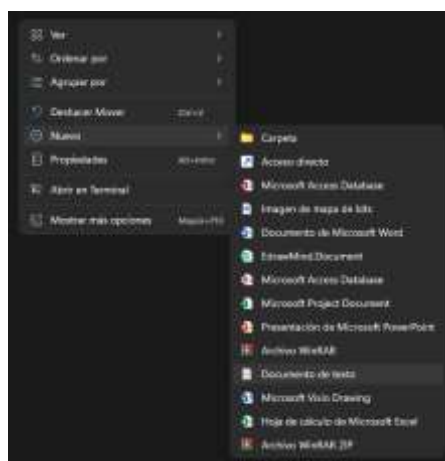
Luego en Git Bash:

```
git clone https://github.com/MatiasArias/AP40-Clase2.git
```

3. Crear archivo localmente:

```
touch miArchivo.txt
```

Podes optar por crear el archivo desde la misma carpeta de la siguiente manera:





Contenido del archivo:

---

Nombre: Matias Arias

Edad: 22 Años

---

Comandos para hacer el commit:

---

```
git add MisDatos.txt
git commit -m "Create misDatos.txt"
```

---

Demostración de que existe localmente pero no remotamente:

---

```
git log
```

---

El `git log` muestra todos los commits que hicimos:

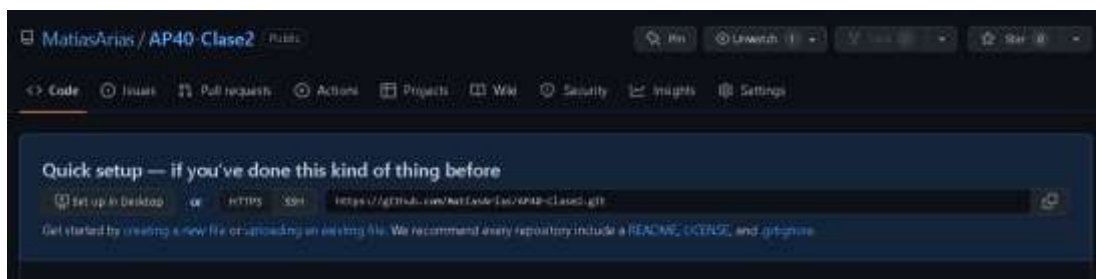
- Commit local



```
matya@MATIAS MINGW64 ~/OneDrive/Escritorio/WORK/Argentina Programa/Clase 2/AP40-Clase2 (main)
$ git log
commit c3fba24637bb9f2f4e32569f51931f8e8ad6139a (HEAD -> main)
Author: Matias Arias <matiasarias384@gmail.com>
Date: Thu Mar 2 14:42:31 2023 -0300

    Create misDatos.txt
```

- No existen commits en el repositorio remoto



El commit no existe en el repositorio remoto debido a que no subimos los cambios que hicimos en nuestro repositorio local. Para realizar los cambios deberíamos realizar un `git push`

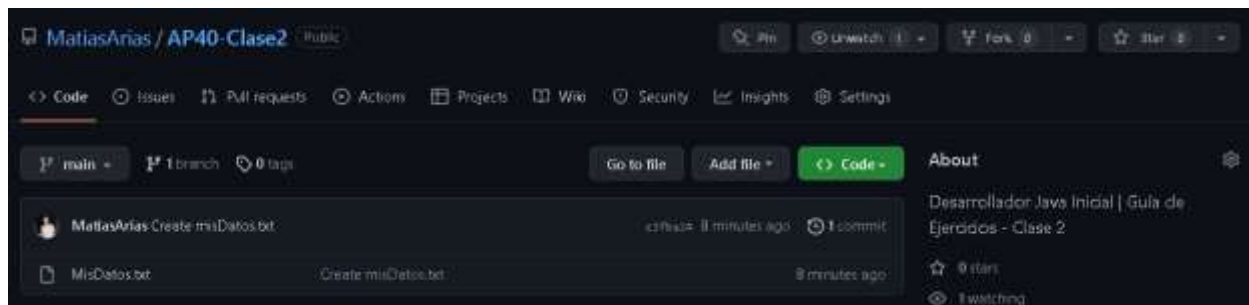


#### 4. Realizamos el push:

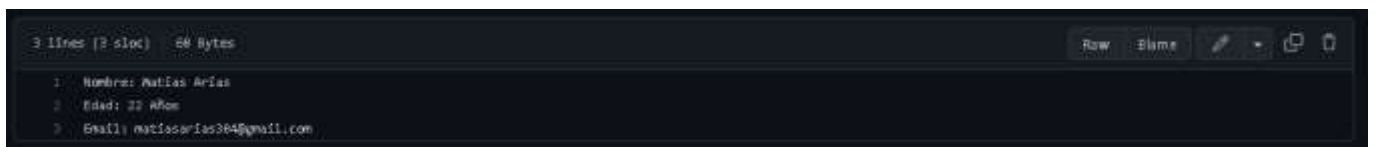
```
git push
```

```
matya@MATIAS MINGW64 ~/OneDrive/Escritorio/WORK/Argentina Programa/Clase 2/AP40-Clase2 (main)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 259 bytes | 86.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MatiasArias/AP40-Clase2.git
 * [new branch]      main -> main
```

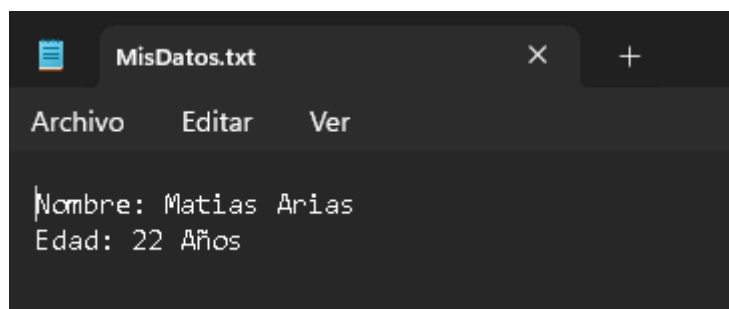
Comprobamos de que esté:



#### 5. Cambios en el repositorio remoto:



Repositorio local:



El cambio no se aplicó localmente porque el no realizamos el pull, es decir no actualizamos los cambios de nuestro repositorio remoto a nuestro repositorio local. En el momento que se comienza a trabajar en grupos siempre es importante realizar el pull antes de realizar cualquier cambio, de esa forma vamos a evitar muchos conflictos.



6. Comando correspondiente:

```
git pull
```

Actualización de los cambios:

```
matya@MATIAS MINGW64 ~/OneDrive/Escritorio/WORK/Argentina Programa/Clase 2/AP40-Clase2 (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 685 bytes | 3.00 KiB/s, done.
From https://github.com/MatiasArias/AP40-Clase2
   c3fba24..9e10919  main       -> origin/main
Updating c3fba24..9e10919
Fast-forward
 MisDatos.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

```
MisDatos.txt
Archivo  Editar  Ver

Nombre: Matias Arias
Edad: 22 Años
Email: matiasarias384@gmail.com
```

7. Generar conflicto:

Archivo modificado en GitHub:

```
4 lines (4 sloc) | 78 Bytes

1  Nombre: Matias
2  Apellido: Arias
3  Edad: 22 Años
4  Email: matiasarias384@gmail.com
```

Archivo modificado localmente:

```
MisDatos.txt
Archivo  Editar  Ver

Nombre: Matias Arias
Edad (Años): 22
Email: matiasarias384@gmail.com
```



Comprometemos los cambios localmente:

---

```
git add MisDatos.txt
```

```
git commit -m "Update MisDatos.txt: edit Edad"
```

---

Tenemos que tener en cuenta que lo que estamos haciendo acá es simular un conflicto con un compañero de trabajo, no es común que editemos nuestro repositorio remoto en GitHub pero es probable que estemos modificando el mismo archivo que nuestro compañero de trabajo, por lo que el fin del ejercicio es simular esa situación e intentar resolverla.

8. Comando:

---

```
git push
```

---

Este comando no se va a poder ejecutar debido al siguiente error:

```
matya@MATIAS MINGW64 ~/OneDrive/Escritorio/WORK/Argentina Programa/Clase 2/AP40-Clase2 (main)
$ git push
To https://github.com/MatiasArias/AP40-Clase2.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/MatiasArias/AP40-Clase2.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Lo que nos está diciendo el problema es que el repositorio al cual queremos subir los cambios tiene una versión más actualizada que no se encuentra localmente. Por lo que se debe realizar en este punto es realizar un pull para actualizar los cambios. Cabe destacar que nos salte este error no indica que tengamos conflictos, sino que no tenemos el repositorio local actualizado. En el siguiente punto intentaremos resolver el conflicto.



## 9. Resolviendo el conflicto:

```
matya@MATIAS MINGW64 ~/OneDrive/Escritorio/WORK/Argentina Programa/Clase 2/AP40-Clase2 (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 696 bytes | 3.00 KiB/s, done.
From https://github.com/MatiasArias/AP40-Clase2
   9e10919..697c825  main      -> origin/main
Auto-merging MisDatos.txt
CONFLICT (content): Merge conflict in MisDatos.txt
error: could not apply 7725574... Update MisDatos.txt: edit Edad
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 7725574... Update MisDatos.txt: edit Edad
```

Yo tengo configurado el Git de forma que a la hora de resolver conflicto entre en estado de REBASE, el REBASE es una utilidad de git para integrar cambios entre ramas/versiones. Solo lo explico por si notan algo distinto a lo que les aparece a ustedes, no hace falta que entiendan en este momento que es lo que hace. Si querés aprender más de Git rebase de dejo el siguiente [link](#).

De cualquier forma, el conflicto se resuelve de la siguiente manera:

- 1- Tenemos que identificar cuales son los archivos que presentan conflictos. Nos podemos ayudar de la información que nos provee la consola, la cual claramente nos dice que el archivo que presenta conflictos es MisDatos.txt
- 2- Al abrir el archivo que presenta conflictos nos encontraremos lo siguiente:

```
MisDatos.txt
Archivo  Editar  Ver

<<<<<<< HEAD
Nombre: Matias
Apellido: Arias
Edad: 22 Años
=====
Nombre: Matias Arias
Edad (Años): 22
>>>>>> 7725574 (Update MisDatos.txt: edit Edad)
Email: matiasarias384@gmail.com
```

Podemos ver que tenemos dos versiones que entraron en conflicto, estas serían **HEAD** y el commit **Update MisDatos.txt: edit Edad**



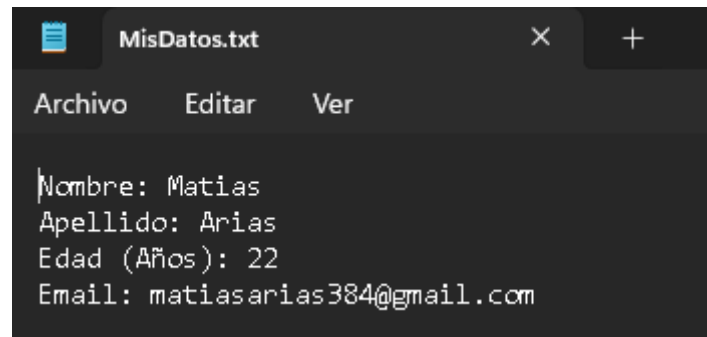


Por lo que tendremos que decidir con que quedarnos, podemos optar por:

- Ambos cambios
- Versión del repositorio remoto
- Versión del repositorio local

Suelen existir herramientas/extensiones que permiten solucionar estos conflictos de forma más rápida por ejemplo en Visual Studio Code. En este caso lo resolveremos manualmente.

### 3- Conflicto resuelto:



Esta es la versión que quiero guardar, por lo que debo eliminar los >>>>, <<<<< y cualquier cosa que le quite funcionalidad a mi archivo, de esa manera evitaremos conflictos del lenguaje de programación.

### 4- Comandos para guardar y subir los cambios:

---

```
git add MisDatos.txt
```

```
git commit -m "R:1/1 - Edit MisDatos.txt"
```

```
git push
```

---

### 5- Comprobamos los cambios:





Repositorio remoto:

Listo!