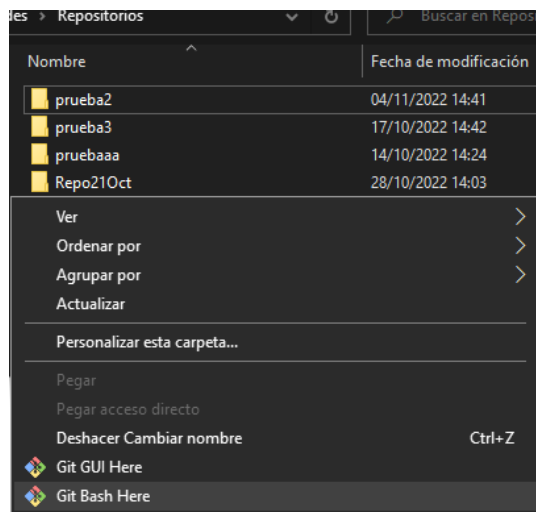
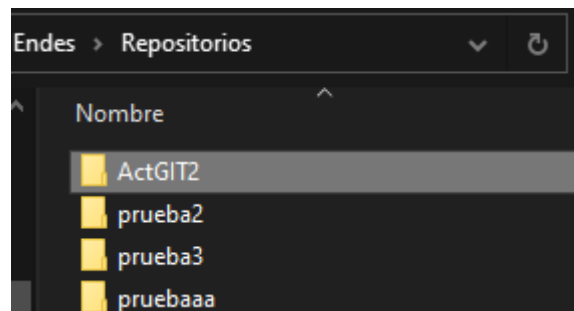


Tarea 3. Git y GitHub

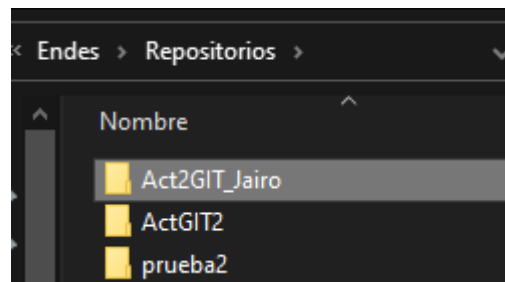
- Abrimos el GUI Bash en el repositorio padre para que se nos guarde en él el repositorio que vamos a clonar escribiendo git clone (seguido del enlace).



```
Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios
$ git clone https://github.com/EnriquePicasso/ActGIT2.git
Cloning into 'ActGIT2'...
remote: Enumerating objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 3
Receiving objects: 100% (3/3), 747.90 KiB | 352.00 KiB/s, done.
```



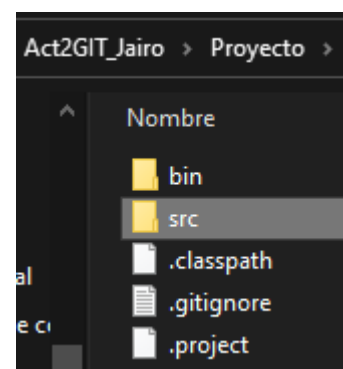
- En el repositorio padre creamos una nueva carpeta con el nombre Act2GIT_Jairo, nos metemos en ella y abrimos el GUI Bash para hacer un git init y convertirlo en un repositorio local.



```
Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo
$ git init
Initialized empty Git repository in C:/Users/Alumnado/Desktop/Endes/Repositorios/Act2GIT_Jairo/.git/
```

- Creamos un método java, que cree un array de 20 números con valores entre 1 y 10, y lo añadimos al repositorio. Posteriormente, nos dirigimos al GUI Bash y hacemos un commit a las modificaciones que hemos realizado.

```
Array20Aleatorio.java
1 public class Array20Aleatorio {
2
3     public static void main(String[] args) {
4
5         final int LONGITUDARRAY = 20;
6         final int RANGOINICIO = 1;
7         final int RANGOFIN = 10;
8         int[] numerosAleatorios = new int[LONGITUDARRAY];
9
10        for (int i = 0; i < LONGITUDARRAY; i++) {
11            numerosAleatorios[i] = (int) (Math.random() * (RANGOFIN - RANGOINICIO + 1) + RANGOINICIO);
12        }
13
14    }
15
16 }
17
18 }
19 }
```



```

$ git status
On branch main

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo/Proyecto (main)
$ git add .
warning: in the working copy of 'Proyecto/.gitignore', LF will be replaced by CR
LF the next time Git touches it

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo/Proyecto (main)
$ git commit -m "Creo proyecto"
[main (root-commit) 7fadb28] Creo proyecto
4 files changed, 42 insertions(+)
create mode 100644 Proyecto/.classpath
create mode 100644 Proyecto/.gitignore
create mode 100644 Proyecto/.project
create mode 100644 Proyecto/src/Array20Aleatorio.java

```

- Nos vamos a GitHub y creamos un nuevo repositorio vacío con el nombre Act2GH_Jairo, después, en el GUI Bash del repositorio local escribimos los siguientes comandos para poder enlazar el hub con el local:

...or push an existing repository from the command line

```

git remote add origin https://github.com/jairopo/Act2GH_Jairo.git
git branch -M main
git push -u origin main

```

```

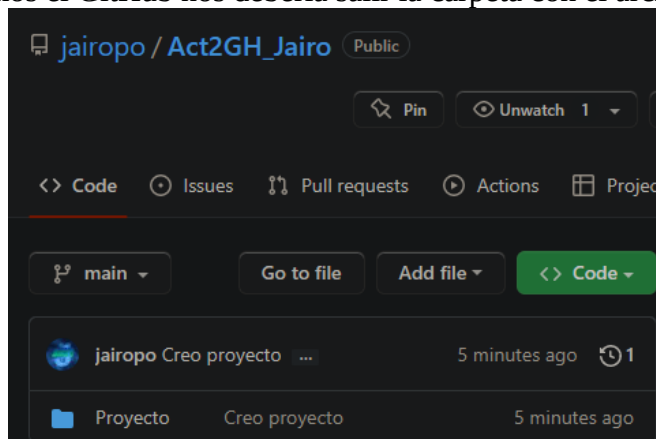
$ git remote add origin https://github.com/jairopo/Act2GH_Jairo.git

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo (main)
$ git branch -M main

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 469 bytes | 156.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/jairopo/Act2GH_Jairo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

```

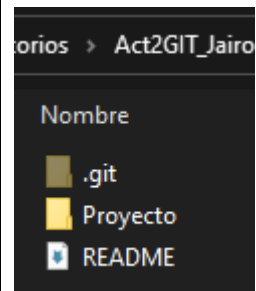
Si actualizamos el GitHub nos debería salir la carpeta con el archivo java.



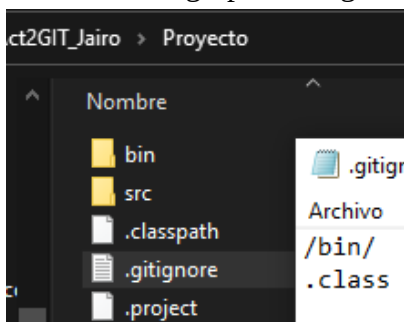
A continuación, en GitHub, le damos a añadir un fichero README.md y lo modificamos de forma que se explique lo que vamos a realizar en la actividad. Después, hacemos “git pull” en el local para tenerlo ahí guardado



```
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 906 bytes | 56.00 KiB/s, done.
From https://github.com/jairopo/Act2GH_Jairo
  7fadb28..71c46f7  main    -> origin/main
Merge made by the 'ort' strategy.
 README.md | 19 ++++++
 1 file changed, 19 insertions(+)
 create mode 100644 README.md
```



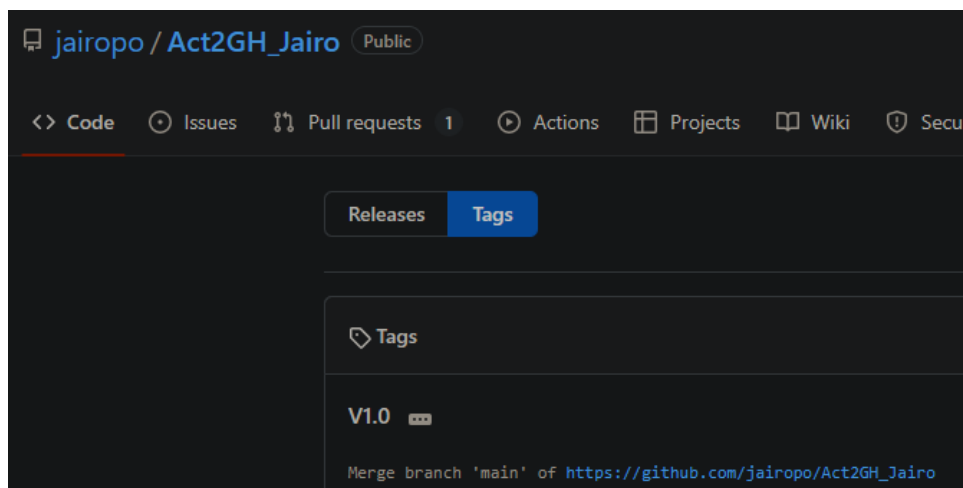
En la carpeta del archivo java encontraremos un archivo .gitignore, por lo que lo usaremos para añadir “.class” de manera que los archivos con esa extensión sean ignorados. Además, guardamos los cambios que hemos hecho, para realizar nuestra primera etiqueta con “git tag V1.0” y lo subimos con “git push --tags”.



```
Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/R
epositorios/Act2GIT_Jairo/Proyecto (main)
$ git commit -m "Modificado .gitignore"
[main 114eda9] Modificado .gitignore
 1 file changed, 1 insertion(+)
```

```
Alumnado@DESKTOP-ADV6Q5L MINGW64
a2)
$ git tag V1.0
```

```
Alumnado@DESKTOP-ADV6Q5L MINGW64
a2)
$ git push --tags
```



- Ahora, crearemos la primera rama con “git branch rama1” y nos metemos en ella con “git checkout rama1”

```
$ git branch rama1
Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes
jairopo/Proyecto (main)
$ git checkout rama1
Switched to branch 'rama1'
```

Nos dirigimos al archivo java y lo modificamos para que sea capaz de imprimir por pantalla el contenido del array y hacemos un commit a la modificación.

```
public class Array20Aleatorio {
    public static void main(String[] args) {
        final int LONGITUDARRAY = 20;
        final int RANGOINICIO = 1;
        final int RANGOFIN = 10;
        int[] numerosAleatorios = new int[LONGITUDARRAY];

        for (int i = 0; i < LONGITUDARRAY; i++) {
            numerosAleatorios[i] = (int) (Math.random() * (RANGOFIN - RANGOINICIO + 1) + RANGOINICIO);
        }

        System.out.println("Contenido del array");
        for (int i = 0; i < LONGITUDARRAY; i++) {
            System.out.print(numerosAleatorios[i] + " ");
        }
    }
}
```

```
$ git add .
Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes
jairopo/Proyecto (rama1)
$ git commit -m "Imprimir por pantalla array"
[rama1 6d25cf5] Imprimir por pantalla array
1 file changed, 4 insertions(+), 1 deletion(-)
```

- Creamos una segunda rama con “git branch rama2” y nos metemos en ella con “git checkout rama2”.

```
$ git branch rama2

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo
Jairo/Proyecto (rama1)
$ git checkout rama2
Switched to branch 'rama2'

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo
Jairo/Proyecto (rama2)
```

Volvemos a acceder al java para añadir un método que nos indique el número de veces que se repite cada número en el array. Después, guardamos realizando un commit.

```
Array20Aleatorio.java X
1 public class Array20Aleatorio {
2
3     public static void main(String[] args) {
4
5         final int LONGITUDARRAY = 20;
6         final int RANGOINICIO = 1;
7         final int RANGOFIN = 10;
8         int[] numerosAleatorios = new int[LONGITUDARRAY];
9
10        for (int i = 0; i < LONGITUDARRAY; i++) {
11            numerosAleatorios[i] = (int) (Math.random() * (RANGOFIN - RANGOINICIO + 1) + RANGOINICIO);
12        }
13
14        System.out.println("Contenido del array");
15        for (int i = 0; i < LONGITUDARRAY; i++) {
16            System.out.print(numerosAleatorios[i] + " ");
17        }
18
19        int repeticiones = 0;
20        for (int i = RANGOINICIO; i <= RANGOFIN; i++) {
21            for (int j = 0; j < LONGITUDARRAY; j++) {
22                if (numerosAleatorios[j] == i) {
23                    repeticiones++;
24                }
25            }
26            System.out.print("\nEl número " + i + " se repite " + repeticiones + " veces.");
27            repeticiones = 0;
28        }
29    }
30 }
31
32 }
```

```
Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo
a2)
$ git status
On branch rama2
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:   Proyecto/src/Array20Aleatorio.java

no changes added to commit (use "git add" and/or "git commit -a")

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo
2)
$ git add .

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo
2)
$ git commit -m "Repeticiones de array (Java)"
[rama2 5448232] Repeticiones de array (Java)
1 file changed, 10 insertions(+)
```

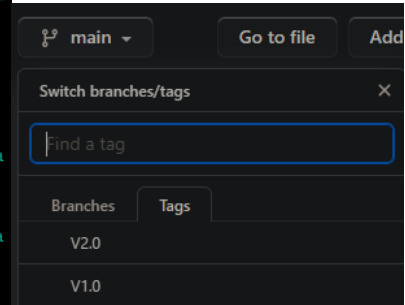
- Para fusionar ambas ramas, debemos irnos primero a la rama1 con el comando “git checkout rama1”, ya que es ahí donde se fusionarán. Una vez ahí, ponemos “git merge rama2”, ya que es la rama que se va a unir. Ahora, guardamos los cambios y creamos otra etiqueta con “git tag V2.0” y la subimos con “git push --tags”

```
$ git checkout rama1
Switched to branch 'rama1'

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo (rama1)
$ git merge rama2
Updating 6d25cf5..5448232
Fast-forward
 Proyecto/src/Array20Aleatorio.java | 11 ++++++++
 1 file changed, 11 insertions(+)

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo (rama1)
$ git tag V2.0

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo (rama1)
$ git push --tags
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 641 bytes | 160.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/jairopo/Act2GH_Jairo.git
 * [new tag]          V2.0 -> V2.0
```



- Accedemos a la rama main mediante “git checkout main” y modificamos el archivo java para que en lugar de 20, el array sea de 30. Guardamos los cambios realizando un commit.

```
Array20Aleatorio.java X
1 public class Array20Aleatorio {
2
3 public static void main(String[] args) {
4
5     final int LONGITUDARRAY = 30;
6     final int RANGOINICIO = 1;
7     final int RANGOFIN = 10;
8     int[] numerosAleatorios = new int[LONGITUDARRAY];
9
10    for (int i = 0; i < LONGITUDARRAY; i++) {
11        numerosAleatorios[i] = (int) (Math.random() * (RANGOFIN - RANGOINICIO + 1) + RANGOINICIO);
12    }
13
14
15
16 }
```

```
$ 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:   Proyecto/src/Array20Aleatorio.java

no changes added to commit (use "git add" and/or "git commit -a")

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo (main)
$ git add .

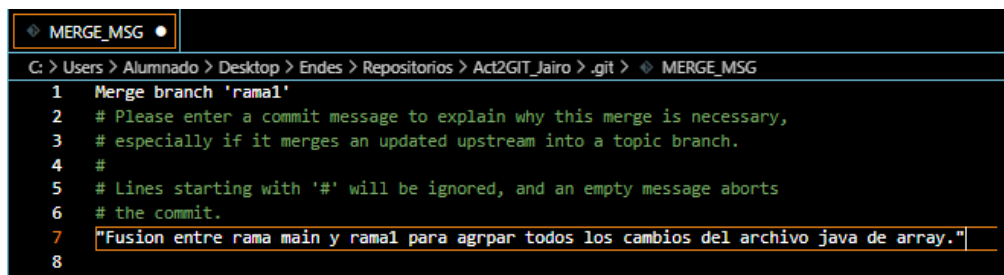
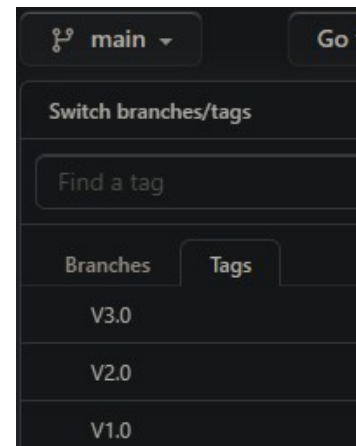
Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo (main)
$ git commit -m "Array de longitud 30 (Java)"
[main da15383] Array de longitud 30 (Java)
1 file changed, 1 insertion(+), 1 deletion(-)
```

- Nos mantenemos en la rama main y volvemos a realizar una fusión, esta vez de la rama1 y la main, por tanto, ponemos “git merge rama1”, a continuación, se nos abrirá el VSCode para que especifiquemos el porqué del merge. Volvemos a hacer una etiqueta llamada V3.0 con el comando “git tag” y la subimos con “git push --tags”.

```
$ git merge rama1
Auto-merging Proyecto/src/Array20Aleatorio.java
Merge made by the 'ort' strategy.
Proyecto/src/Array20Aleatorio.java | 14 ++++++
1 file changed, 14 insertions(+)

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo (main)
$ git tag V3.0

Alumnado@DESKTOP-ADV6Q5L MINGW64 ~/Desktop/Endes/Repositorios/Act2GIT_Jairo (main)
$ git push --tags
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 923 bytes | 307.00 KiB/s, done.
Total 10 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To https://github.com/jairopo/Act2GH_Jairo.git
 * [new tag]         V3.0 -> V3.0
```

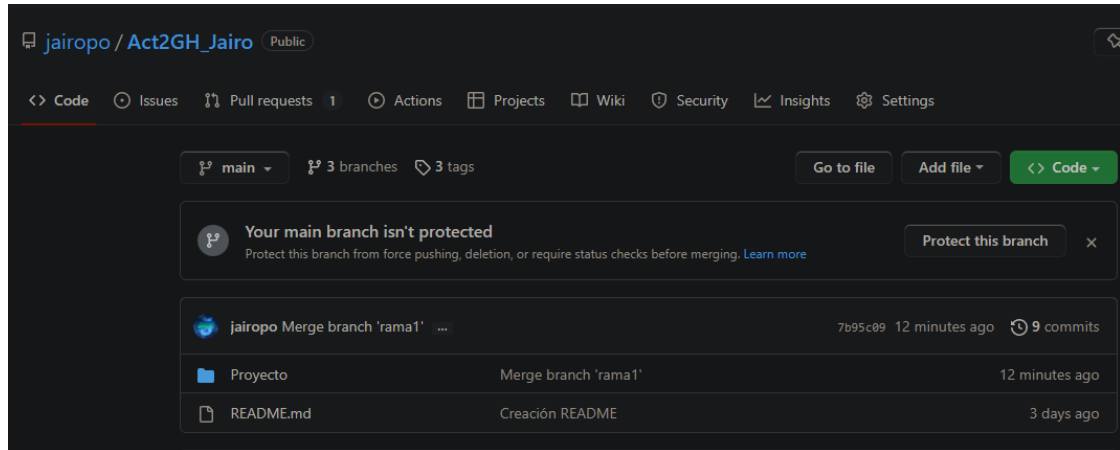


```
Array20Aleatorio.java X
1 public class Array20Aleatorio {
2
3     public static void main(String[] args) {
4
5         final int LONGITUDARRAY = 30;
6         final int RANGOINICIO = 1;
7         final int RANGOFIN = 10;
8         int[] numerosAleatorios = new int[LONGITUDARRAY];
9
10        for (int i = 0; i < LONGITUDARRAY; i++) {
11            numerosAleatorios[i] = (int) (Math.random() * (RANGOFIN - RANGOINICIO + 1) + RANGOINICIO);
12        }
13
14        System.out.println("Contenido del array");
15        for (int i = 0; i < LONGITUDARRAY; i++) {
16            System.out.print(numerosAleatorios[i] + " ");
17        }
18
19        int repeticiones = 0;
20        for (int i = RANGOINICIO; i <= RANGOFIN; i++) {
21            for (int j = 0; j < LONGITUDARRAY; j++) {
22                if (numerosAleatorios[j] == i) {
23                    repeticiones++;
24                }
25            }
26            System.out.print("\nEl número " + i + " se repite " + repeticiones + " veces.");
27            repeticiones = 0;
28        }
29
30    }
31 }
32 }
33 }
```

Como se puede observar en la imagen anterior, el programa tiene todas las modificaciones que hemos estado realizando.

- Por último, solo nos queda guardar todos los cambios que hemos realizado subiéndolos a GitHub, así que deberemos escribir “git push -all”, de esta forma, se suben automáticamente todos los cambios. Además, deberemos actualizar el contenido del archivo Readme con las últimas acciones realizadas.

```
$ git push --all
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/jairopo/Act2GH_Jairo.git
 c0e5f1d..7b95c09  main -> main
 6d25cf5..5448232  rama1 -> rama1
 0b003c9..5448232  rama2 -> rama2
```



- Realizamos un fork en el repositorio de un compañero. Nos vamos a GitHub y buscamos su perfil