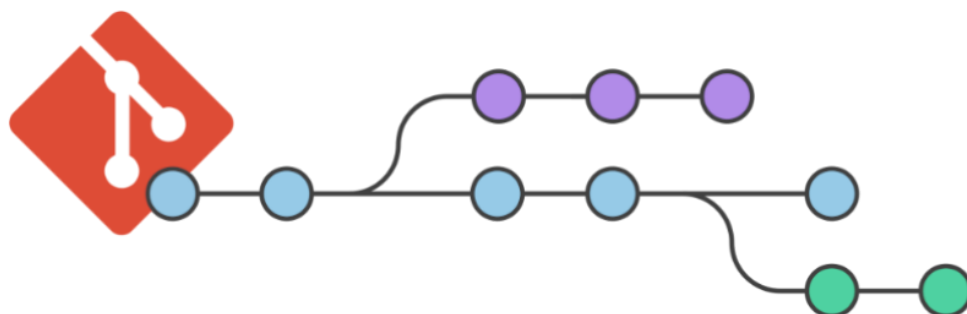


UD00

Cómo usar el VSCode integrado en GitHub



Este material está bajo una [Licencia Creative Commons Atribución-Compartir-Igual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).
Derivado a partir de material de David Martínez Peña (<https://github.com/martinezpenya>).

1. **VsCode** integrado en **GitHub**

1. 1. Localizar repositorio
1. 2. Abrir **VsCode** integrado
1. 3. Modificar un archivo
1. 4. Git integrado
1. 5. Pull request

2. **Tareas**

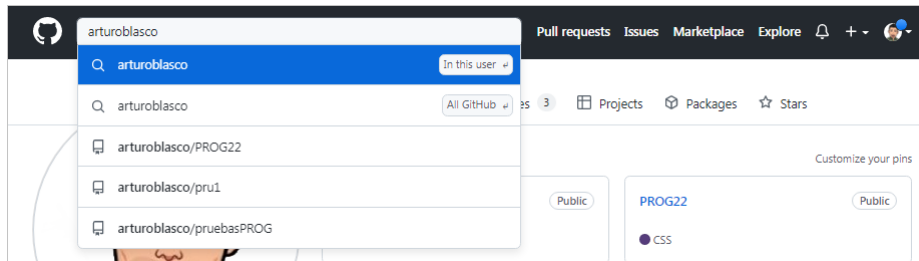
2. 1. [GitHub 1](#)
2. 2. [GitHub 2](#)
2. 3. [Repositorio Local + VsCode + Repositorio remoto](#)

1. VsCode integrado en GitHub

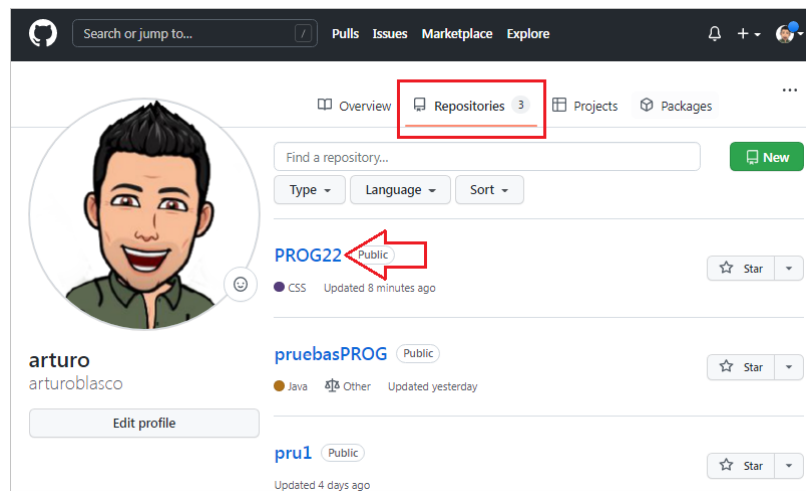
1.1. Localizar repositorio

Primero localizamos el repositorio con el que queremos colaborar:

1. Buscamos al usuario (En todo GitHub):



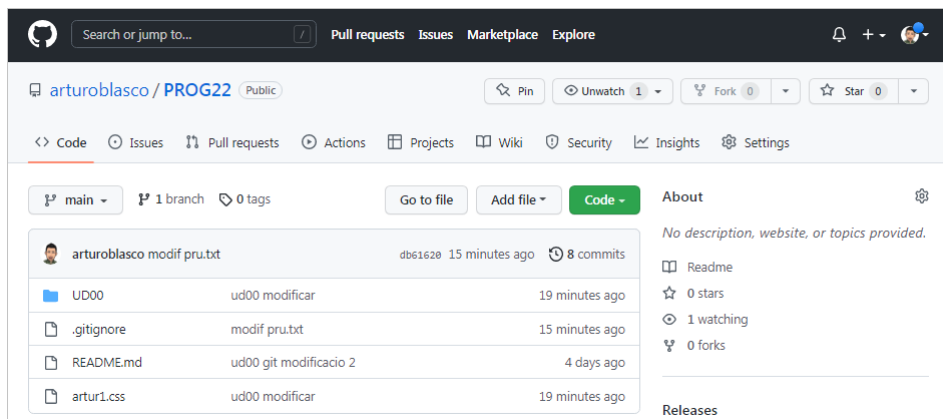
2. Elegimos la pestaña **Repositorios** y después el repositorio en cuestión:



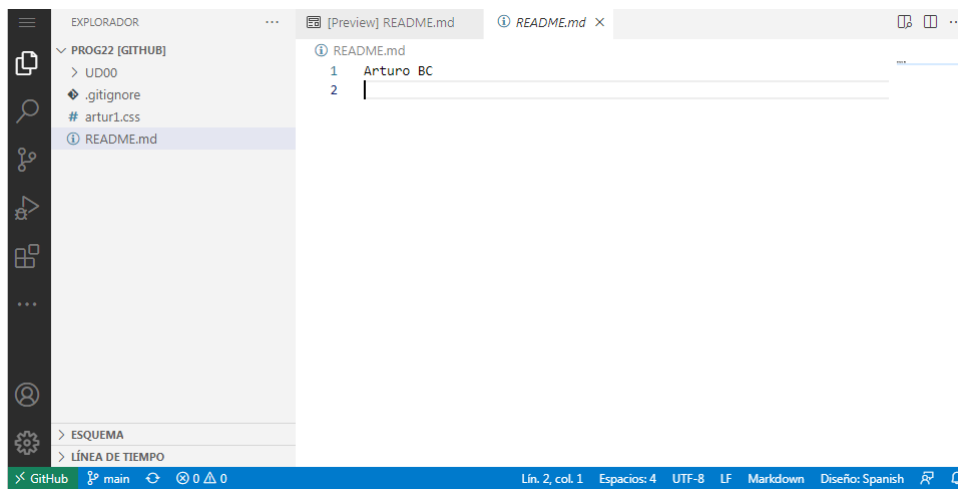
1.2. Abrir VSCode integrado

Github cuenta con un editor online muy potente basado en VSCode.

Una vez visualizamos el código del repositorio en cuestión:



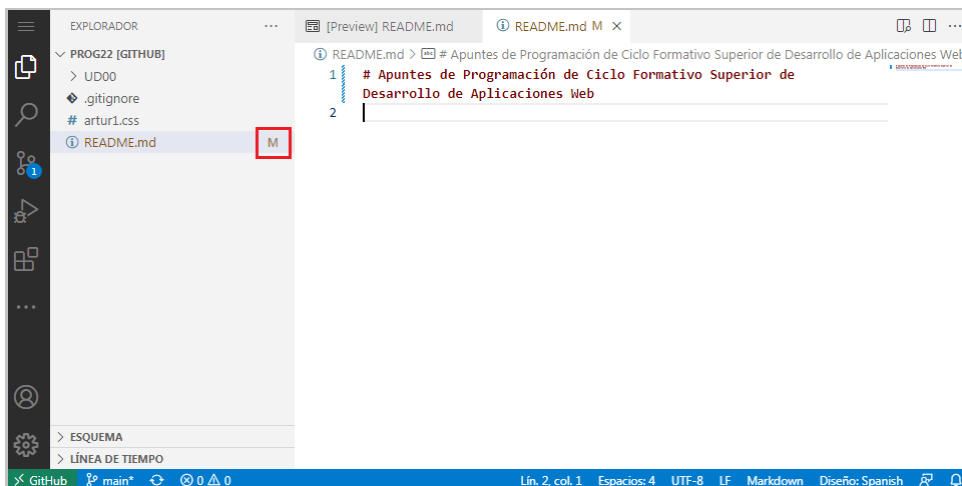
para abrir el editor solo debemos pulsar la tecla "." (punto) de nuestro teclado:



Visualizaremos la estructura de carpetas y archivos en un editor **VSCode** integrado en el navegador Web.

1.3. Modificar un archivo

Una vez detectada la errata dentro del código [markdown](#) que es muy fácil de interpretar (a poco que le dediquéis unos minutos) podemos modificar el archivo en cuestión, y a su lado aparecerá una **M** porque el archivo está modificado.

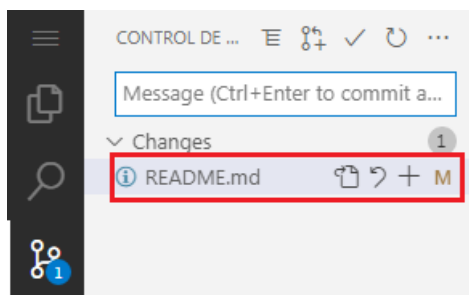


1.4. Git integrado

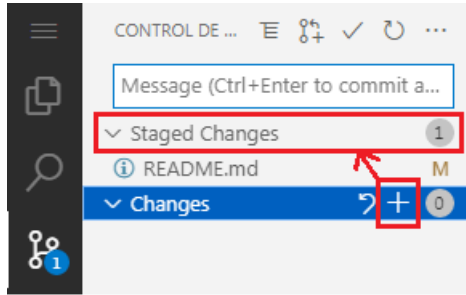
VSCode lleva integrado un gestor de GIT, el tercer icono de la barra lateral:



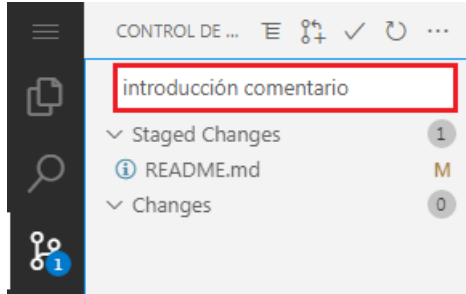
1. **Verificamos** los archivos cambiados:



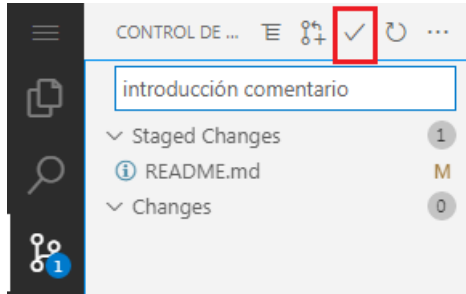
2. Los pasamos a "**staged area**" con el símbolo "+":



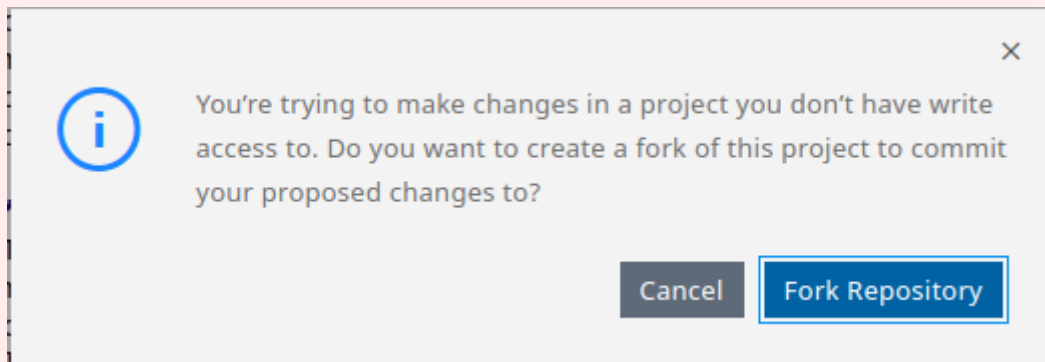
3. Añadimos el **comentario** del commit:



4. Realizamos el **commit**:

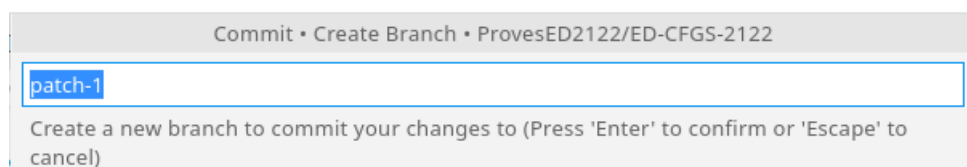


Si el repositorio no es nuestro mostrará este mensaje:

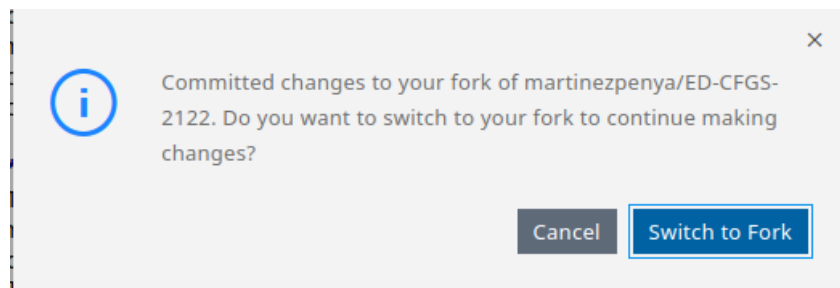


En realidad podemos hacer el fork antes o después, aquí tienes un pequeño [vídeo](#) que explica qué es un fork.

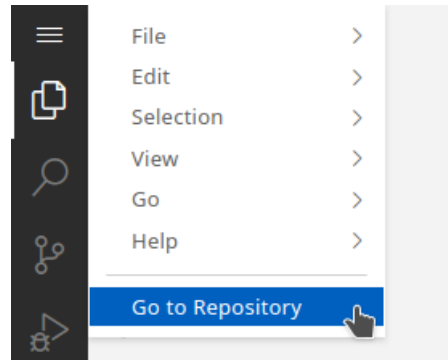
A continuación nos pide el nombre de la rama que se creará y que luego podremos solicitar se incluya en el proyecto original:



Ahora nos pregunta si ya que hemos creado un fork, queremos cambiar el repositorio, y trabajar sobre nuestro fork en lugar de sobre el proyecto original, pulsamos sobre [Switch to Fork]:

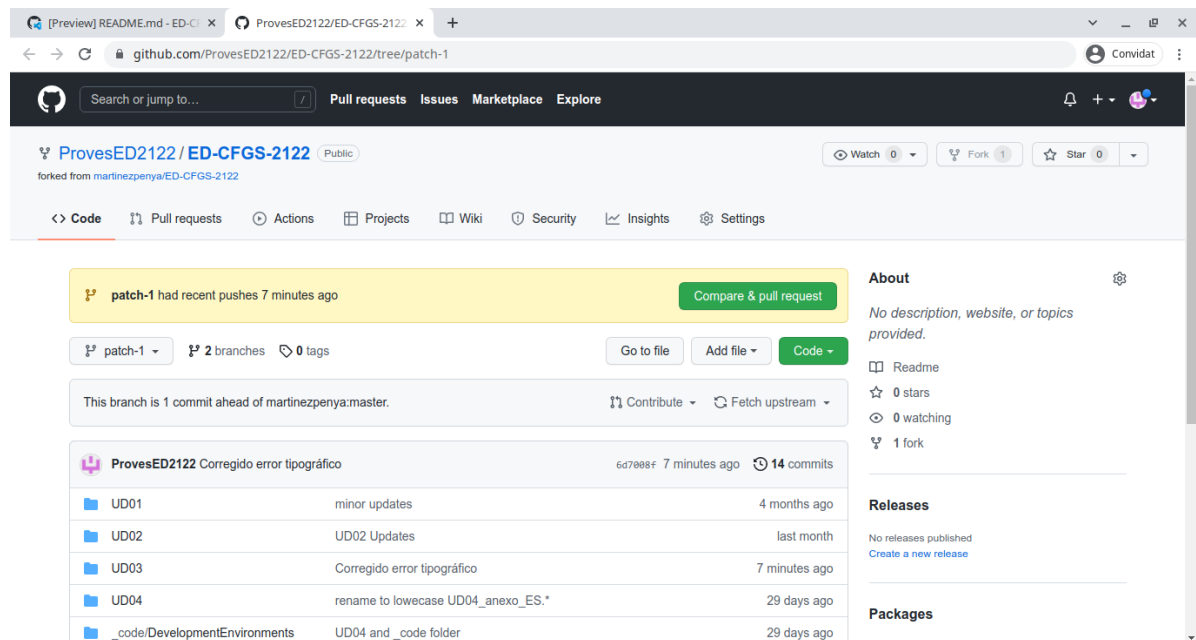


Ahora ya podemos salir del editor VSCode pulsando sobre el botón de las tres líneas horizontales y elegir la opción "Go to Repository":



1.5. Pull request

Una vez volvemos a nuestro repositorio (nuestro fork), detectará que hay cambios respecto al repositorio original y nos propone que realicemos un pull request (una petición al usuario propietario del repositorio original para que incluya nuestra modificación).



Una vez pulsado el botón [Compare & pull request] nos aparece la siguiente pantalla:

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base repository: martinezpenya/ED-CFGS-2122

base: master

←

head repository: arturoblasco/ED-CFGS-2122

compare: patch-1

✓ Able to merge

These branches can be automatically merged.

Update UD03_GitHubVsCode_ES.md

Write Preview

H B I

modificació

Attach files by dragging & dropping, selecting or pasting them.

☒ Allow edits by maintainers

Create pull request

Helpful resources
[GitHub Community Guidelines](#)

Debemos asegurarnos de que la modificación se puede agregar al repositorio original "*Able to merge*", y que indicamos en los comentarios nuestro nombre completo para que el *profesor* nos identifique. Fíjate que el nombre del pull request es el nombre del commit que hicimos desde VSCode.

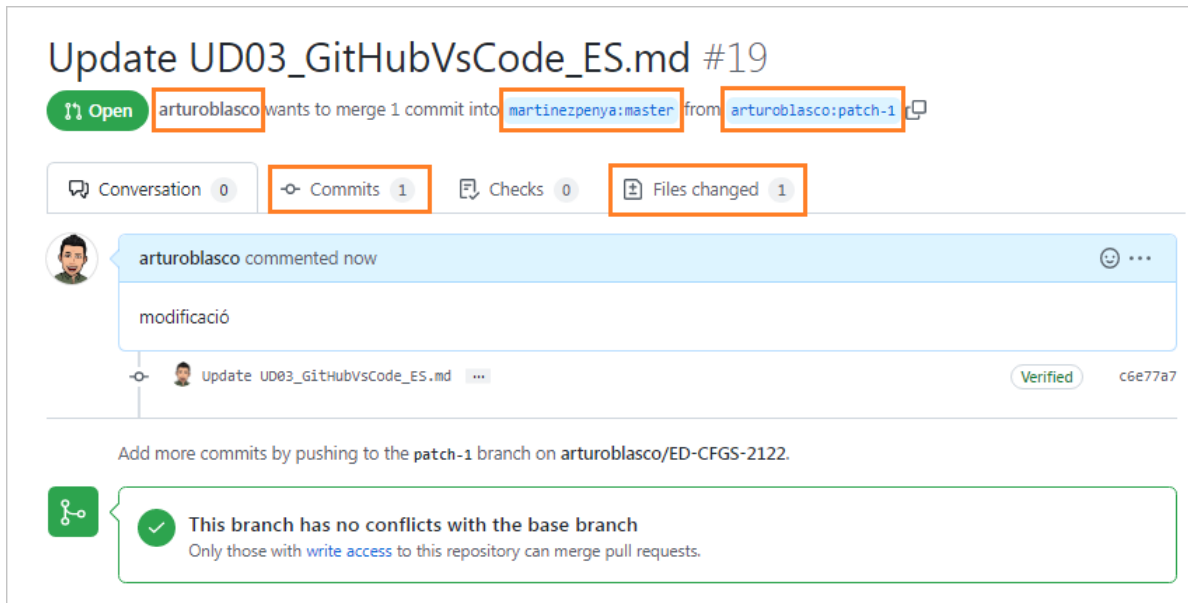
Ahora debemos pulsar el botón [Create pull request].

2. Tareas

2.1. GitHub 1

Sigue los pasos de esta práctica guiada para sugerir una modificación de cualquiera de los archivos de los repositorios del profesor [arturoblasco](#).

1. Adjunta a la tarea de AULES un pdf de nombre actividad01tunombre con la captura de pantalla similar a esta donde se vé que has solicitado el pull request y que estás esperando a que se integre en el repositorio original.
2. Explica qué significan cada uno de los 5 apartados señalados en la captura:



2.2. GitHub 2

Siguiendo los pasos de la práctica anterior,;

1. crearemos un repositorio en el que un compañero propondrá un pull-request que nosotros deberemos integrar en nuestro repositorio original y viceversa.
2. deberemos realizar un pull-request a nuestro compañero y que él lo acepte.

Debes adjuntar a la tarea de AULES un pdf de nombre actividad02tunombre con los pasos a seguir una vez recibido el *pull-request* para aceptarlo y que se integre en nuestro repositorio, y mostrar además donde aparece el nombre del usuario que ha colaborado con nosotros en el repositorio.

2.3. Repositorio Local + VsCode + Repositorio remoto

En la siguiente práctica vamos a configurar el espació de trabajo en el que estaremos a lo largo de este curso/módulo de Programación.

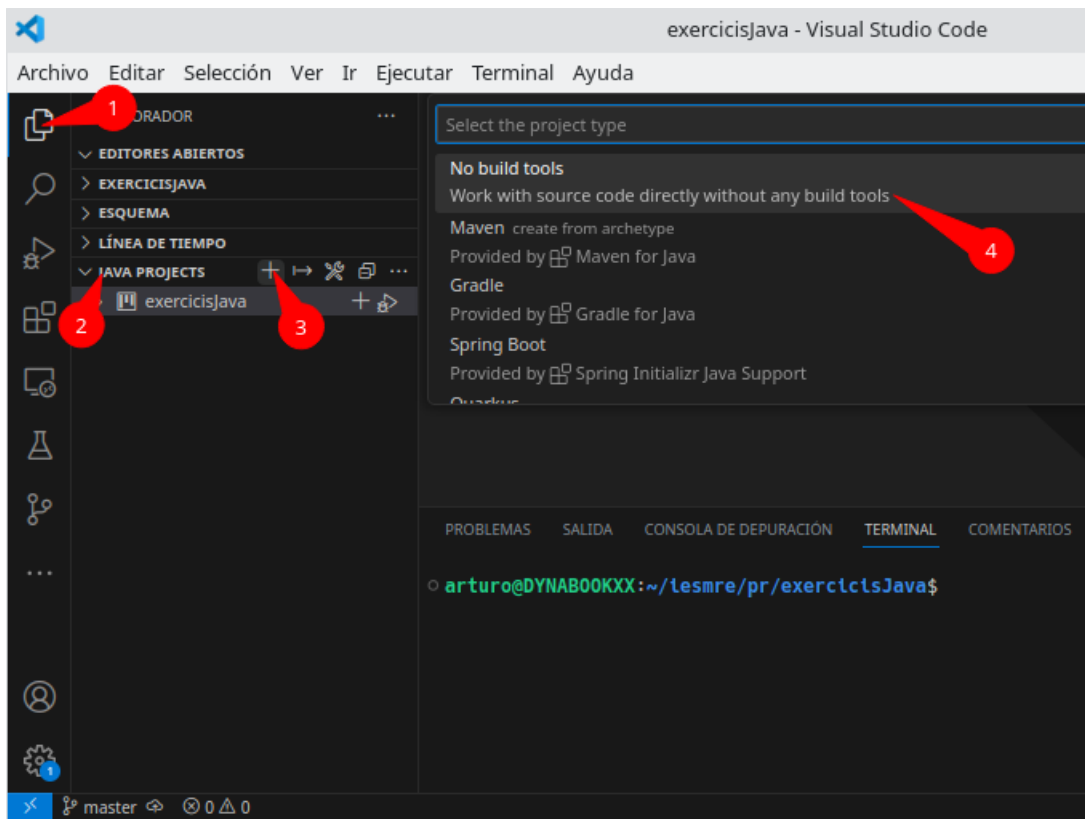
La idea va a consistir en:

1. crear un proyecto Java en VS Code,

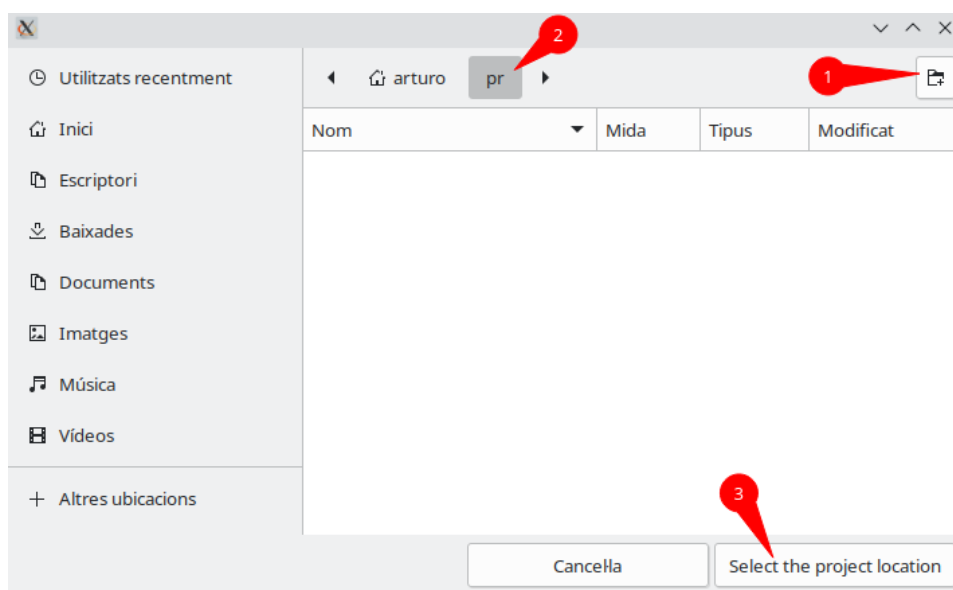
3. por otra parte, crearemos un repositorio remoto en GitHub (será privado y daremos acceso al profesor/a).
4. enlazaremos nuestro Git local con nuestro Git remoto (en GitHub); así, podremos ir subiendo todos nuestros programas.

Empecemos:

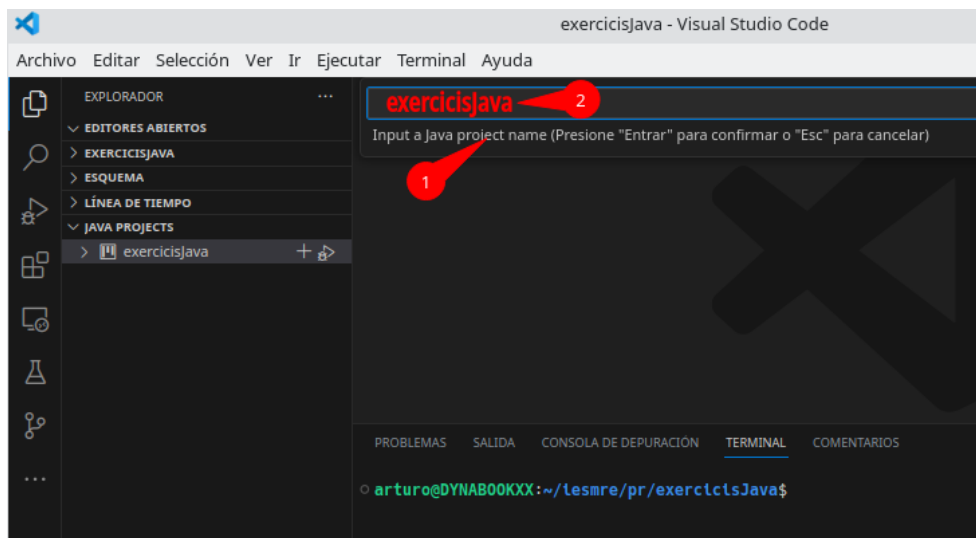
1. Abrimos VS Code. Accedemos a la pestaña de JAVA PROJECTS, pulsamos el icono de **+** y elegimos la primera opción **No build tools**.



Se abre otra ventana flotante en la que elegimos el lugar donde va a ir nuestro proyecto; crearemos en nuestra raíz `~` la carpeta `pr` y pulsamos botón **Select the project location**.



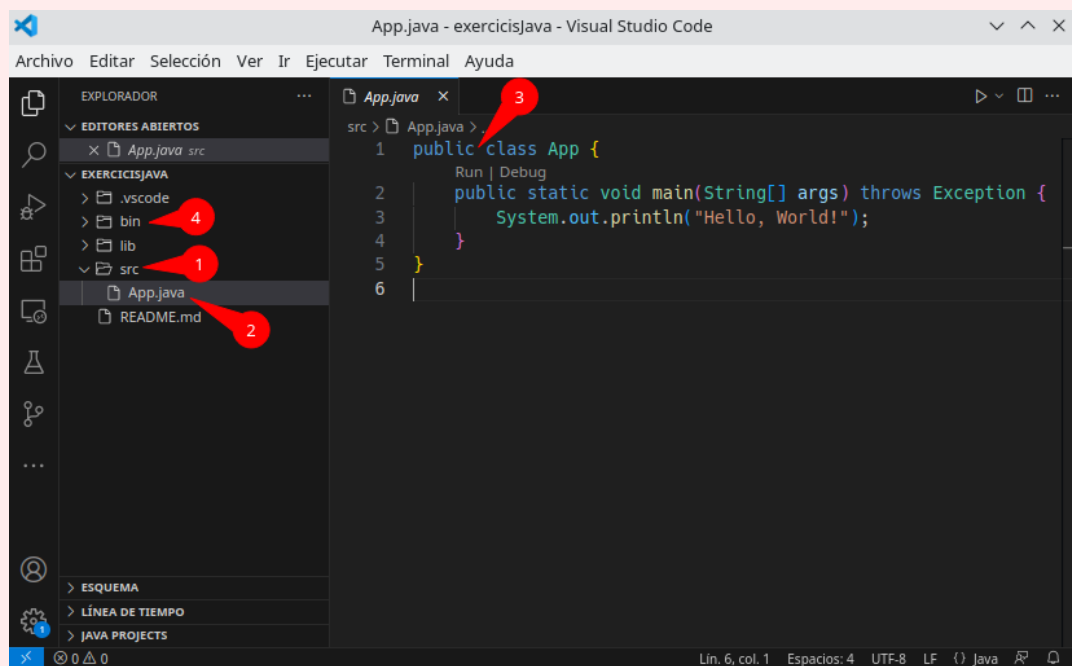
Escribimos el nombre del proyecto `exercicisJava`.



Podemos observar la estructura de nuestro primer proyecto Java.

Observamos en primer lugar la carpeta `src` que será la que contenga todos nuestros ficheros fuente `.java`. A la derecha podemos observar el contenido del fichero `App.java`.

La carpeta `bin` contendrá los ficheros ejecutables `.class` que se generen de la compilación de nuestros ficheros `.java`.



2. Des de un terminal, accedemos dentro de la carpeta de nuestro recién creado proyecto `~\pr\exercicisJava`.

```
cd ~\pr\exercicisJava
```

```
arturo@DYNABOOKXX:~/pr/exercicisJava$ ls -la
total 28
drwxrwxr-x 6 arturo arturo 4096 de set. 12 10:49 .
drwxrwxr-x 3 arturo arturo 4096 de set. 12 10:48 ..
drwxrwxr-x 2 arturo arturo 4096 de set. 12 10:49 bin
drwxrwxr-x 2 arturo arturo 4096 de set. 12 10:48 lib
-rw-r--r-- 1 arturo arturo 708 de set. 12 10:48 README.md
drwxrwxr-x 2 arturo arturo 4096 de set. 12 10:48 src
drwxrwxr-x 2 arturo arturo 4096 de set. 12 10:48 .vscode
```

```
git init
```

```
arturo@DYNAB00KXX:~/pr/exercicisJava$ git init
S'ha inicialitzat un repositori buit del Git en /home/arturo/pr/exercicisJava/.git/
```

Si vemos la estructura de la carpeta del proyecto vemos una carpeta oculta `.git`.

```
arturo@DYNAB00KXX:~/pr/exercicisJava$ ls -la
total 32
drwxrwxr-x 7 arturo arturo 4096 de set. 12 10:55 .
drwxrwxr-x 3 arturo arturo 4096 de set. 12 10:48 ..
drwxrwxr-x 2 arturo arturo 4096 de set. 12 10:49 bin
drwxrwxr-x 7 arturo arturo 4096 de set. 12 10:55 .git
drwxrwxr-x 2 arturo arturo 4096 de set. 12 10:48 lib
-rw-r--r-- 1 arturo arturo 708 de set. 12 10:48 README.md
drwxrwxr-x 2 arturo arturo 4096 de set. 12 10:48 src
drwxrwxr-x 2 arturo arturo 4096 de set. 12 10:48 .vscode
```

Si realizamos un `git status` observamos que todavía no hemos introducido las carpetas contenidas en el proyecto en nuestro repositorio.

```
git status
```

```
arturo@DYNAB00KXX:~/pr/exercicisJava$ git status
En la branca master

No s'ha fet cap comissió encara

Fitxers no seguits:
  (useu «git add <fitxer>...» per a incloure'ls en la comissió)
    .vscode/
    README.md
    bin/
    src/

no hi ha res afegit a cometre però hi ha fitxers no seguits (useu «git add» per a seguir-los)
```

Para ello: `git add .` y a continuación `git commit -m "1) cargar proyecto inicial"`.

```
git add .
```

```
git commit -m "1) carga proyecto inicial"
```

```
arturo@DYNAB00KXX:~/pr/exercicisJava$ git add .
arturo@DYNAB00KXX:~/pr/exercicisJava$ git commit -m "1) carga proyecto inicial"
[master (comissió arrel) 6ea1e4e] 1) carga proyecto inicial
4 files changed, 30 insertions(+)
create mode 100644 .vscode/settings.json
create mode 100644 README.md
create mode 100644 bin/App.class
create mode 100644 src/App.java
```

Se puede comprobar con `git status` que, en este momento, no existe ninguna modificación en el área de trabajo que no se encuentre en el repositorio.

```
arturo@DYNAB00KXX:~/pr/exercicisJava$ git status
En la branca master
no hi ha res a cometre, l'arbre de treball està net
```

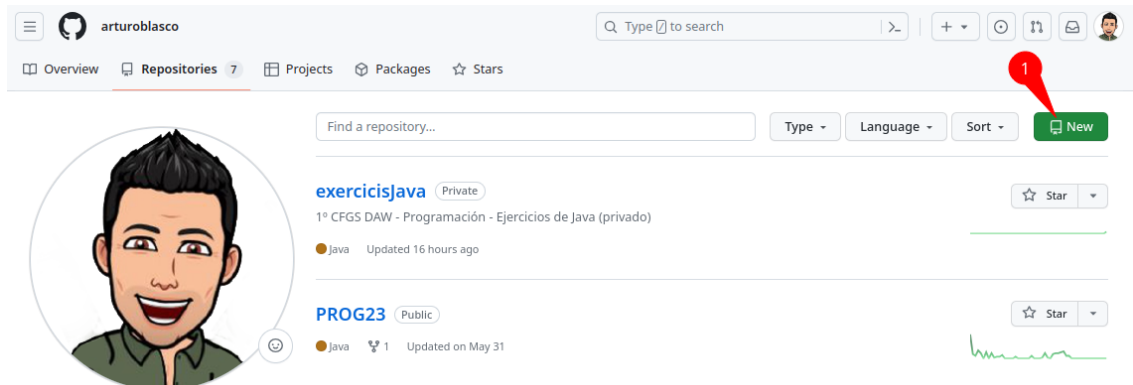
Se puede observar este primer commit con `git log`.

```
git log
```

```
arturo@DYNABOOKXX:~/pr/exercicisJava$ git log
commit 6ea1e4e29bc5250f1690e6bd77536223b24b8ee3 (HEAD -> master)
Author: arturo bc <arturoblasco@iesmre.com>
Date: Tue Sep 12 11:07:58 2023 +0200

1) carga proyecto inicial
```

3. Es hora de crear nuestro repositorio remoto en GitHub:




El nombre de nuestro repositorio `exercicisJava`; descripción `1º CFGS DAW - Programación - Ejercicios de Java`, por ejemplo, acceso `Private` y NO crearemos el fichero README ni `.gitignore` (por ahora).

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 *

 arturoblasco

Repository name *

exerciciJava

✓ exerciciJava is available.

Great repository names are short and memorable. Need inspiration? How about [redesigned-fiesta](#) ?

Description (optional)

1º CFGS DAW - Programación - Ejercicios de Java (privado)

3

☐

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.

Initialize this repository with:

☐

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

ⓘ You are creating a private repository in your personal account.

4
Create repository

Si vemos el resultado de crear el anterior repositorio, entre otras opciones de terminar encontramos:

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

git remote add origin git@github.com:arturoblasco/exerciciJava.git

Vamos a terminar ejecutando esta orden en nuestro terminal (DENTRO de nuestra carpeta de proyecto Java / repositorio local Git):

```
git remote add origin git@github.com:arturoblasco/exercicisJava.git
```

```
arturo@DYNAB00KXX:~/pr/exercicisJava$ git remote add origin git@github.com:arturoblasco/exerciciJava.git
```

4. Para terminar, enlazamos el repositorio local de Git con el repositorio remoto en GitHub:

```
git push origin master
```

```
arturo@DYNAB00KXX:~/pr/exercicisJava$ git push origin master
Enter passphrase for key '/home/arturo/.ssh/id_ed25519':
Enter passphrase for key '/home/arturo/.ssh/id_ed25519':
S'estan enumerant els objectes: 9, fet.
S'estan comptant els objectes: 100% (9/9), fet.
Compressió de diferències usant fins a 4 fils
S'estan comprimint els objectes: 100% (6/6), fet.
S'estan escrivint els objectes: 100% (9/9), 1.44 KiB | 735.00 KiB/s, fet.
Total 9 (diferències 0), reusat 0 (diferències 0)
To github.com:arturoblasco/exerciciJava.git
* [new branch]      master -> master
```

Si vemos en nuestro repositorio de GitHub la sincronización ha sido realizada:

Para terminar, ponemos al profesor/a del módulo de Programación como colaborador de nuestro repositorio: