



PROGRAMACIÓN 3

TRABAJO SCRUM

Grupo: ERROR-404

Integrantes:

Ahumada, Brian

Alancay, Abel Matias

Alsina, Maximiliano Gabriel

Berrini, Alejandro

Calle, Sonia

Chavez, Rodrigo

Costa, Maria Eugenia

Navarro, Lucas

Sanguinetti Flores, Pablo

CONSIGNA

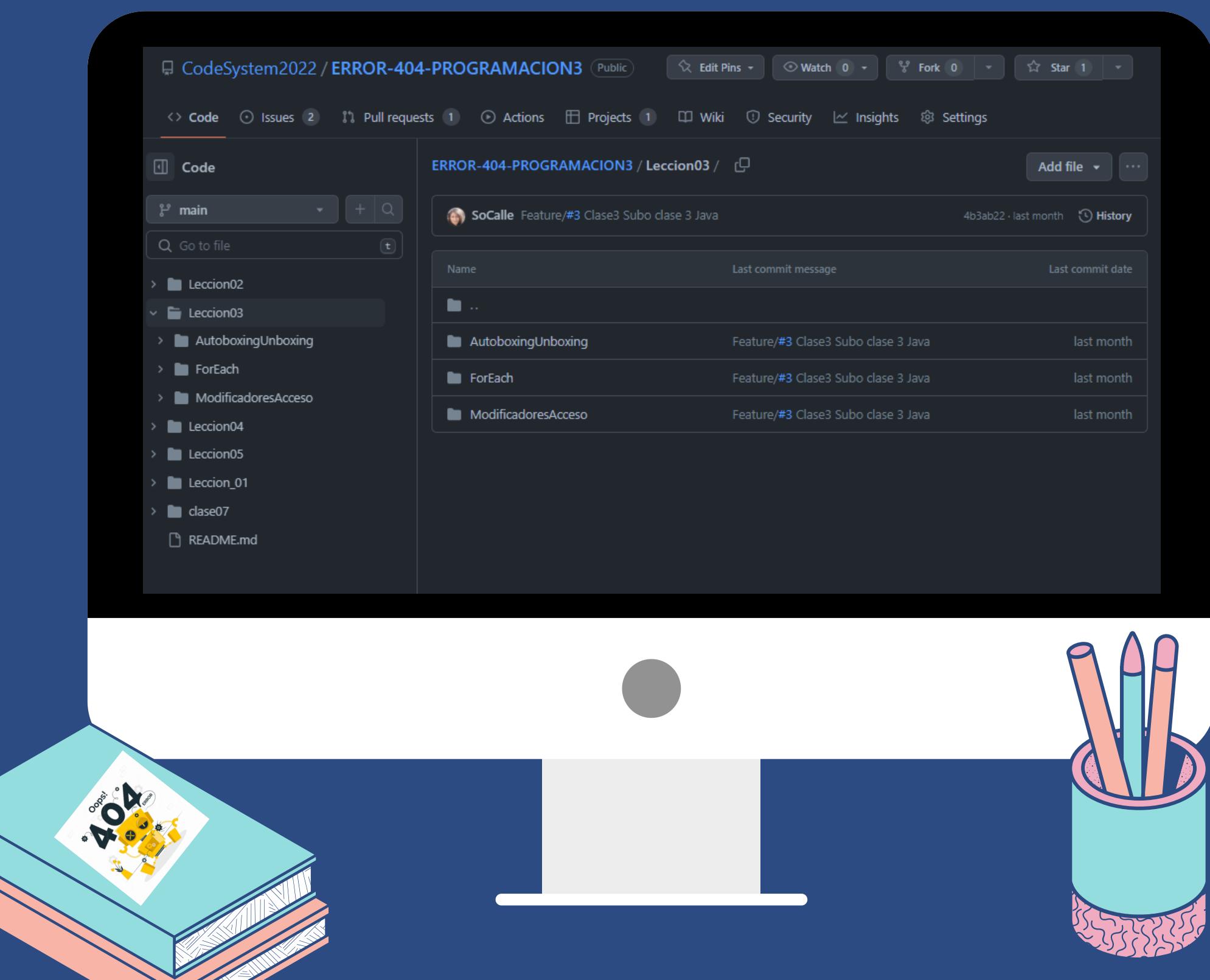


- ELEGIR UNA SEMANA DESDE PRINCIPIO DE CLASE A FINALES DE MAYO
- EXPLICAR COMO SE DESENVUELVE EL EQUIPO CON METODOLOGÍA SCRUM Y MÉTODOS DE TRABAJO

SEMANA ELEGIDA : CLASE 9 (7 Junio)

Temas: Proyecto: CalculadoraUTN

- 9.1 Instalación de IntelliJ y abrimos el primer archivo
- 9.2 Definiendo las variables
- 9.3 Ponemos la entrada de información
- 9.4 Cargamos el menú
- 9.5 Agregamos el switch
- 9.6 Ciclo y su salida con break
- 9.7 Agregamos try catch para las excepciones
- 9.8 Factorizamos el código a través del método mostrarMenu
- 9.9 Función ejecutarOperacion y pruebas





¿Cómo empezamos?

1) En GitHub, dentro de la organización "CodeSystem2022", creamos el repositorio: **ERROR-404-PROGRAMACION3**

The screenshot shows the GitHub repository page for 'ERROR-404-PROGRAMACION3'. The repository is public and part of the organization 'CodeSystem2022'. Key details shown include:

- Code**: The active tab.
- Issues**: 2 issues.
- Pull requests**: 1 pull request.
- Actions**: 1 action.
- Projects**: 1 project.
- Wiki**, **Security**, **Insights**, **Settings**: Other repository sections.

Branches: main (selected), 2 branches, 0 tags.

Commits (by eugenia1984):

- Actualizo los integrantes (bbf4452, 3 days ago)
- Leccion02 (feature-#2-clase2.java, 2 months ago)
- Leccion03 (Feature/#3 Clase3 Subo clase 3 Java, last month)
- Leccion04 (Subo la clase04 polimorfismo,sobrescritura, instanceof, last month)
- Leccion05 (Clase 5 Conversión de objetos, Downcasting, Upcasting, hashCode, equals, 2 weeks ago)
- Leccion_01/MundoPc (Feature/#1 Clase_01- subimos proyecto de la primera clase., 2 months ago)
- clase07 (agrego los ejercicios de la clase 7 con interfaces y actualizo el link, 4 days ago)
- README.md (Actualizo los integrantes, 3 days ago)

About: Repository del grupo ERROR 404 para la materia Programación 3. Languages: Java 100.0%.

Releases: No releases published. Create a new release.

Packages: No packages published. Publish your first package.

Contributors: 5 contributors.

Languages: Java 100.0%.

2) Creamos un TABLERO para tener la visualización de los trabajos de cada SPRINT (establecimos una duración de una semana por Sprint)

En BACKLOG tenemos la tarea a realizar en la semana (subir la clase).

El integrante que tiene asignada la tarea la pasa a TODO.

Cuando comienza la tarea la pasa a DOING, se crea la rama y trabaja en al misma.

Una vez finalizada, se solicita el MERGE REQUEST y se pasa a CODE REVIEW para que otro integrante lo vea. Una vez que el otro integrante lo aprueba, se mergea a la rama "main", se elimina la rama y se pasa a DONE

¿Qué tareas tiene cada miembro del equipo?

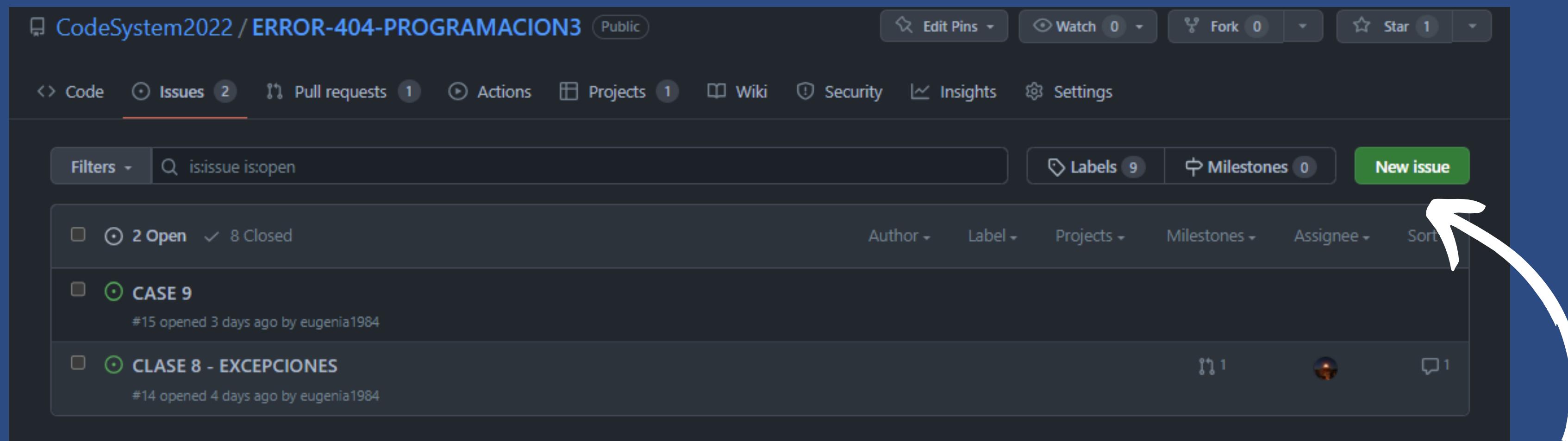
- Uno de los integrantes del equipo va a ser el SCRUM MASTER del Sprint, encargándose que todos los miembros del equipo realicen la tarea del mismo (la clase de la semana).
- 1.
- Otro integrante del equipo tomará la tarea del Sprint.
- 2.
- El resto del equipo realiza la misma tarea (localmente, sin subirla al repositorio en GitHub).

Si alguien del equipo tiene dudas de ejercicio lo consulta en el grupo para ayudarnos entre todos.



¿Qué hace el integrante con la tarea asignada por Sprint?

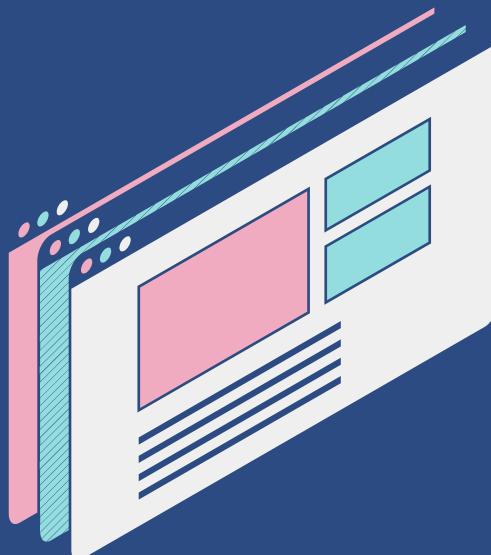
1- En el REPOSITORIO de GitHub crea un ISSUE



HACEMOS CLICK PARA CREAR EL NUEVO ISSUE

**2- Completa en el título con el número de clase,
opcionalmente se puede agregar el tema de la misma**

**3- En el cuerpo
del issue se
deja la tarea,
por ejemplo:
subir la clase**



A screenshot of a GitHub repository titled "CodeSystem2022 / ERROR-404-PROGRAMACION3". The "Issues" tab is selected, showing one open issue. The issue title is "clase 9" and the body contains the text "- Subir los ejercicios de la clase 9". The right sidebar shows the issue details: Assignees (Pablo1653), Labels (enhancement), Projects (None yet), Milestone (No milestone), and Development (Shows branches and pull requests linked to this issue). Hand-drawn annotations include a large green arrow pointing from the "clase 9" title to the "Issues" tab, another green arrow pointing from the "clase 9" title to the issue body, and a green checkmark above the title bar.

4- Se autoasigna el issue

6- Hacemos click para commitear issue

**5- Agrega una
etiqueta**



6 - Abre la terminal en el repositorio, se situa en la rama "main" y la actualiza con el comando: git pull, para asegurarse que esté actualizada la rama local.

```
MINGW64:/c/Users/juan/Desktop/error-404-programacion-3
juan@user MINGW64 ~/Desktop/error-404-programacion-3 (main)
$ git pull
Already up to date.
```

**7 - Desde la rama "main" actualizada se crea la nueva rama con el comando:
git checkout -b <nombre-de-la-nueva-rama>**

```
MINGW64:/c/Users/juan/Desktop/programacion3
juan@user MINGW64 ~/Desktop/programacion3 (main)
$ git pull
Already up to date.

juan@user MINGW64 ~/Desktop/programacion3 (main)
$ git checkout -b feat/clase9
Switched to a new branch 'feat/clase9'

juan@user MINGW64 ~/Desktop/programacion3 (feat/clase9)
$ |
```

Nombramos las ramas con feat/#<nro de issue>-clase<nro de clase>

8- Viendo los videos de la clase se realizan los ejercicios de Java



A screenshot of the NetBeans IDE 8.2 interface. The title bar says "AutoboxingUnboxing - NetBeans IDE 8.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has various icons for file operations. The left sidebar shows a project named "AutoboxingUnboxing [feat#3-clase3]" with a "Source Packages" node and a "test" folder containing "TestAutoboxingUnboxing.java". The main editor window displays the Java code for "TestAutoboxingUnboxing.java". The code demonstrates autoboxing and unboxing:

```
1 package test;
2
3 public class TestAutoboxingUnboxing {
4     public static void main(String[] args) {
5         // Clase envolvente o Wrapper
6         /*
7             Clase envolventes de tipo primitivos
8             int = la clase envolvente es Integer
9         */
10
11        int entero = 10; // Tipo primitivo
12        System.out.println("entero = " + entero);
13        Integer entero2 = 10; // Tipo object
14        System.out.println("entero2 = " + entero2.toString()); // Autoboxing
15
16        int entero3 = entero; // Unboxing
17        System.out.println("entero3 = " + entero3);
18    }
19 }
20
21
22
```

The "Output" window at the bottom shows the execution results:

```
run:
entero = 10
entero2 = 10
entero3 = 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

9- Se pasa al STAGE los ejercicio de la clase en la rama en que se está trabajando con el comando: `git add .`

```
Pablito@DESKTOP-CS5APG7 MINGW64 ~/Desktop/ERROR-404-PROGRAMACION3 (javaclass)
$ git add .
```

10- Se commitea con el comando:
`git commit -m "se sube el ejercicio de la clase 9"`

```
Pablito@DESKTOP-CS5APG7 MINGW64 ~/Desktop/ERROR-404-PROGRAMACION3 (javaclass)
$ git commit -m "ls"
[javaclass f214471] ls
5 files changed, 87 insertions(+)
create mode 100644 Leccion_09/CalculadoraUTN/CalculadoraUTN.iml
create mode 100644 Leccion_09/CalculadoraUTN/out/production/CalculadoraUTN/CalculadoraUTN.class
create mode 100644 Leccion_09/CalculadoraUTN/src/CalculadoraUTN.java
create mode 160000 Leccion_09/CalculadoraUTN/src/ERROR-404-PROGRAMACION3
create mode 100644 Leccion_09/Leccion_09.iml
```

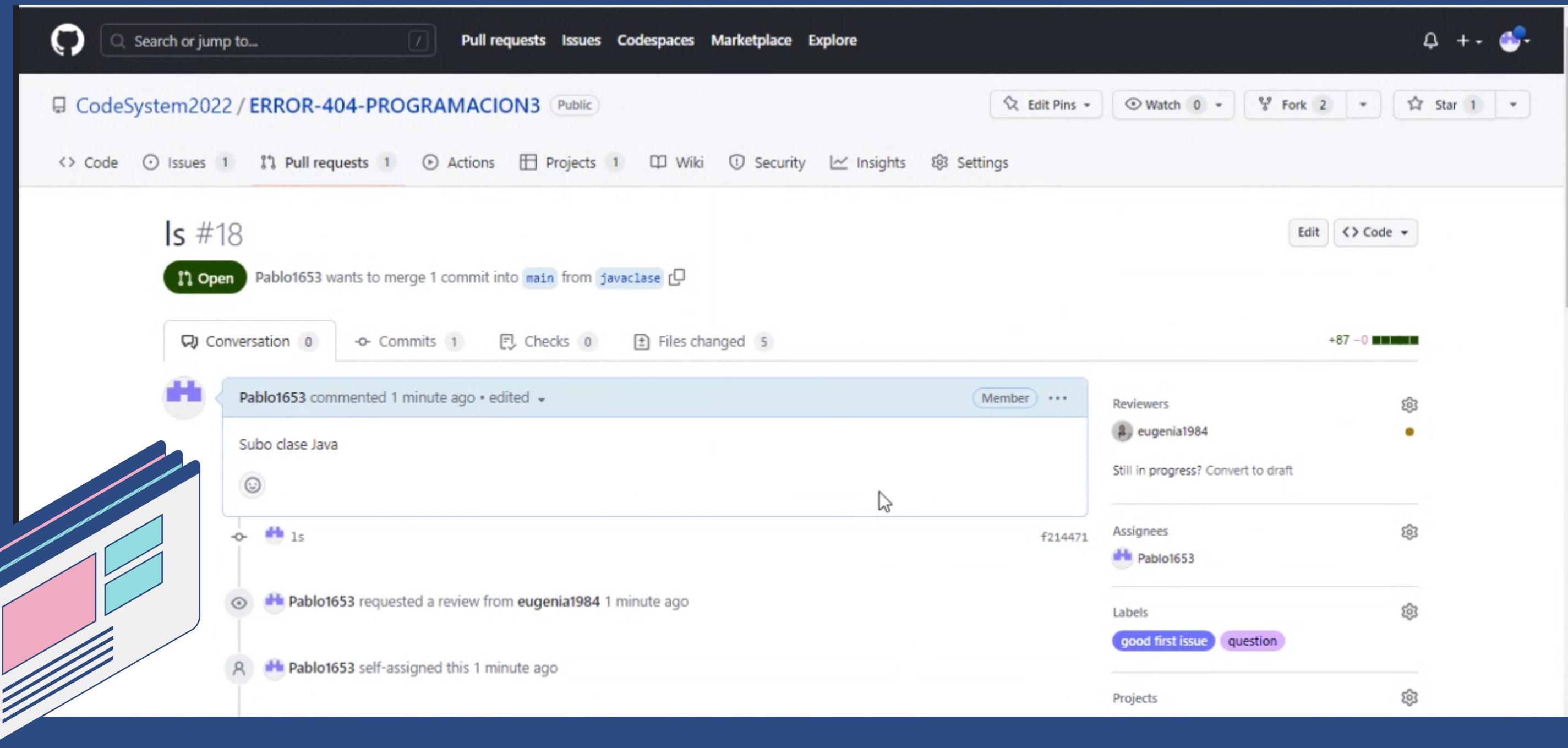


11- Se sube con el comando: *git push*

```
Pablito@DESKTOP-C55APG7 MINGW64 ~/Desktop/ERROR-404-PROGRAMACION3
$ git push --set-upstream origin javaclass
```



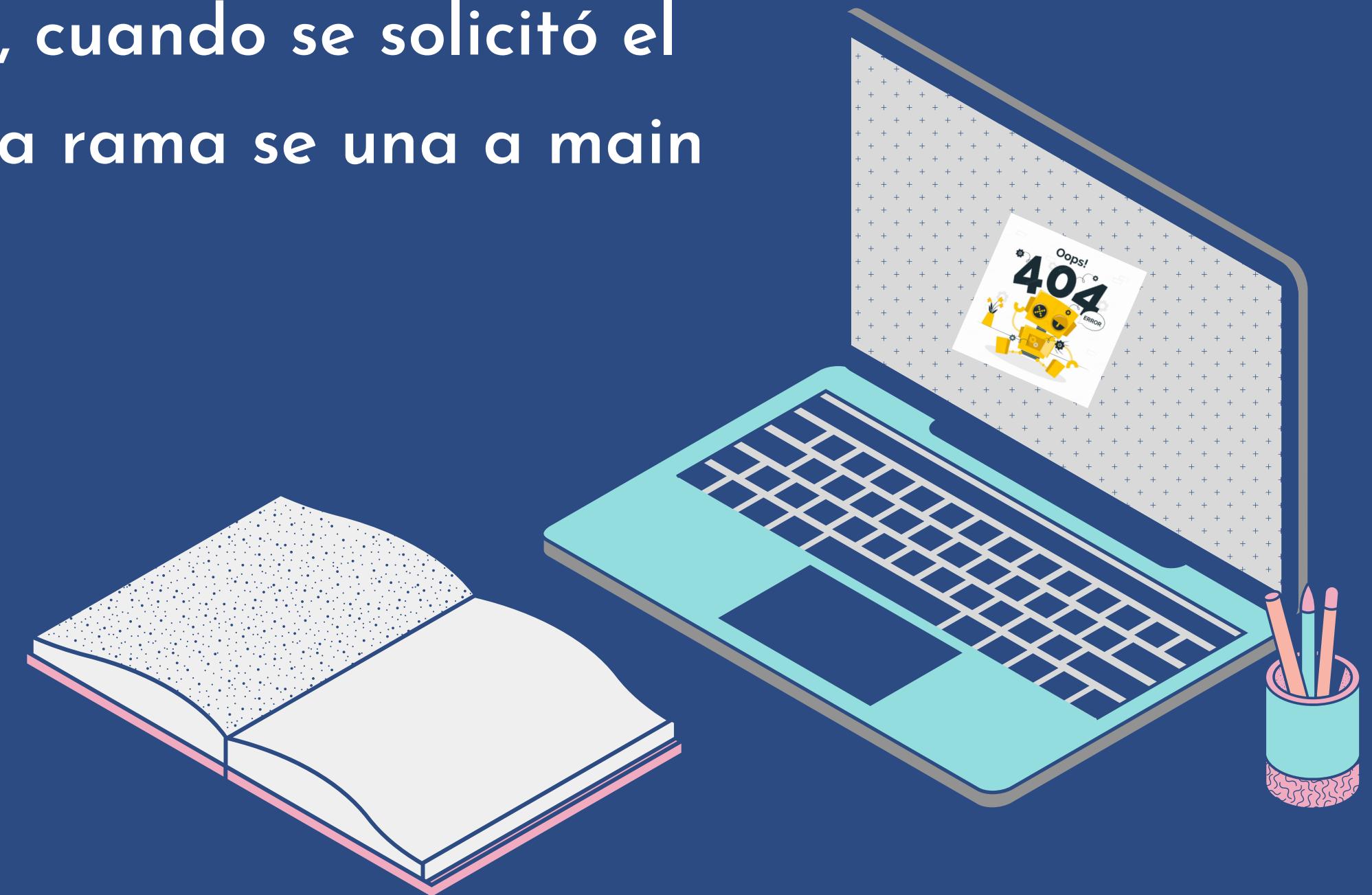
12- Se solicita el MERGE REQUEST, para que el integrante que es el SCRUM en el SPRINT revise los ejercicios y si están bien, mergee la rama con 'main' y borre la rama en la que se estuvo trabajando.



13- Se hace un comentario en el ISSUE del repositorio avisando que ya se creo la rama, agrego la clase y solicito el merge request.



14- La TARJETA de la CLASE 3 del TABLERO fue pasando de BACKLOG a TODO cuando se tomo la tarea. Luego a DOING cuando se fue creando la rama y subiendo los ejercicios de la clase. Y finalmente a CODE REVIEW, cuando se solicitó el merge request. Una vez que la rama se una a main y se elimine pasará a DONE.



15- Como ya mi rama local fue mergead a main, no la necesito tener más, por lo que la borro con el comando:
git branch -D <nombre-de-la-rama-a-borrar>

```
Pablito@DESKTOP-C55APG7 MINGW64 ~/Desktop/ERROR-404-PROGRAMACION3 (main)
$ git branch -d javaclase
warning: deleting branch 'javaclase' that has been merged to
      'refs/remotes/origin/javaclase', but not yet merged to HEAD.
Deleted branch javaclase (was f214471).

Pablito@DESKTOP-C55APG7 MINGW64 ~/Desktop/ERROR-404-PROGRAMACION3 (main)
```



Dejamos el link del repositorio grupal de la materia:

<https://github.com/CodeSystem2022/ERROR-404-PROGRAMACION3>

