

# TRABAJO SCRUM



LABORATORIO III



### Integrantes

Agustín Perez Pesce

Mario Cristian Sánchez

Braian Troncoso

Santiago Mendoza

Franco Sebastián Genre

Sebastián Galván

Mariano Farias

# TRABAJO SCRUM

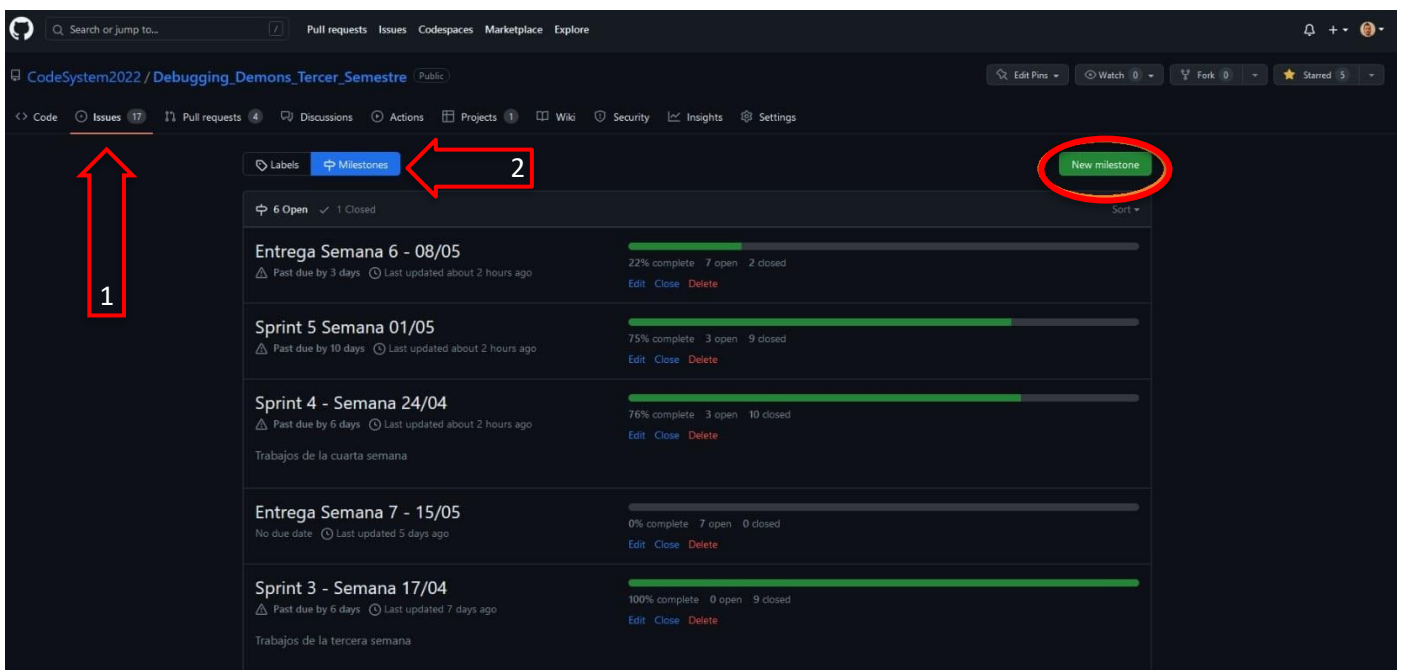
**Scrum master elegido por el grupo:** Agustín Pérez Pesce

**Semana 8 entrega de prácticos (22/05/2023)**

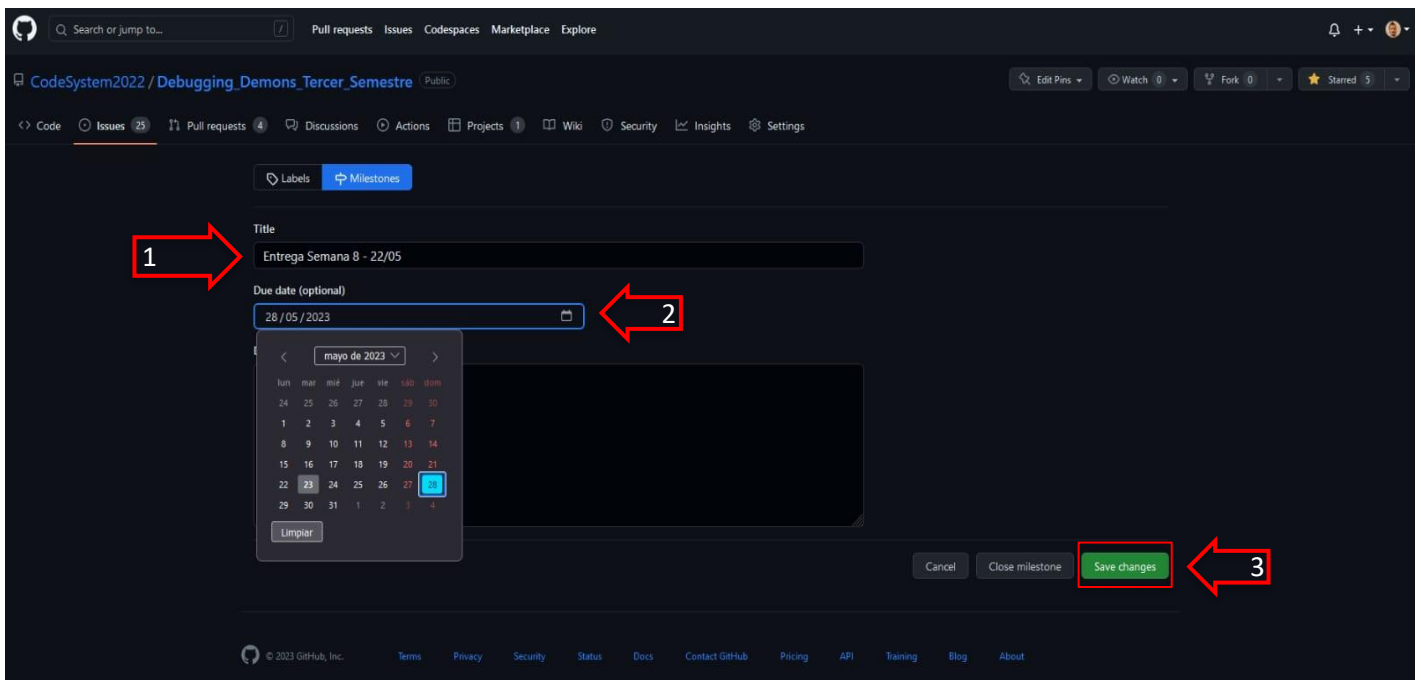
## TAREAS DEL SCRUM MASTER

### 1) LA PRIMERA TAREA QUE REALIZO EL SCRUM MASTER FUE CREAR EL MILESTONE

Las “**MILESTONES**” (hitos) son utilizadas para agrupar y realizar un seguimiento de los problemas, tareas o actividades relacionadas dentro de un proyecto.



Nos dirigimos al menú de **ISSUES**(1), seleccionamos **MILESTONE**(2). Presionamos sobre el botón de “**NEW MILESTONE**” y se desplegará la pantalla en donde se crean las **MILESTONE**.

CodeSystem2022 / Debugging\_Demons\_Tercer\_Semestre

Labels Milestones

Title  
Entrega Semana 8 - 22/05

Due date (optional)  
28/05/2023

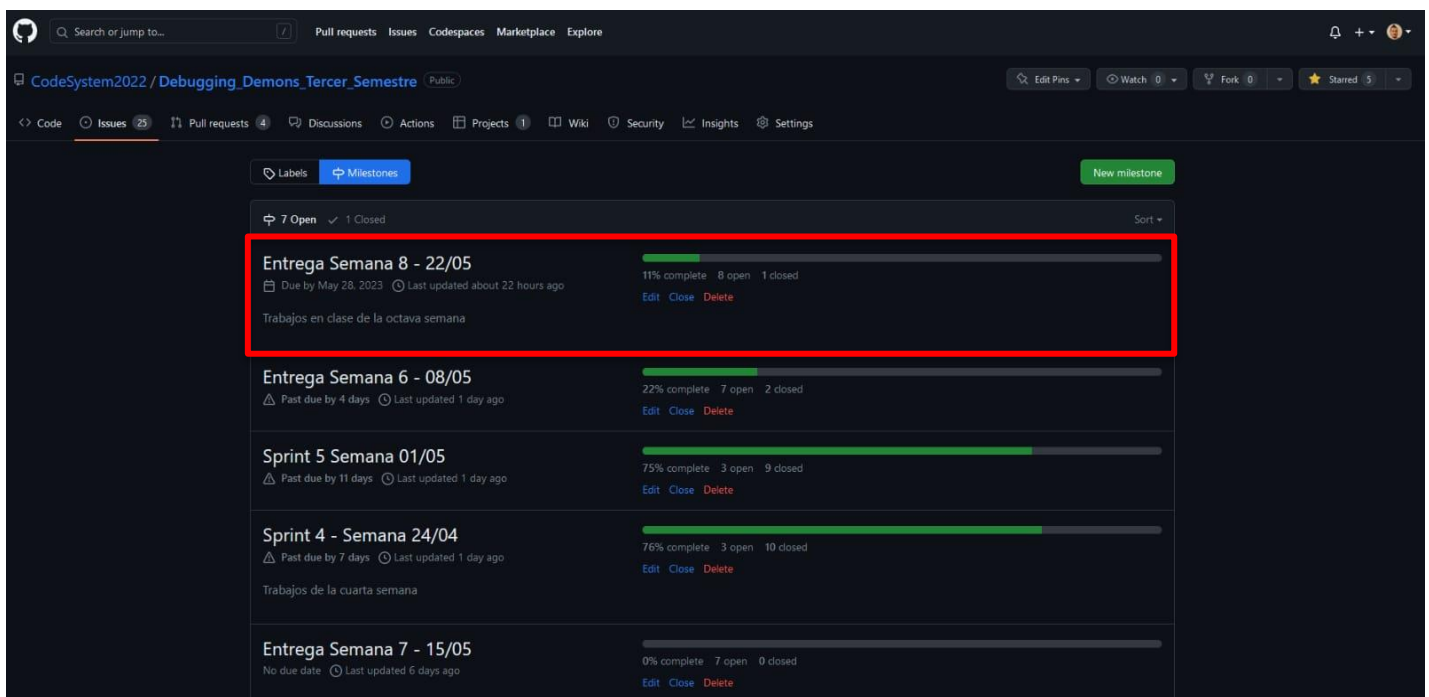
mayo de 2023

1 2 3 4 5 6 7  
8 9 10 11 12 13 14  
15 16 17 18 19 20 21  
22 23 24 25 26 27 28  
29 30 31 1 2 3 4

Limpiar

Cancel Close milestone Save changes

Ingresamos el nombre(1) de la **MILESTONE**, asignándole la semana correspondiente como nombre, y establecemos la fecha(2) de entrega de la tarea, finalmente guardamos los cambios(3).



CodeSystem2022 / Debugging\_Demons\_Tercer\_Semestre

Labels Milestones

New milestone

7 Open 1 Closed

Entrega Semana 8 - 22/05  
Due by May 28, 2023 Last updated about 22 hours ago  
11% complete 8 open 1 closed  
Edit Close Delete

Trabajos en clase de la octava semana

Entrega Semana 6 - 08/05  
Past due by 4 days Last updated 1 day ago  
22% complete 7 open 2 closed  
Edit Close Delete

Sprint 5 Semana 01/05  
Past due by 11 days Last updated 1 day ago  
75% complete 3 open 9 closed  
Edit Close Delete

Sprint 4 - Semana 24/04  
Past due by 7 days Last updated 1 day ago  
76% complete 3 open 10 closed  
Edit Close Delete

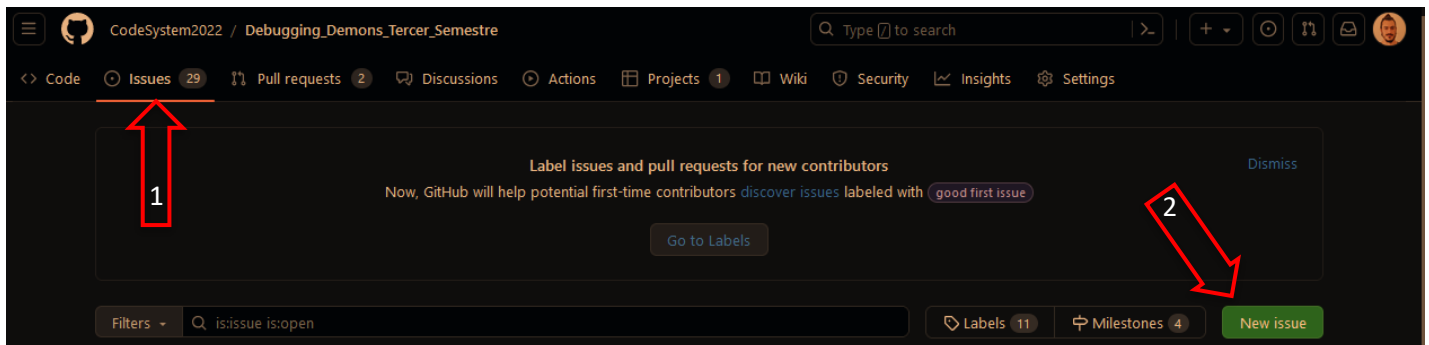
Trabajos de la cuarta semana

Entrega Semana 7 - 15/05  
No due date Last updated 6 days ago  
0% complete 7 open 0 closed  
Edit Close Delete

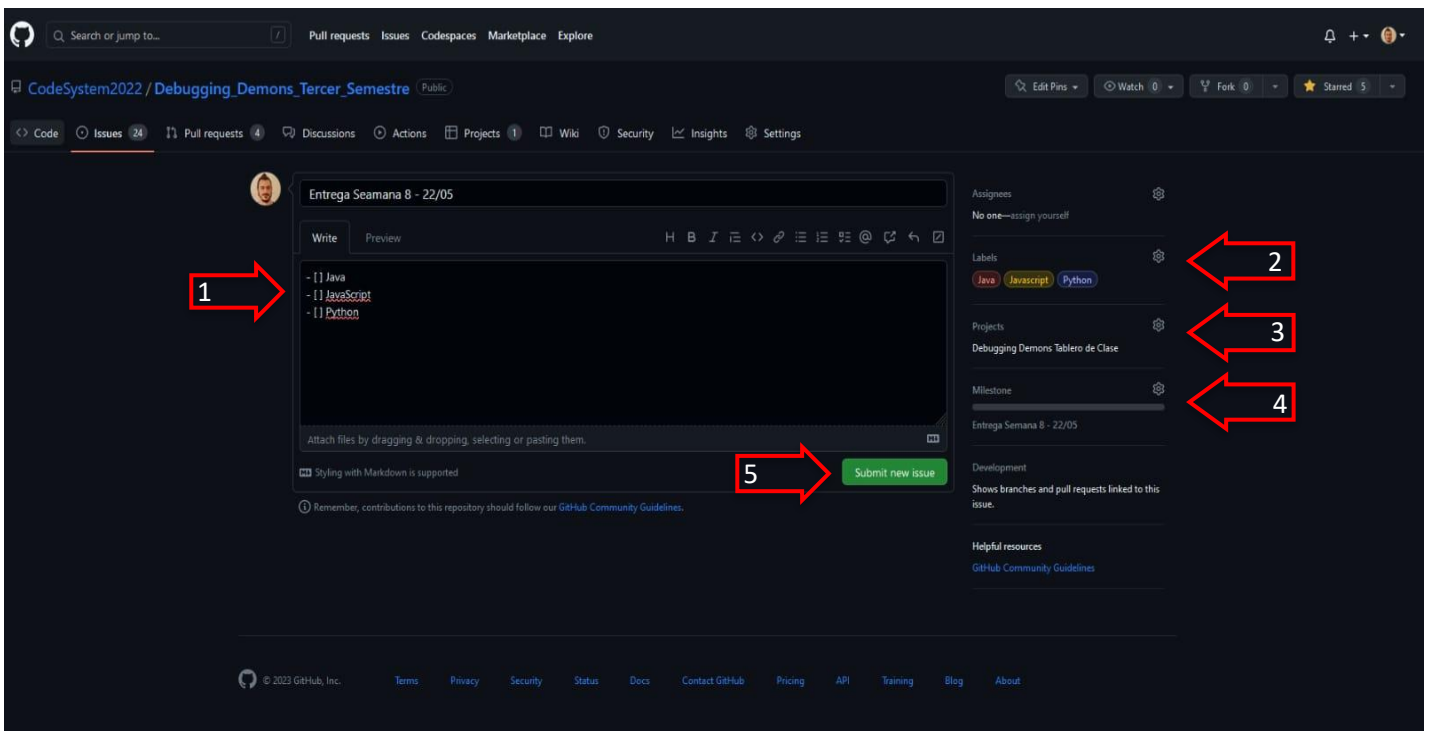
Una vez que hemos guardado los cambios, aparecerá la **MILESTONE** que creamos en el listado de **MILESTONES**.

## 2) CREAR LA ISSUE DE LA SEMANA

Las **ISSUES** en **GitHub** son herramientas que permiten realizar un seguimiento de tareas, problemas y solicitudes relacionadas con un proyecto. Se crean para informar sobre problemas, solicitar características o cualquier trabajo relacionado. Las **ISSUES** se pueden asignar a miembros del equipo, se pueden comentar y discutir, y se pueden etiquetar para organizarlas. Una vez que se resuelve una **ISSUE**, se puede cerrar. Las **ISSUES** son fundamentales para la comunicación y colaboración en GitHub, y ayudan a mantener un flujo de trabajo organizado y transparente.)



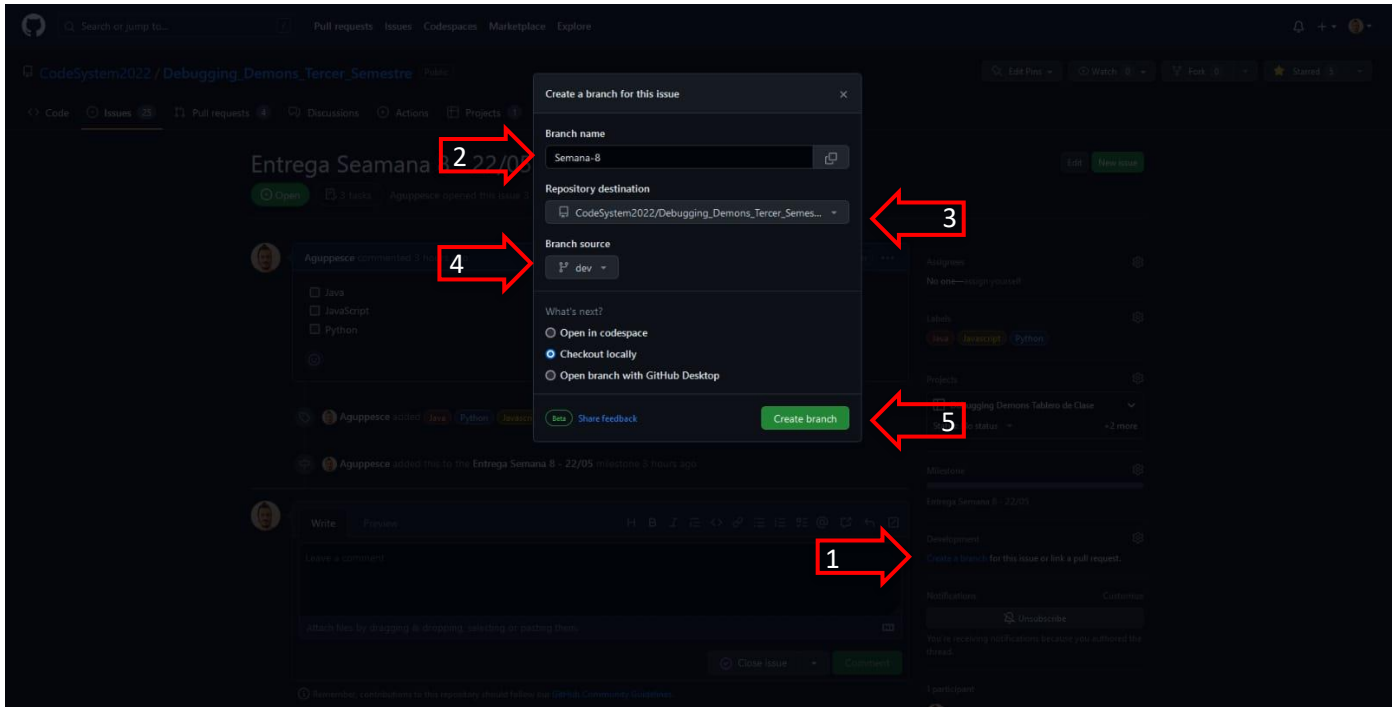
Nos dirigimos a la pestaña de **ISSUE(1)** para luego crear una **Nueva ISSUE(2)**.



Esta **ISSUE** no se la asignamos a nadie en particular, ya que la utilizamos para crear la **BRANCH** principal de la **Semana 8**.

Establecemos los **CHECKBOX(1)**, para cada uno de los ejercicios de cada lenguaje a subir, los **LABELS(3)**, el proyecto(3) al cual corresponde la **ISSUE**, luego asignamos el **MILESTONE(4)**, y finalmente confirmamos la nueva **ISSUE(5)**.

## 3) CREACIÓN DE LA BRANCH SEMANA-8



En la sección **DEVELOPMENT**, elegimos la opción **CREATE BRANCH(1)**, se desplegará un cuadro de dialogo para crear la rama para esta **ISSUE**, ingresamos el nombre de la rama(2), luego la asignamos al repositorio(3) de nuestro proyecto, seleccionamos la rama de la cual va a derivar, en este caso **DEV(4)** y procedemos a crear la rama(5).

## 4) CREAR LA CARPETA DE LA SEMANA Y PONER UN ARCHIVO .TXT

En local creamos la carpeta de la semana y dentro un archivo **".txt"**. Esto lo hacemos para que una vez creada la carpeta y la subamos **GIT**, esta sea visible para los demás integrantes del equipo, ya que si subimos la carpeta vacía no será visible hasta que contenga algún archivo (**GIT** no permite la gestión de carpetas vacías).

```

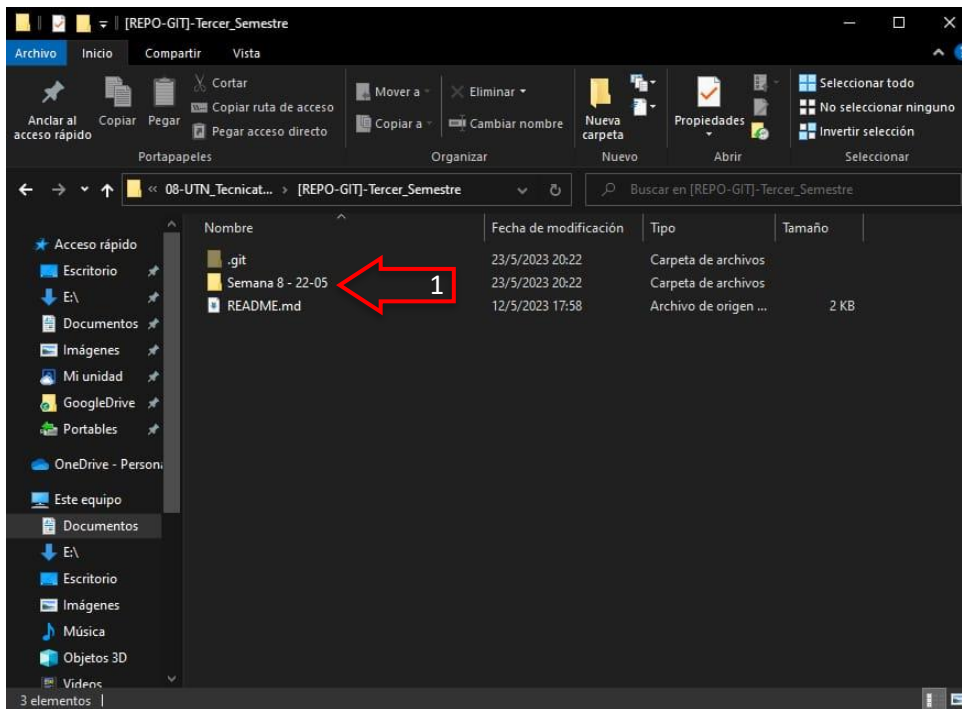
Agu@Admin MINGW64 /d/Documentos/01-Desarrollo/08-UTN_Tecnicatura/[REPO-GIT]-Tercer_Semestre (main)
$ git pull
From https://github.com/CodeSystem2022/Debugging_Demons_Tercer_Semestre
* [new branch]   Semana-8 -> origin/Semana-8
Already up to date.

Agu@Admin MINGW64 /d/Documentos/01-Desarrollo/08-UTN_Tecnicatura/[REPO-GIT]-Tercer_Semestre (main)
$ git checkout Semana-8
Switched to a new branch 'Semana-8'
branch 'Semana-8' set up to track 'origin/Semana-8'.

Agu@Admin MINGW64 /d/Documentos/01-Desarrollo/08-UTN_Tecnicatura/[REPO-GIT]-Tercer_Semestre (Semana-8)
$ |

```

Abrimos la consola de **GIT BASH** sobre la carpeta del repositorio local, introducimos el comando **GIT PULL** para traer los cambios del repositorio, y nos movemos a la rama **Semana-8(1)**.



Una vez hemos ingresado a la carpeta del proyecto y nos hemos posicionado en la rama **Semana-8** creamos la carpeta de la semana(1) donde cada integrante va ir subiendo la carpeta con su nombre, una carpeta por cada lenguaje y sus respectivos ejercicios.

```

MINGW64:/d/Documentos/01-Desarrollo/08-UTN_Tecnica/[REPO-GIT]-Tercer_Semestre
Agu@Admin MINGW64 /d/Documentos/01-Desarrollo/08-UTN_Tecnica/[REPO-GIT]-Tercer_Semestre (Semana-8)
$ git status
On branch Semana-8
Your branch is up to date with 'origin/Semana-8'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Semana 8 - 22-05/

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

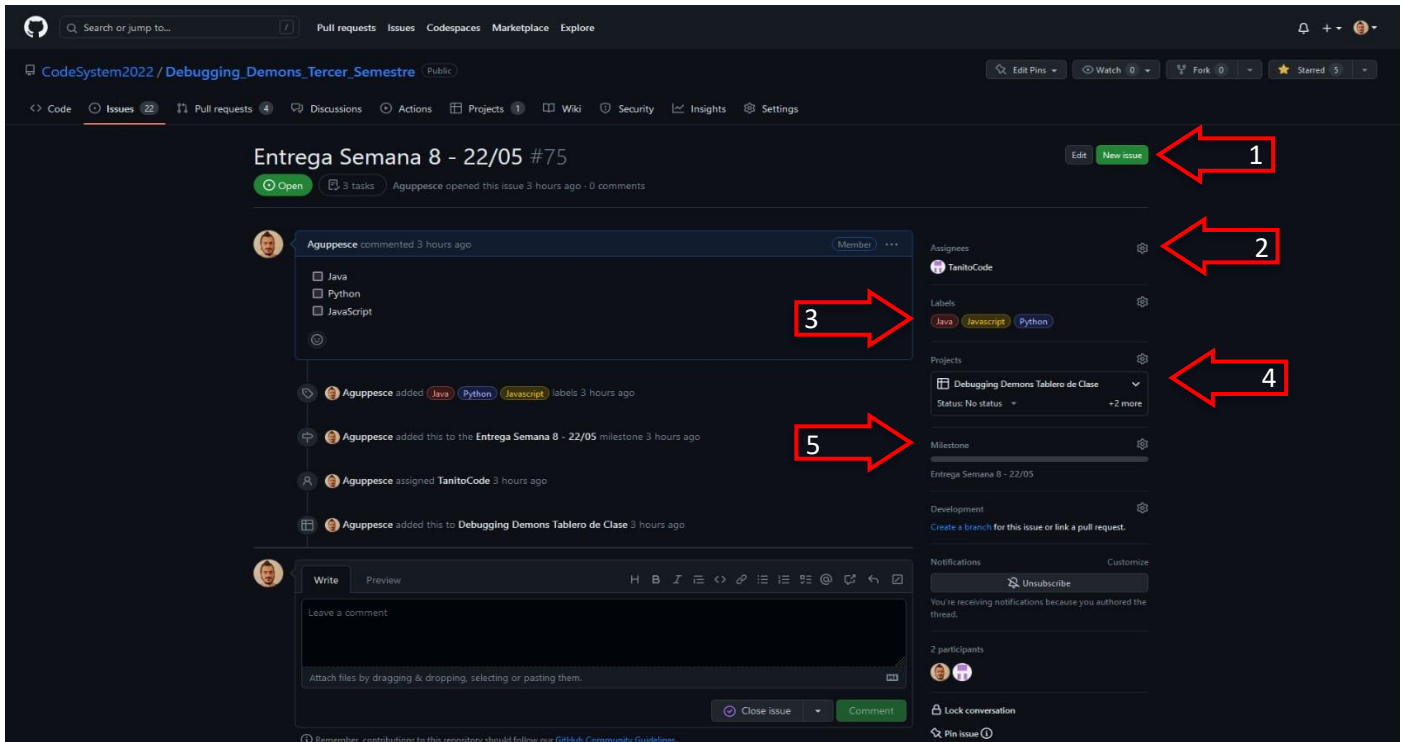
Agu@Admin MINGW64 /d/Documentos/01-Desarrollo/08-UTN_Tecnica/[REPO-GIT]-Tercer_Semestre (Semana-8)
$ git add .
Agu@Admin MINGW64 /d/Documentos/01-Desarrollo/08-UTN_Tecnica/[REPO-GIT]-Tercer_Semestre (Semana-8)
$ git commit -m "[ADD] Carpeta Semana 8"
[Semana-8 be557f5] [ADD] Carpeta Semana 8
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Semana 8 - 22-05/index.html
Agu@Admin MINGW64 /d/Documentos/01-Desarrollo/08-UTN_Tecnica/[REPO-GIT]-Tercer_Semestre (Semana-8)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 349 bytes | 116.00 KiB/s, done.
Total 4 (delta 1), reused 1 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/CodeSystem2022/Debugging_Demons_Tercer_Semestre.git
5e7f390..be557f5  Semana-8 -> Semana-8
$
  
```

Verificamos los cambios con el comando **GIT STATUS(1)**, luego agregamos los cambios realizados del directorio de trabajo actual al área de preparación (**STAGING AREA**) con **GIT ADD .(2)**, **confirmamos(3)** los cambio y escribimos un mensaje.

En esta parte se puede observar el archivo(4) que se creo en la carpeta “Semana 8 - 22-05” para que no quede vacía y pueda subirse a **GITHUB**. Finalmente enviamos los cambios al repositorio remoto con el comando **GIT PUSH(5)**.

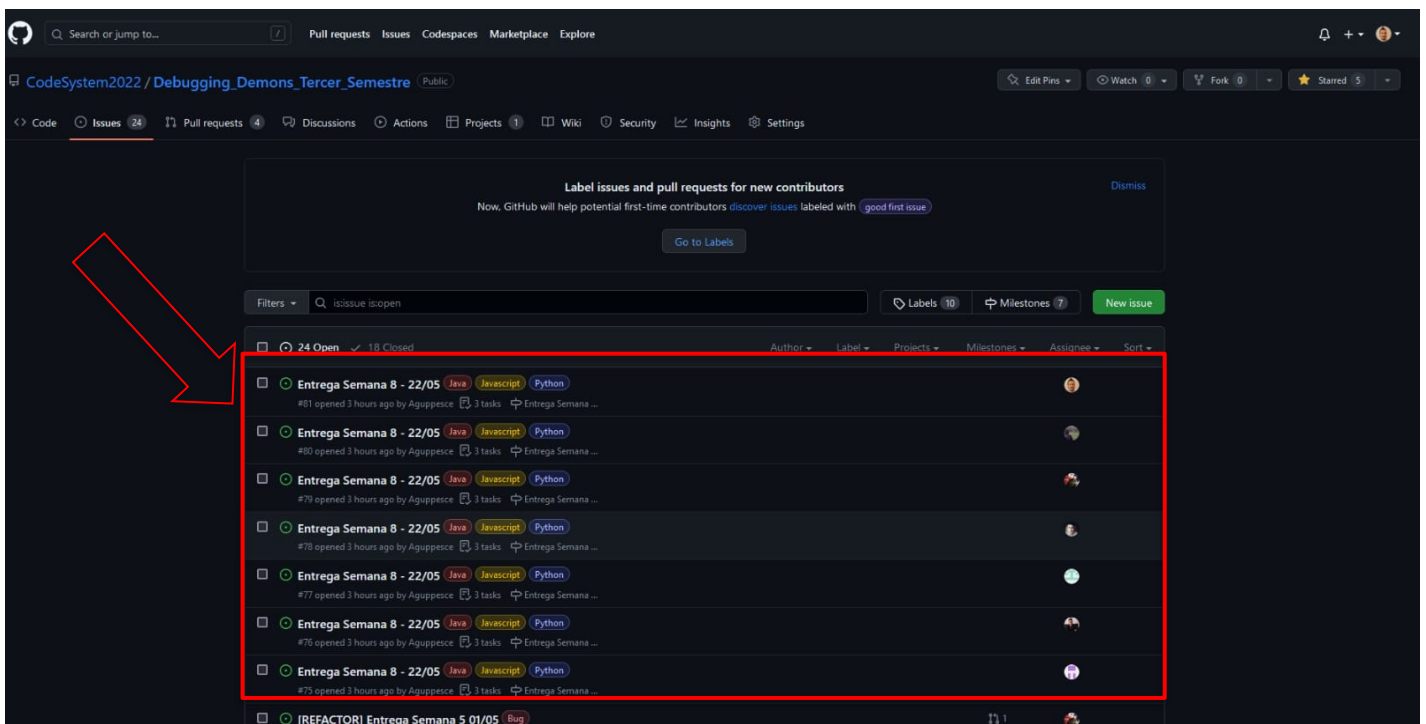


## 5) CREAR ISSUES PARA CADA UNO DE LOS MIEMBROS DEL EQUIPO



Aquí hacemos el mismo procedimiento que hicimos para crear la **ISSUE** principal de la semana, comenzando por crear una nueva **ISSUE**(1), solo que en este caso asignamos(2) una **ISSUE** para cada uno de los integrantes, cómo así también los **LABELS**(3), el proyecto(4) y el **MILESTONE**(5). Luego de crea la **ISSUE**, al integrante asignado le llegará un mail de que tiene una tarea por realizar.

En la siguiente imagen se visualizan todas las **ISSUE** asignadas para cada uno de los integrantes del equipo, incluido el **SCRUM MASTER**.



## TAREAS DEL EQUIPO EN LA SEMANA

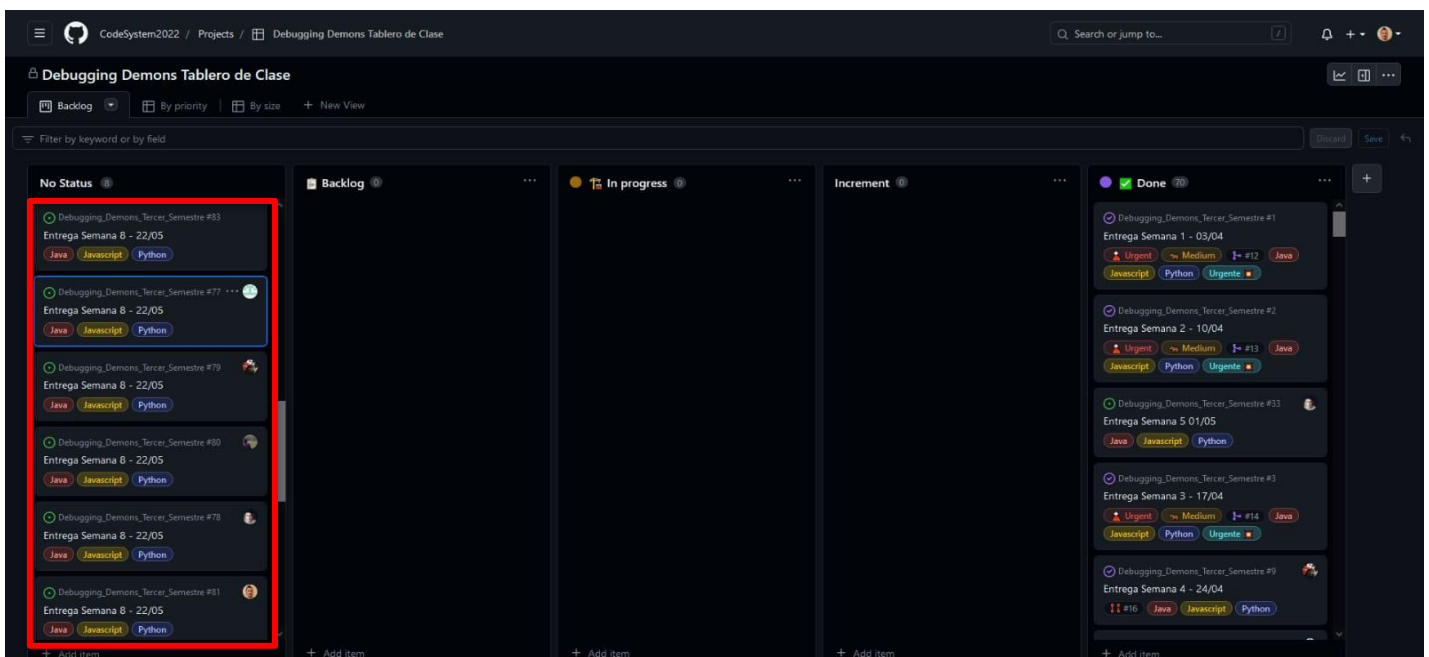
### DAILY (Reuniones diarias)

#### 6) APROBAR LOS PULL REQUEST, SOLICITAR REVIEWS DADO EL CASO.

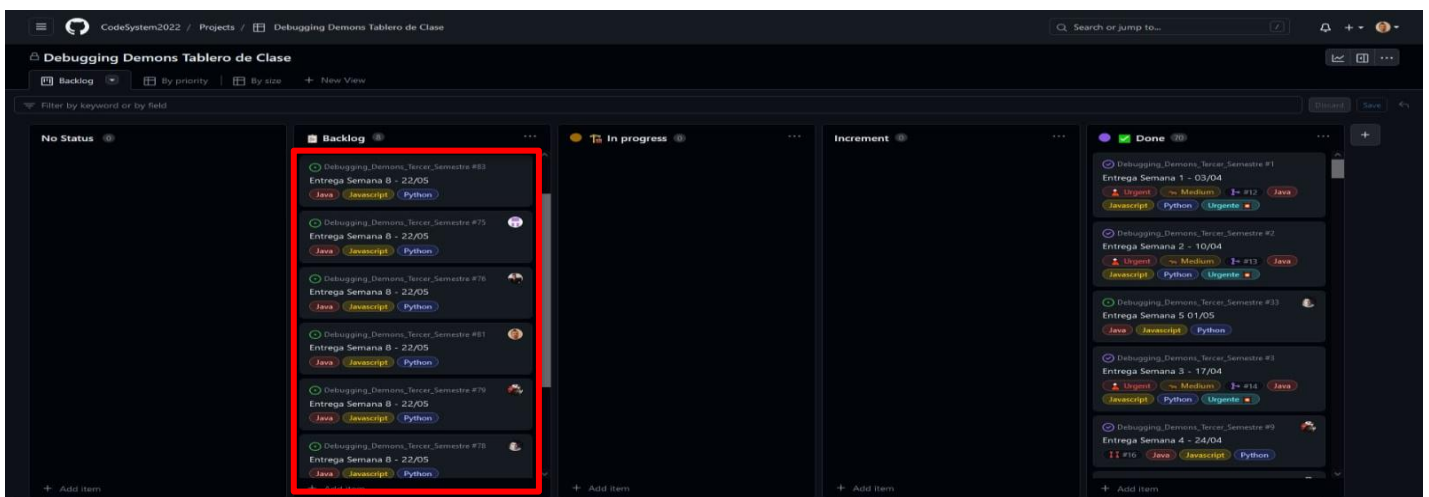
Organizamos una **PLANNING**, que es una reunión para planificar el “**SPRINT**” donde determinamos las tareas que le toca a cada uno y el tiempo que tiene para cumplirlas.

Estas reuniones son esenciales para el funcionamiento eficiente del equipo y permiten la colaboración y la resolución de problemas.

Una vez que a cada integrante se le asignó su tarea, controlamos en el tablero el progreso de estas, para ver el progreso por cada integrante.

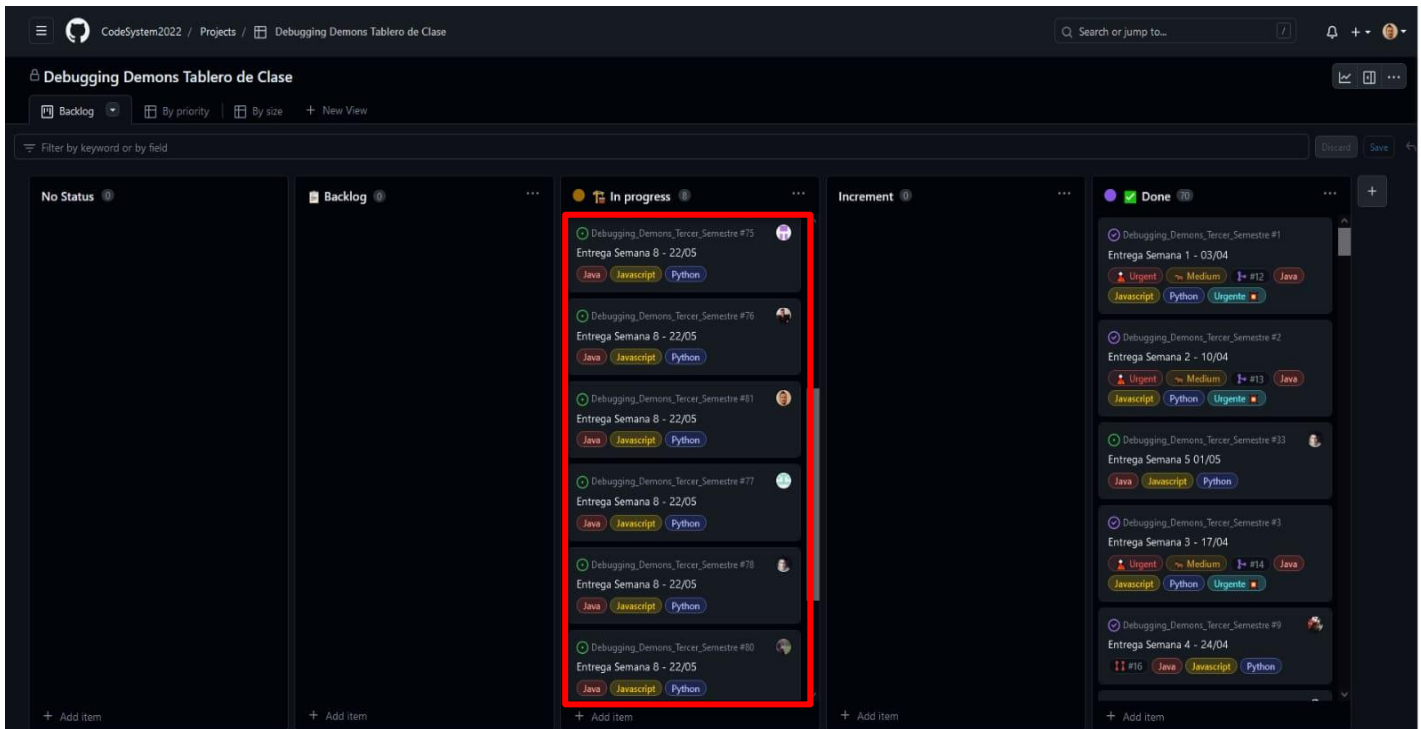


El uso del tablero nos proporciona una representación gráfica de los diferentes elementos del proyecto, como en este caso, las tareas de la semana, lo que nos permite tener un seguimiento del progreso de cada integrante al ir moviendo las tarjetas entre columnas.





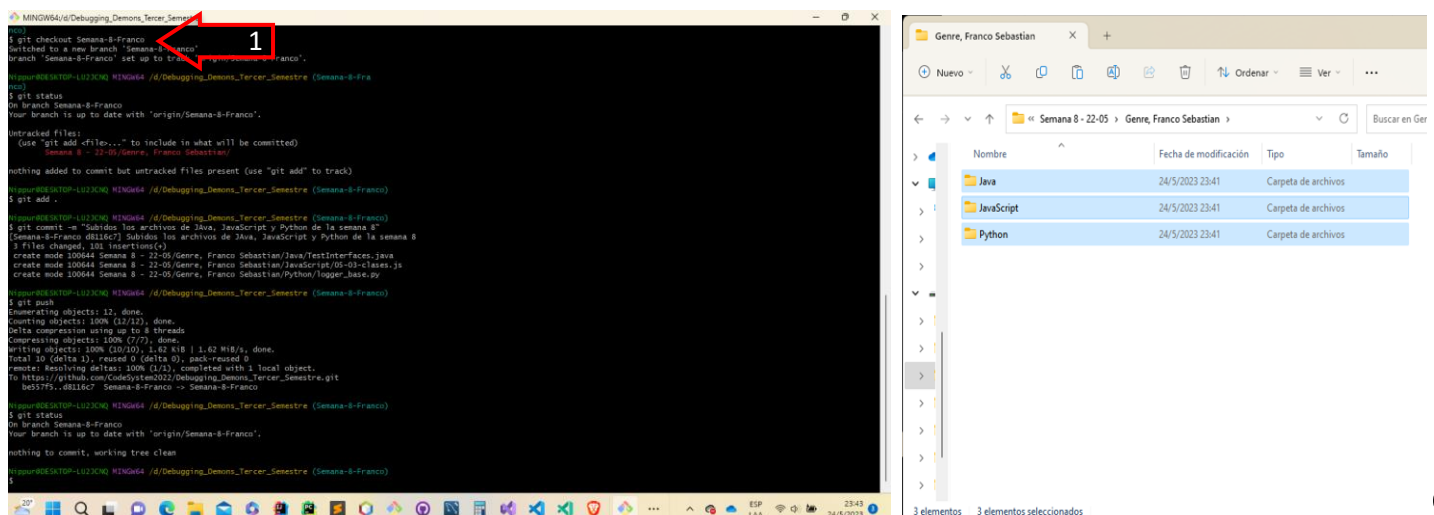
A medida que las tareas se van realizando se van moviendo a las diferentes columnas del tablero según el estado de la tarea.



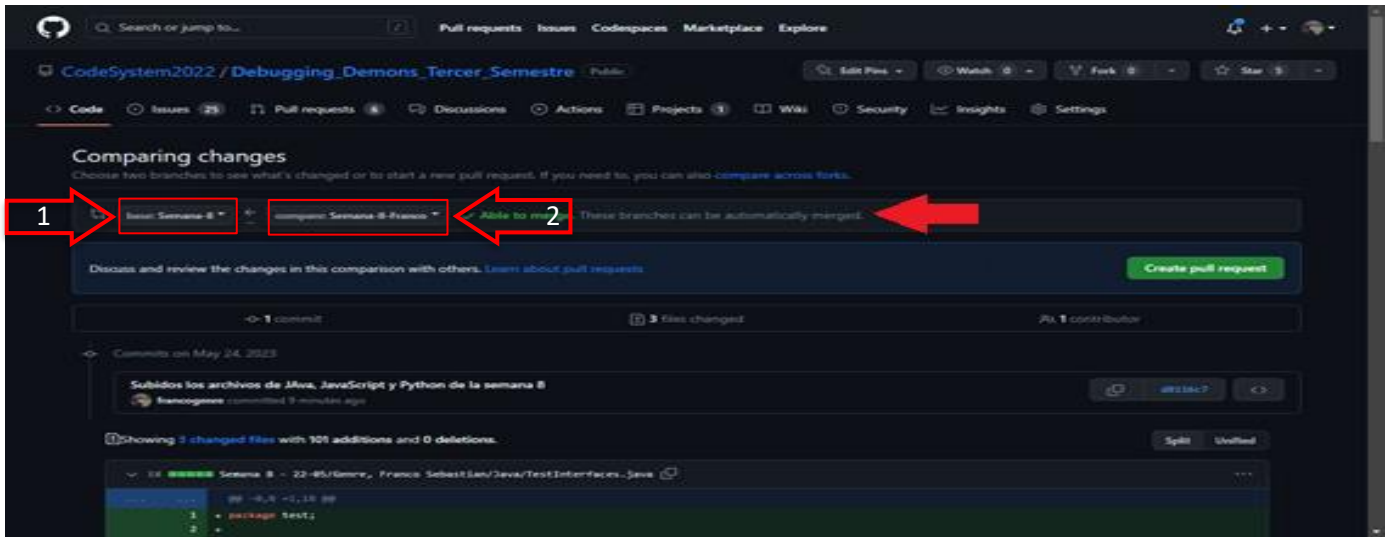
## ISSUE DE LA SEMANA DE: FRANCO GENRE

El procedimiento es el mismo para cada uno de los integrantes del equipo. Para evitar repetir imágenes y texto optamos por poner las imágenes de solamente uno de los integrantes sobre como se realiza este procedimiento, el cual consiste en crear las carpetas, luego agregar los archivos de cada lenguaje hechos durante la semana para luego desde **GIT BASH** agregar, confirmar y subir los archivos desde local.

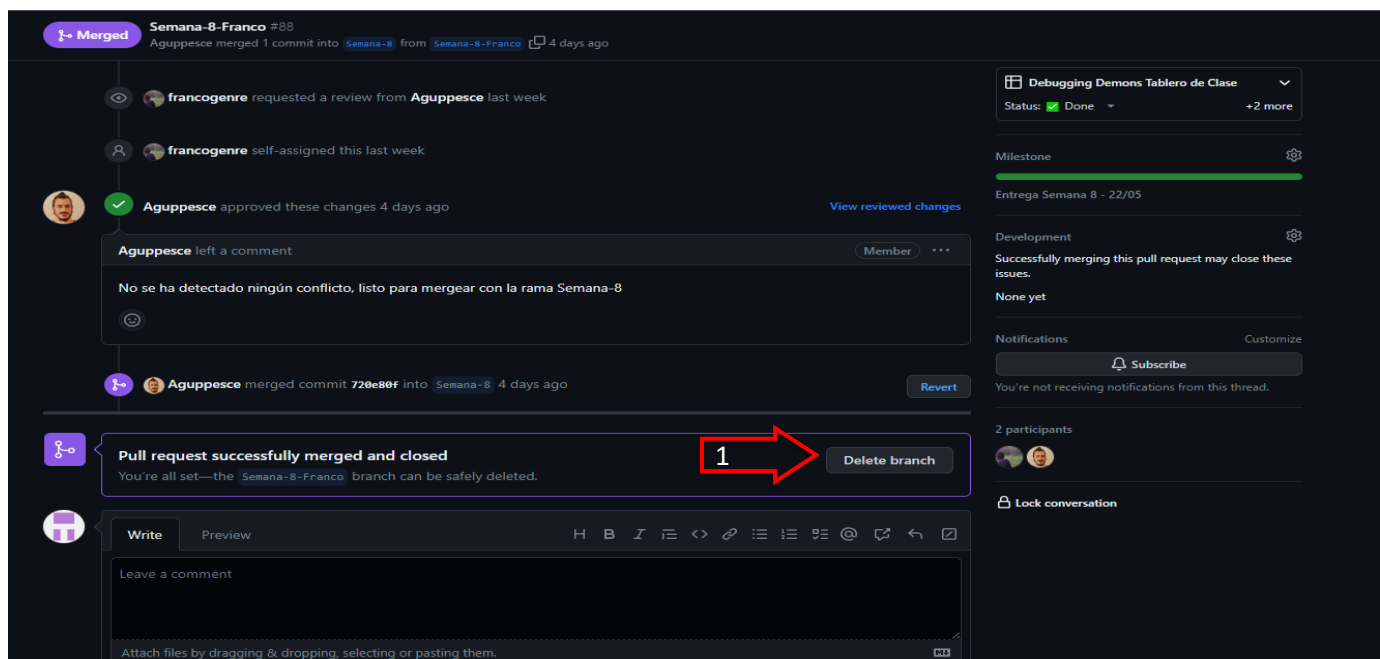
En esta imagen se muestra que una vez creadas las carpetas y agregado los archivos de cada lenguaje a sus respectivas carpetas, se introduce el comando **GIT PULL** y posteriormente se mueve a la rama de la **Semana-8-Franco(1)**, cabe aclarar que esta rama es la que se ha creado a partir de la **ISSUE** que se le asigno al integrante por medio de **GITHUB**.



Posteriormente se hace la solicitud del **PULL REQUEST** desde **GITHUB**, para que una vez que el **SCRUM MASTER** haya revisado el código se proceda a fusionar con la rama principal. En la solicitud del **MERGE** lo que hacemos es seleccionar la **rama base** que sería **Semana-8(1)** y la rama de el integrante donde ha subido los ejercicios solicitados en la **ISSUE** la cual es **Semana-8-Franco(2)**.



Se asigna al **SCRUM MASTER** para que haga el **REVIEW** (revisión), controle que todo este correctamente, y confirme el **MERGE**.



Como no existía ningún conflicto, se deja un mensaje de que esta listo para fusionar y se aprueba el **MERGE**.

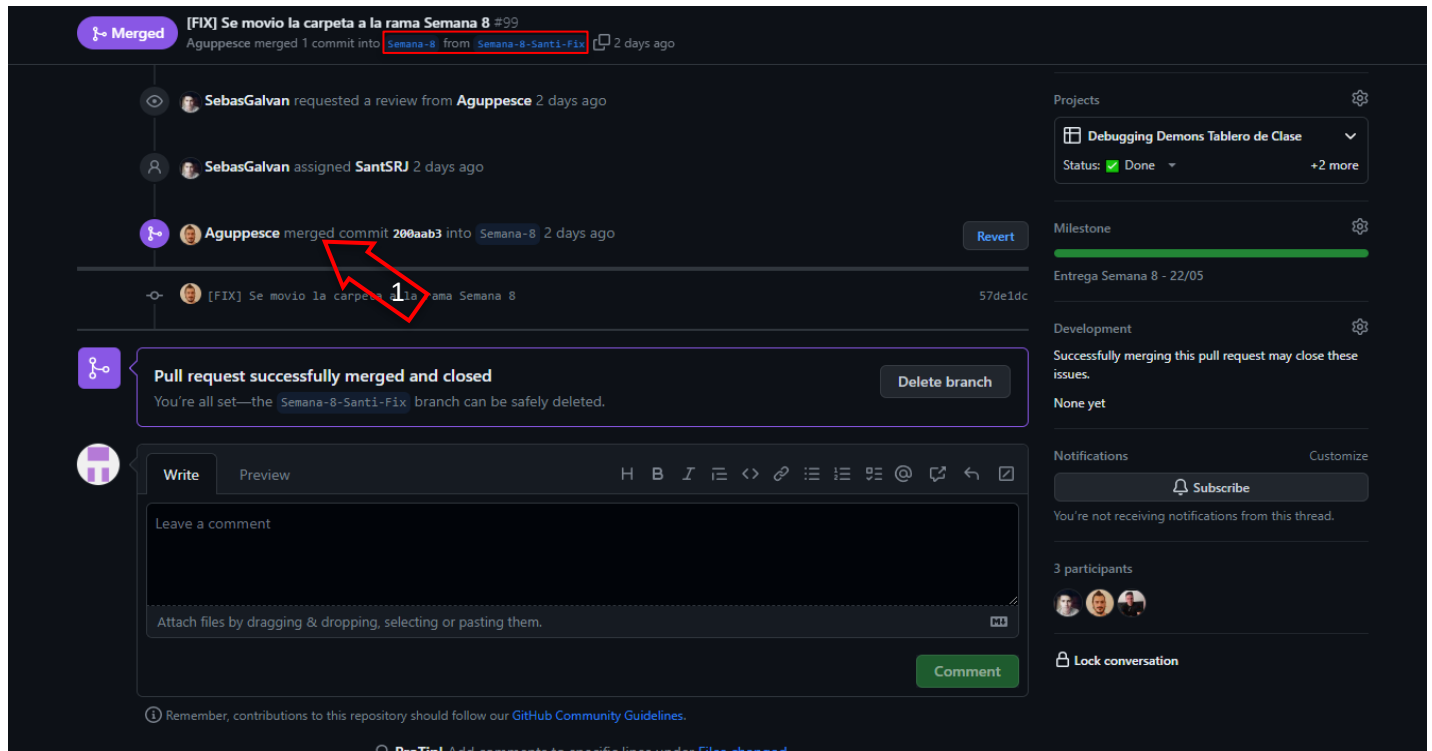
Algo importante para destacar también en este proceso, es que una vez que se ha realizado el **MERGE**, se deben eliminar la rama que se utilizó para trabajar sobre la **ISSUE** para no dejar residuos en el repositorio. Por lo tanto el **SCRUM MASTER** o el integrante asignado a la **ISSUE**, una vez que se ha hecho el **MERGE** y ha cerrado la **ISSUE**, debe eliminar(1) la rama, esto es posible hacerlo desde la misma pantalla del **PULL REQUEST**.

Con esto concluyen la/s tareas de la semana del integrante Franco Genre.

**ISSUE DE LA SEMANA DE: SANTIAGO MENDOZA**



En este caso hubo un conflicto al momento de controlar el **PULL REQUEST**, ya que el integrante subió las tareas a la rama de la **Semana-7**, y no a la rama de la **Semana-8**.

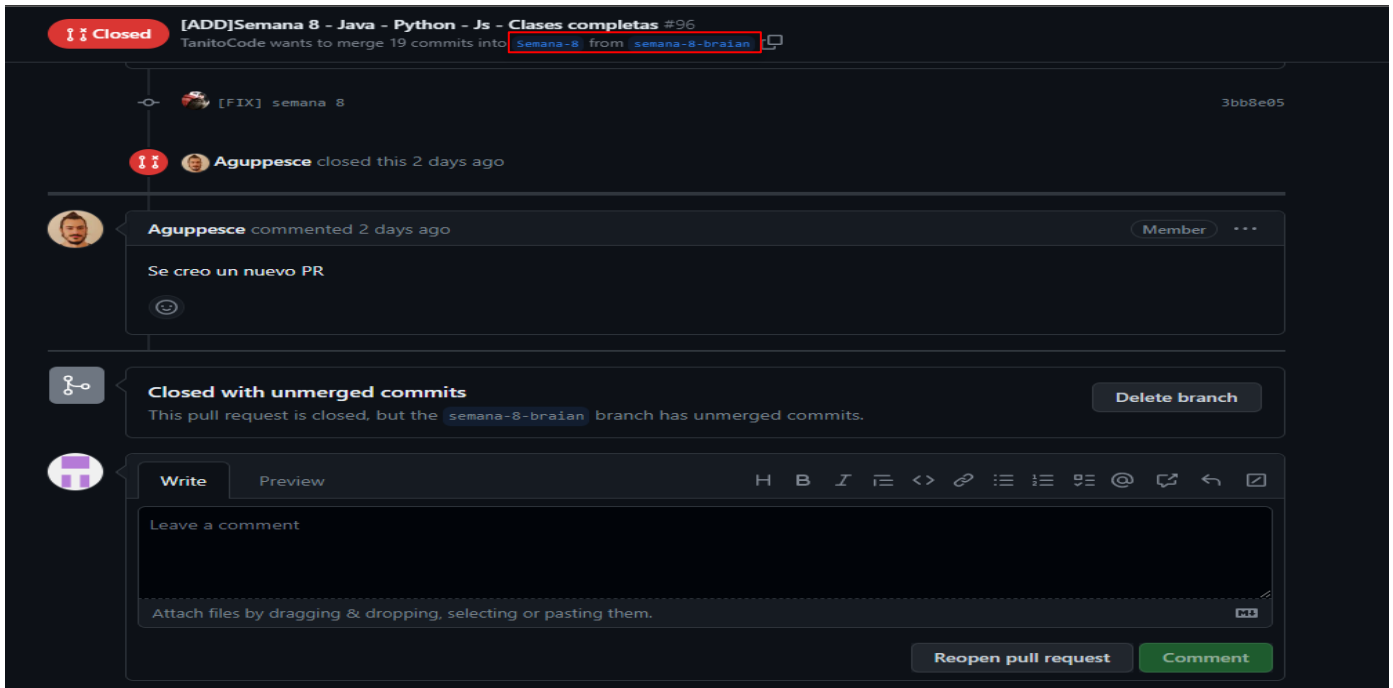


Para solucionar este conflicto organizamos una **DAILY** con el equipo, y discutimos como resolver este inconveniente, concluimos que lo mejor era crear otro **PULL REQUEST**, por lo cual movimos los archivos de la **Semana-7** a la **BRANCH** de la **Semana-8**, creando de por medio una rama que denominada **semana-8-santi-fix** para posteriormente aprobarla y realizar el **MERGE(1)**.

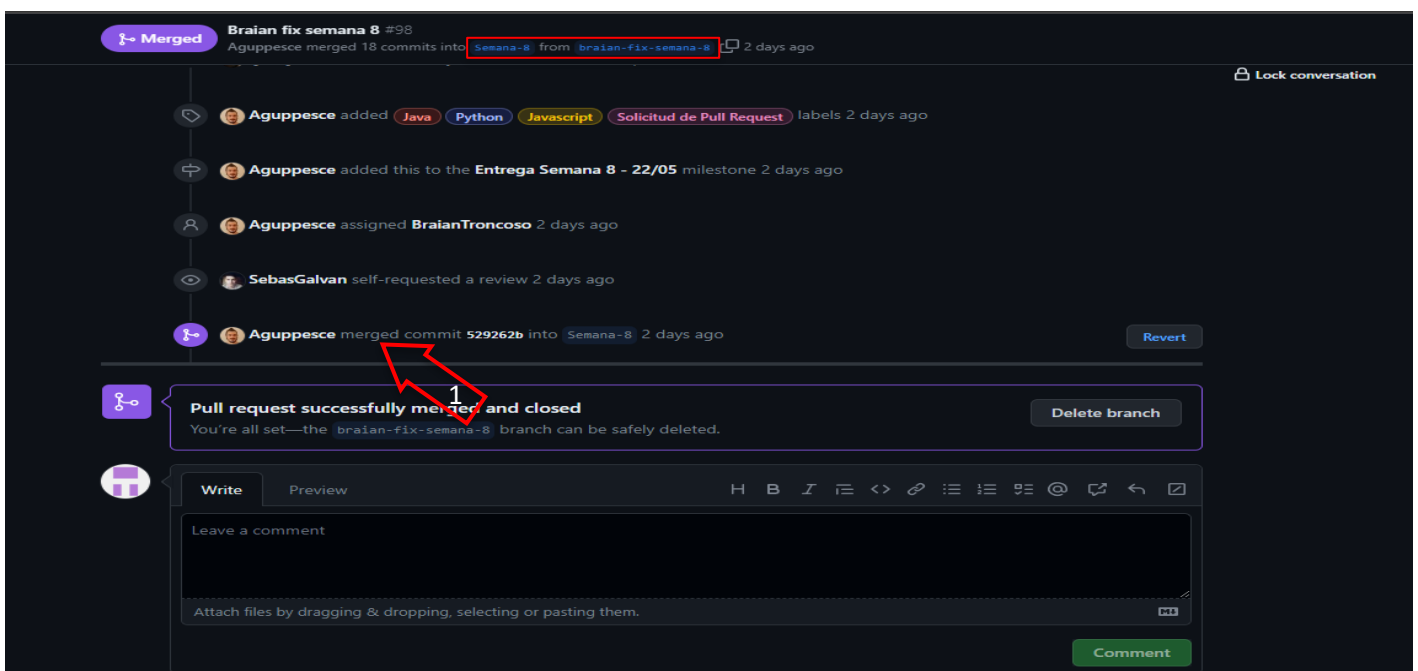


## ISSUE DE LA SEMANA DE: BRAIAN TRONCOSO

Al revisar el **PULL REQUEST** del compañero, esta rama traía los archivos de las clases de semanas anteriores que estaban en la rama **MAIN**. Si bien esto no creaba ninguna clase de problema en el código o en **GIT**, era un conflicto ya que iba a quedar código y carpetas replicadas. Por lo que se coordinó una **MEET** en la cual se concluyó que lo mejor era cerrar el **PULL REQUEST** sin hacer el **MERGE** y crear una nueva rama.



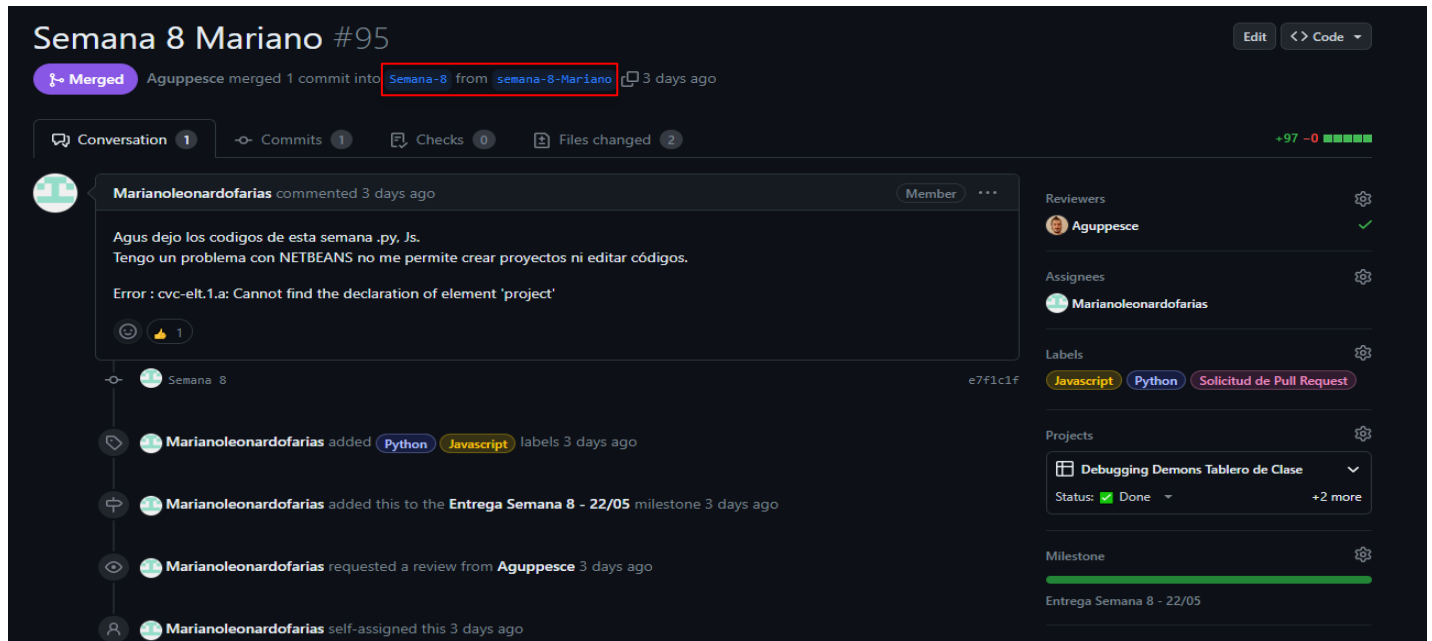
Esta nueva rama la nombramos como **"braian-fix-semana-8"**, y en esta solo pusimos los archivos de las clases correspondientes a la **Semana-8**, para posteriormente realizar el **MERGE(1)**.





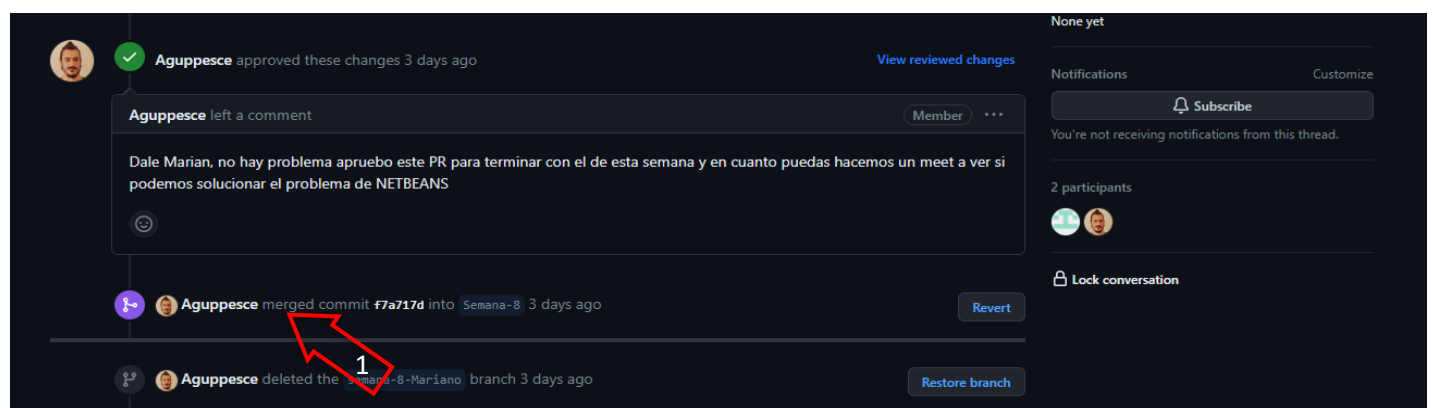
## ISSUE DE LA SEMANA DE: MARIANO FARIAS

En la solicitud del **PULL REQUEST**, el compañero explicaba que tuvo un conflicto con el **IDE NetBeans**, por lo que no pudo subir las clases de Java de la semana, por lo tanto se hizo el **MERGE** con las clases de Python y JavaScript y luego entre semana se pudo solucionar el problema con el programa.



The screenshot shows a GitHub Pull Request titled "Semana 8 Mariano #95". The status is "Merged". The description mentions a conflict with the IDE NetBeans and a merge with Python and JavaScript classes. The comment section shows a user reporting an error: "Error : cvc-elt.1.a: Cannot find the declaration of element 'project'". The right sidebar shows the pull request details, including labels like "Javascript", "Python", and "Solicitud de Pull Request", and a project named "Debugging Demons Tablero de Clase".

En la siguiente imagen se muestra un **FEEDBACK**, aclarando que se realizará el **MERGE(1)** y se procederá a solucionar la incidencia del compañero.

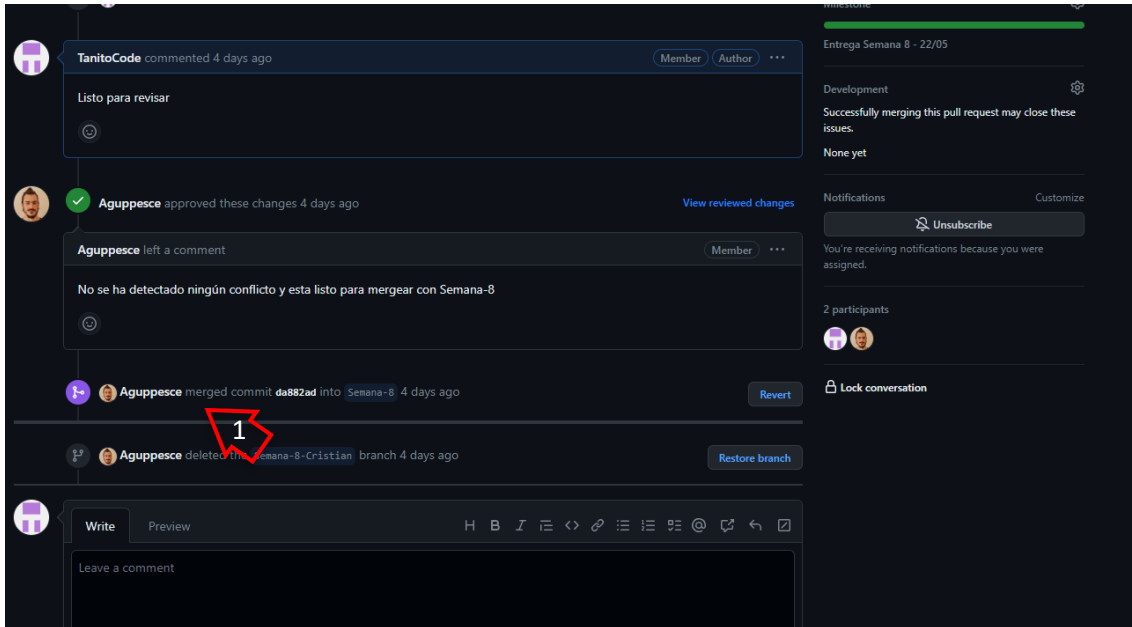


The screenshot shows a GitHub Pull Request #95 with a comment from "Aguppesce" stating: "Dale Marian, no hay problema apruebo este PR para terminar con el de esta semana y en cuanto puedas hacemos un meet a ver si podemos solucionar el problema de NETBEANS". Below the comment, a red arrow points to a merge action: "Aguppesce merged commit f7a717d into Semana-8". A red circle with the number "1" is placed over the merge action.



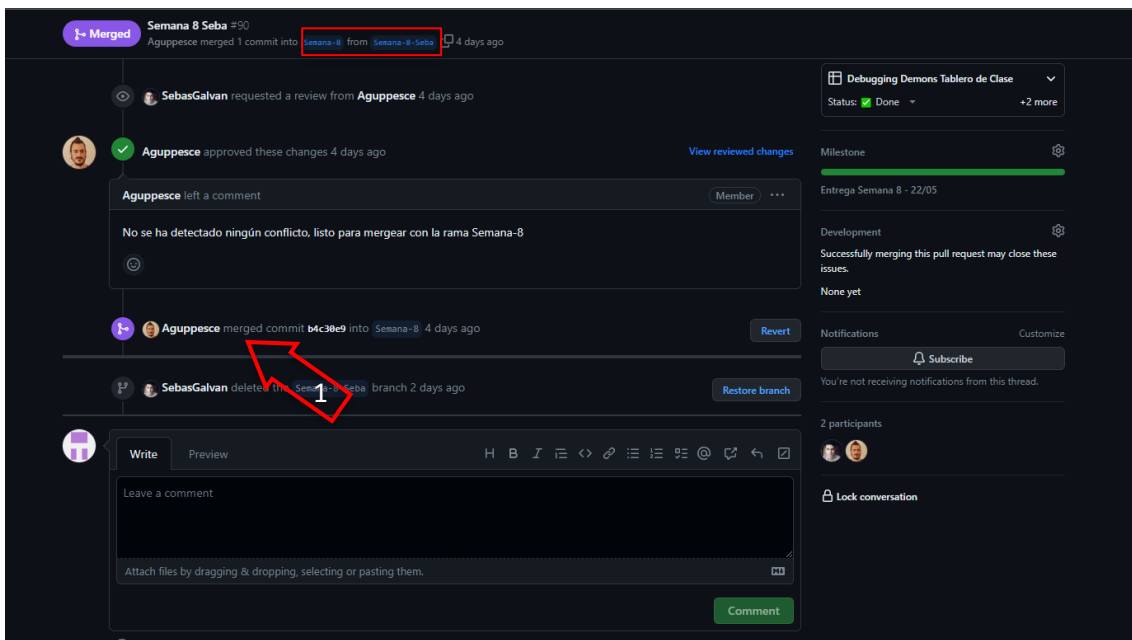
## ISSUE DE LA SEMANA DE: CRISTIAN SÁNCHEZ

Solicito el **PULL REQUEST** y no se encontró ningún conflicto y se procedió a confirmar el **MERGE(1)**.



## ISSUE DE LA SEMANA DE: SEBASTIÁN GALVÁN

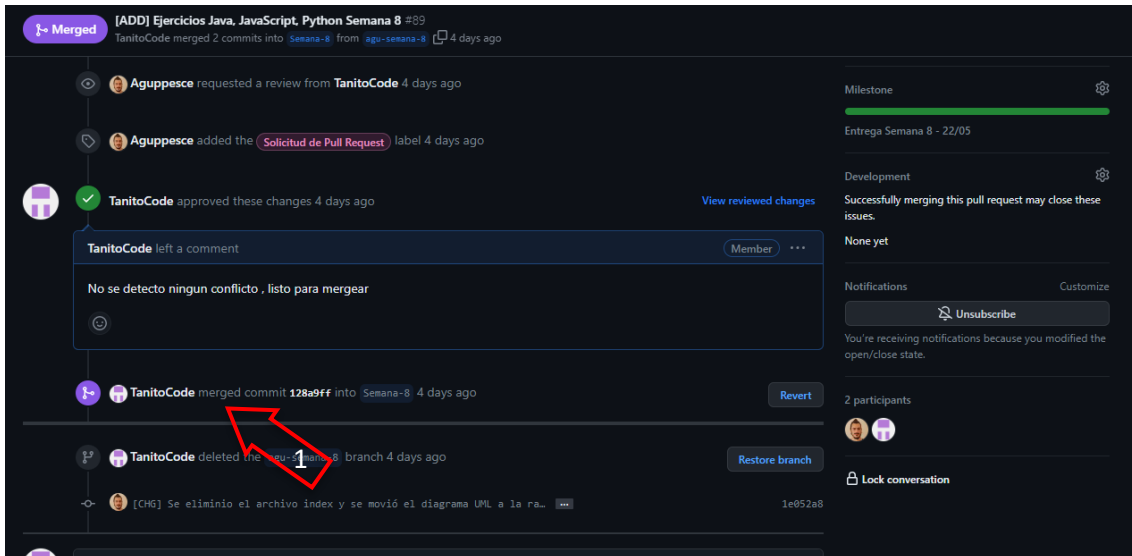
Solicito el **PULL REQUEST** y al no encontrarse ningún conflicto se procedió a confirmar el **MERGE(1)**.



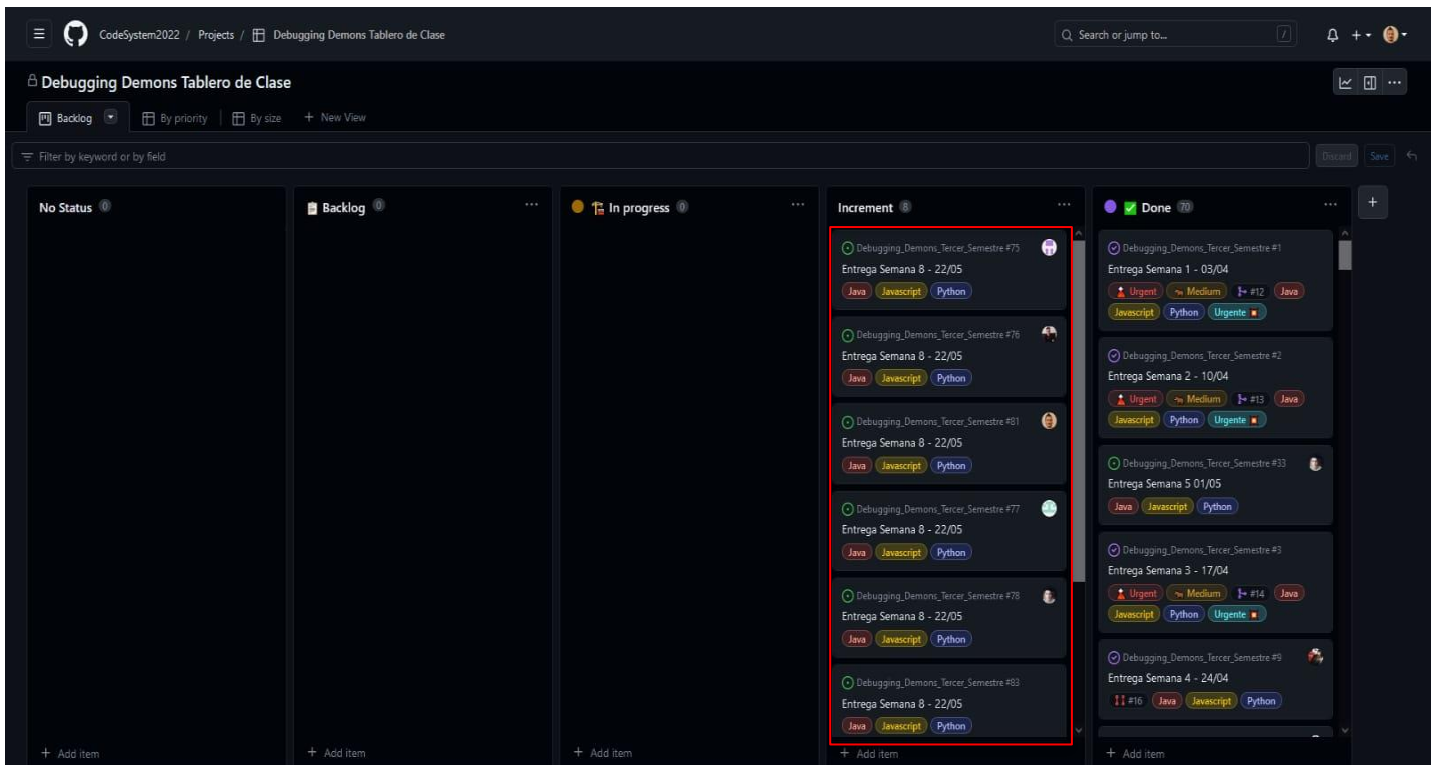


## ISSUE DE LA SEMANA DE: AGUSTÍN PEREZ PESCE

El **SCRUM MASTER** subió las tareas de la semana y le solicito a uno de los integrantes para que lo revise, y al no encontrarse ningún conflicto se aprobó el **MERGE(1)**.



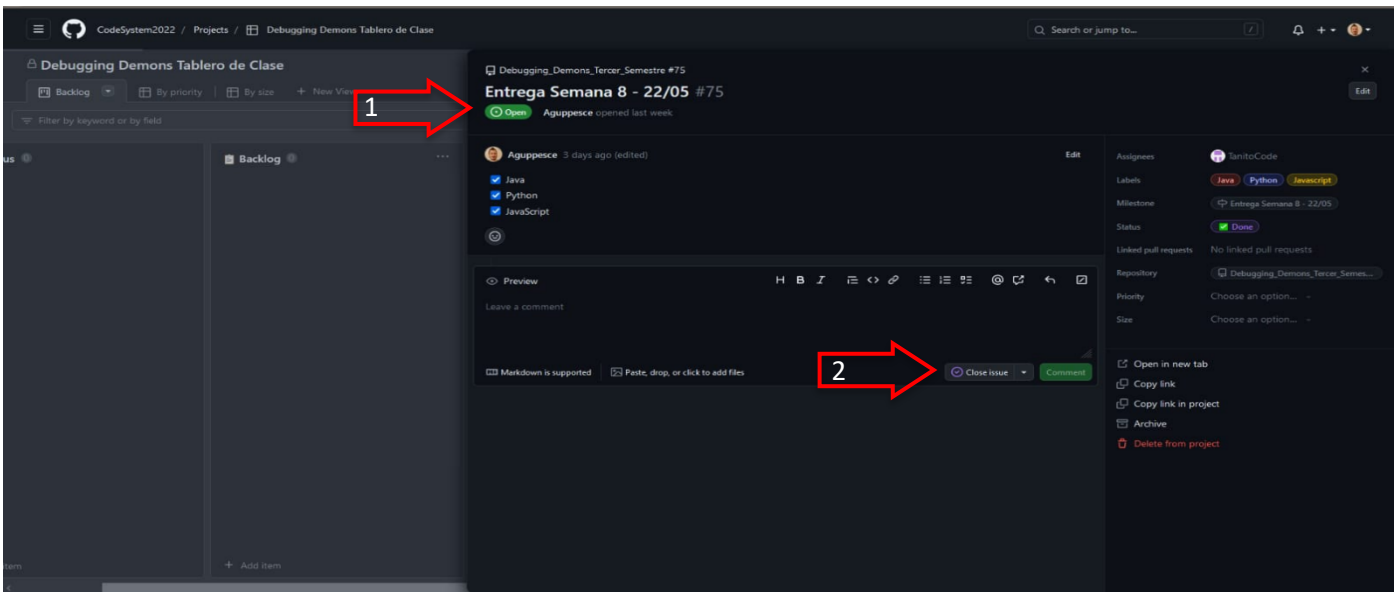
Una vez finalizadas las **ISSUES**, cada integrante mueve la tarea de la columna **"IN PROGRESS"** a la columna **"INCREMENT"** en donde posteriormente el **SCRUM MASTER** cerrará una por una las tareas de cada integrante.



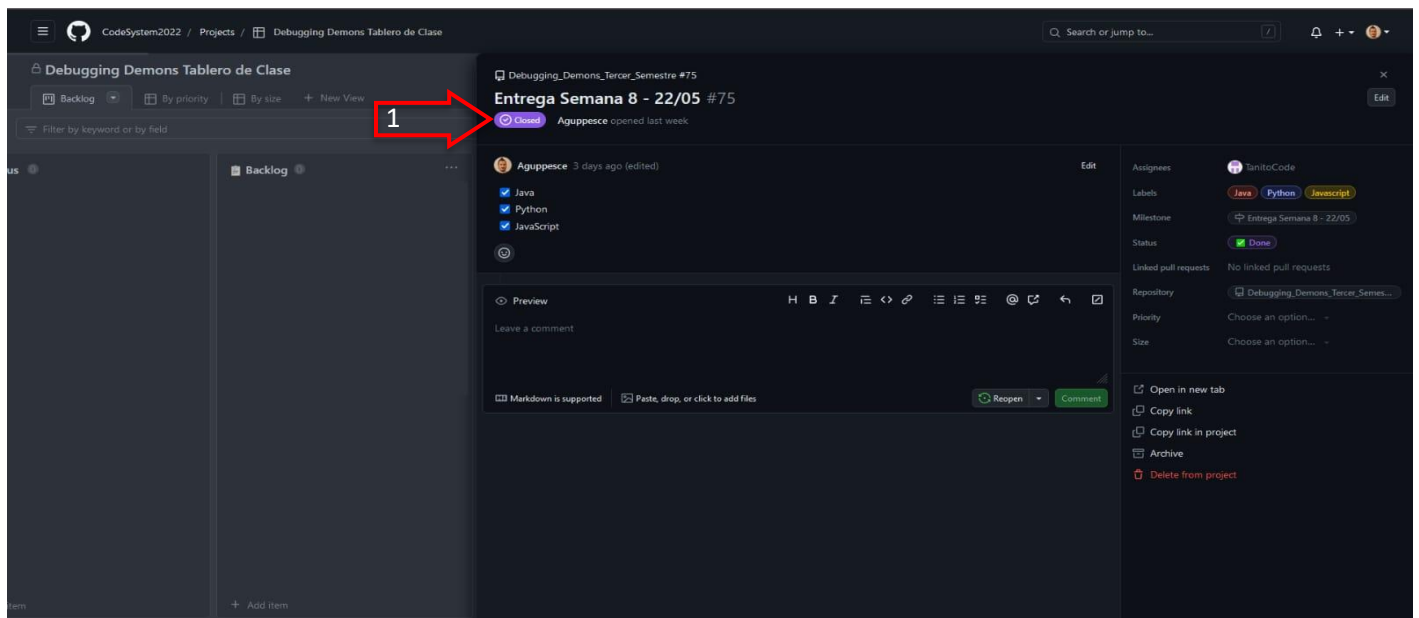
## FINALIZANDO LAS TAREAS DE LA SEMANA DEL EQUIPO

### 7) CERRAR ISSUE DE LA SEMANA, HACER EL MERGE A DEV.

Una vez que la tarea esta en la columna **"IN PROGRESS"** el **scrum master** ingresa a la **ISSUE** desde el tablero. En la imagen se observa que la **ISSUE** aún esta **abierta(1)**, por lo tanto aquí el **SCRUM MASTER** debe cerrar la **ISSUE(2)** para que la tarea, automáticamente, se mueva de la columna **IN PROGRESS** a la columna **DONE**.

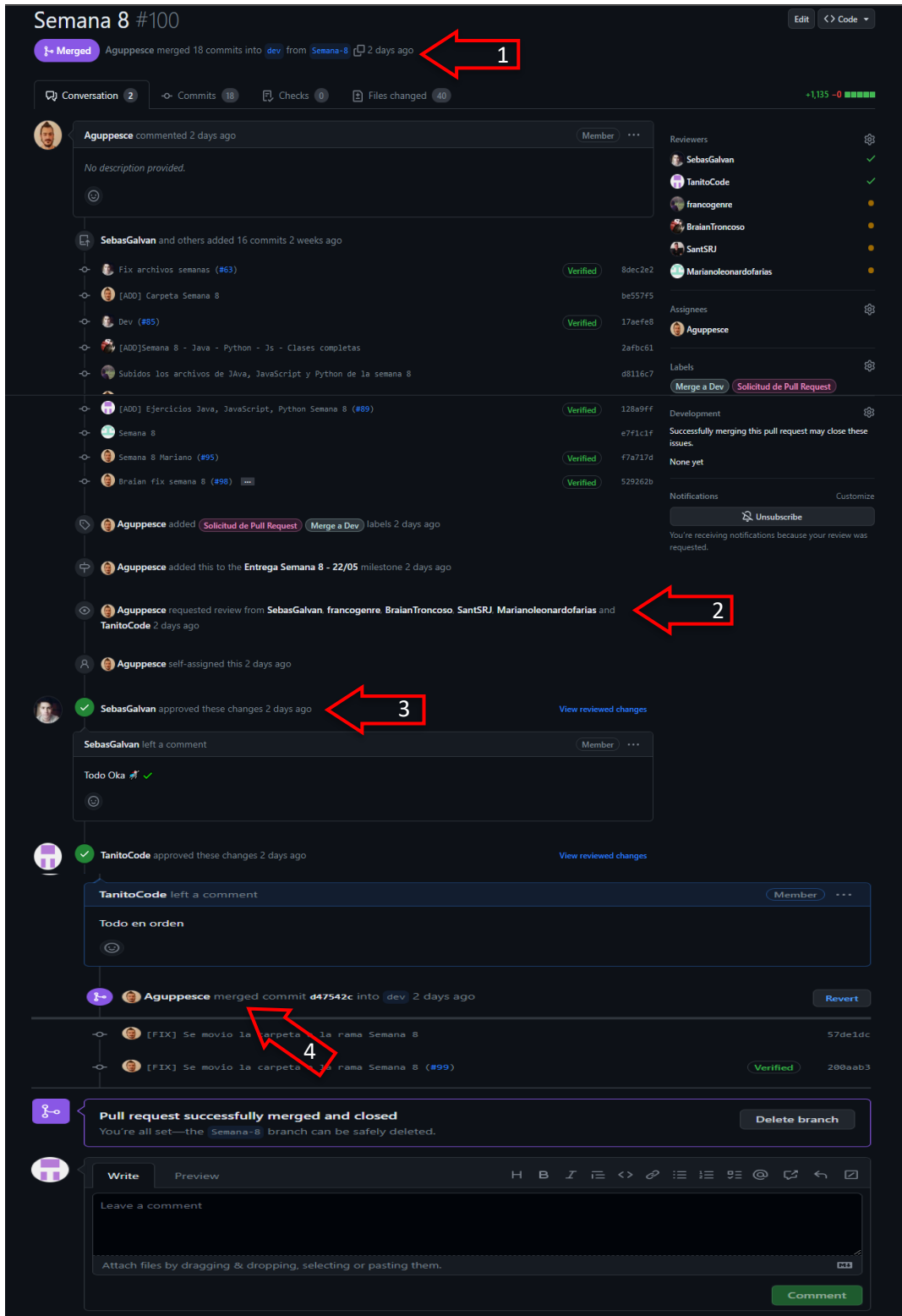


En esta imagen se puede observar que una vez que la **ISSUE** ha sido cerrada se cambia el estado de **OPEN** a **CLOSED(1)**.





El **SCRUM MASTER** procede a hacer un **MERGE** de la rama de la **Semana-8** en la cual están todas las tareas realizadas por los integrantes, con la rama **DEV(1)**. Esta es una imagen importante ya que se muestra todos los **COMMITTS** que se han hecho en la rama “Semana-8”.



The screenshot shows a GitHub Pull Request titled "Semana 8 #100". The interface includes a header with "Merged" status, a list of commits, and a sidebar with reviewers and assignees. Red arrows and numbers highlight key steps: 1 points to the merge commit message, 2 points to the review request, 3 points to the approval, and 4 points to the merge action.

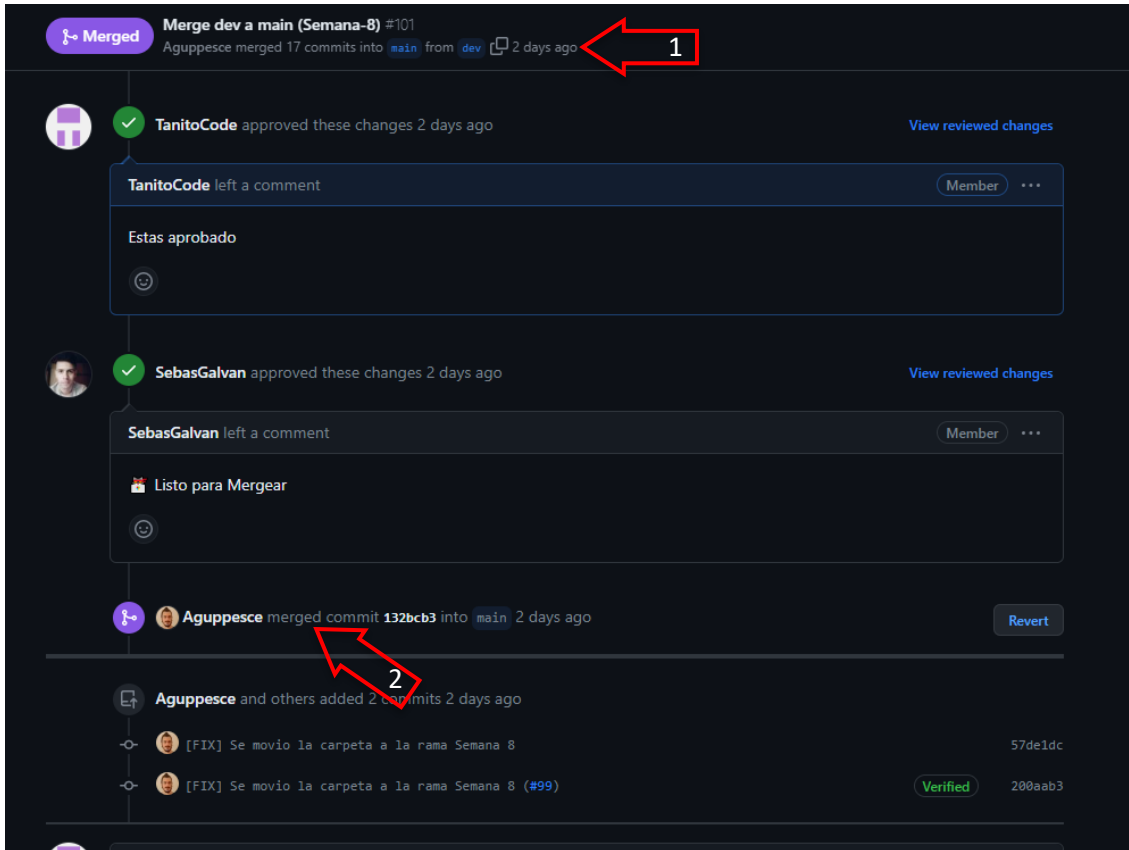
Se solicita la revisión del **PULL REQUEST** a los integrantes del equipo(2) para hacer el **MERGE** de la rama **DEV** con **Semana-8**.

Posteriormente uno o más integrantes(3) aprueban el **PULL REQUEST**, y se procede a realizar el **MERGE(4)**.



## 8) HACER EL MERGE A MAIN.

En esta imagen se visualiza el **PULL REQUEST** para hacer el **MERGE** de la rama **DEV** a la rama **MAIN(1)**, repetimos el mismo proceso que en la imagen anterior solicitando la revisión y posteriormente aprobación de uno o más de los integrantes del equipo y finalmente se hace el **MERGE(2)**.





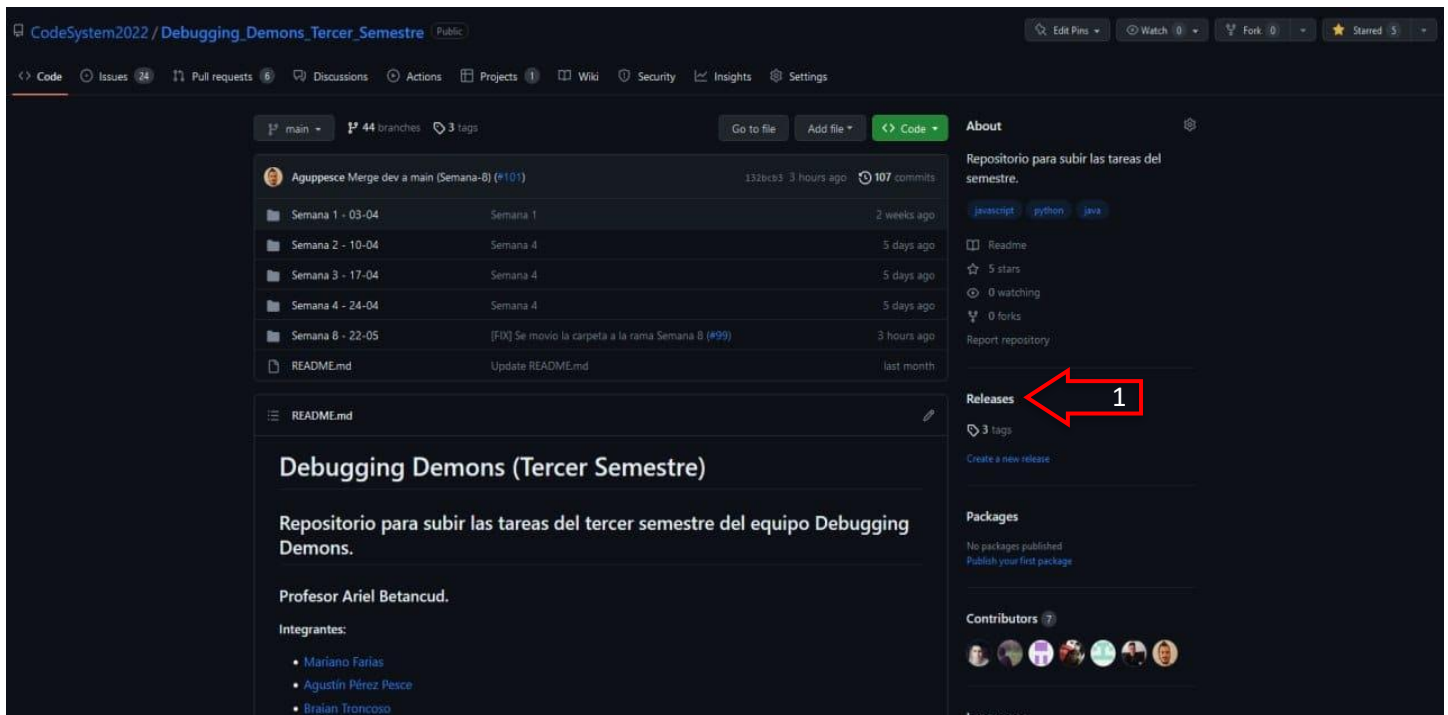
## 9) CREAR EL TAG DE LA SEMANA AL CERRAR EL SPRINT.

En la **DAILY** final de la semana, se discutieron los últimos detalles del proceso y al ser la última semana del mes se procedió a crear el **"RELEASE"**.

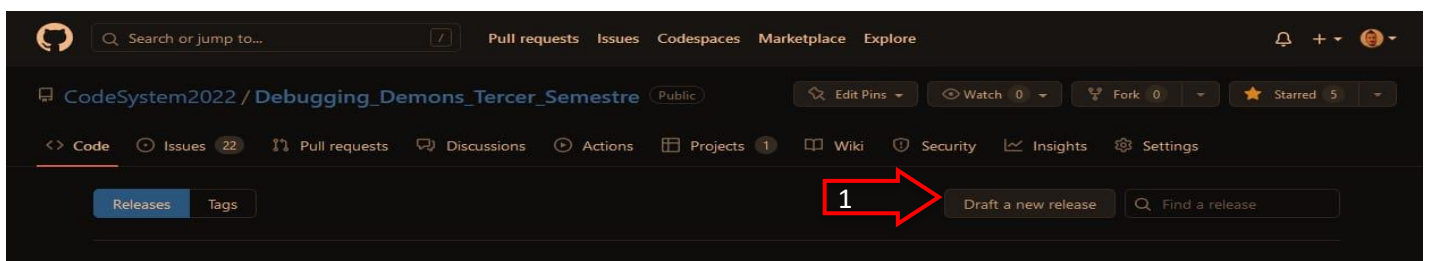
Este es un paso importante, ya que, si en un futuro hay un error o un conflicto en el proyecto, podemos volver a una versión anterior del mismo. Además, nos permite proporcionar información adicional sobre los cambios realizados en esa versión, como: notas de la versión, cambios importantes, correcciones de errores o nuevas características.

Es una forma de marcar hitos en el desarrollo y proporcionar versiones estables y documentadas del proyecto.

Para crear el **RELEASE** clickeamos sobre **RELEASES(1)** y esto nos llevará al menú en donde se crean las **"RELEASE"**

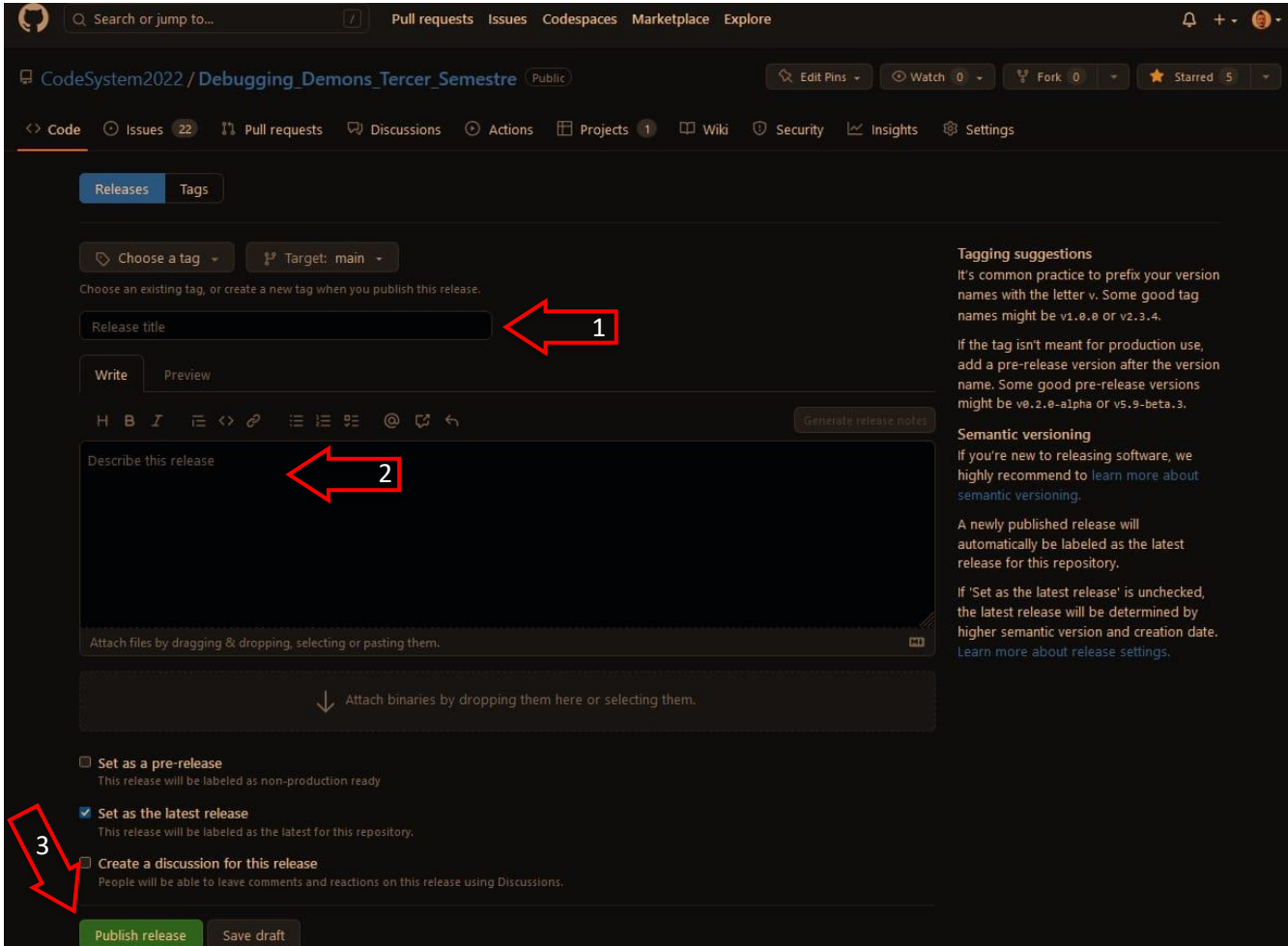


En la siguiente ventana es donde procedemos a la creación del **RELEASE**, para ello presionamos sobre el botón **DRAFT A NEW RELEASE(1)**.

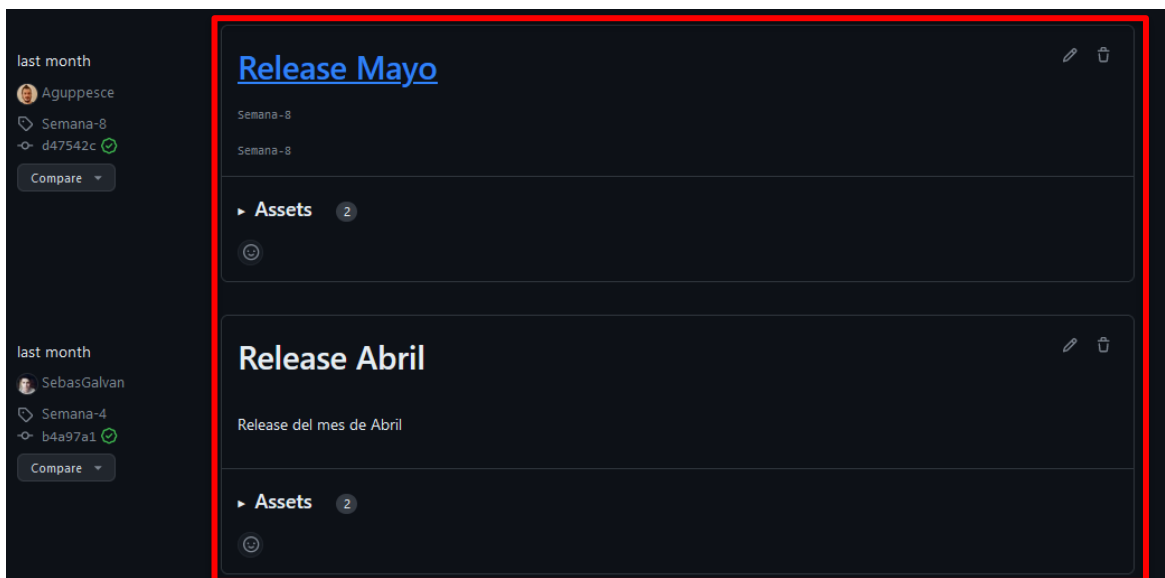




Una vez presionado el botón **DRAFT A NEW RELEASE** nos llevará a la siguiente ventana, en donde podemos asignar el nombre(1) del **RELEASE** como así también, si es necesario, hacer una descripción(2) del mismo. Una realizado esto, procedemos a crear el nuevo **RELEASE**(3).

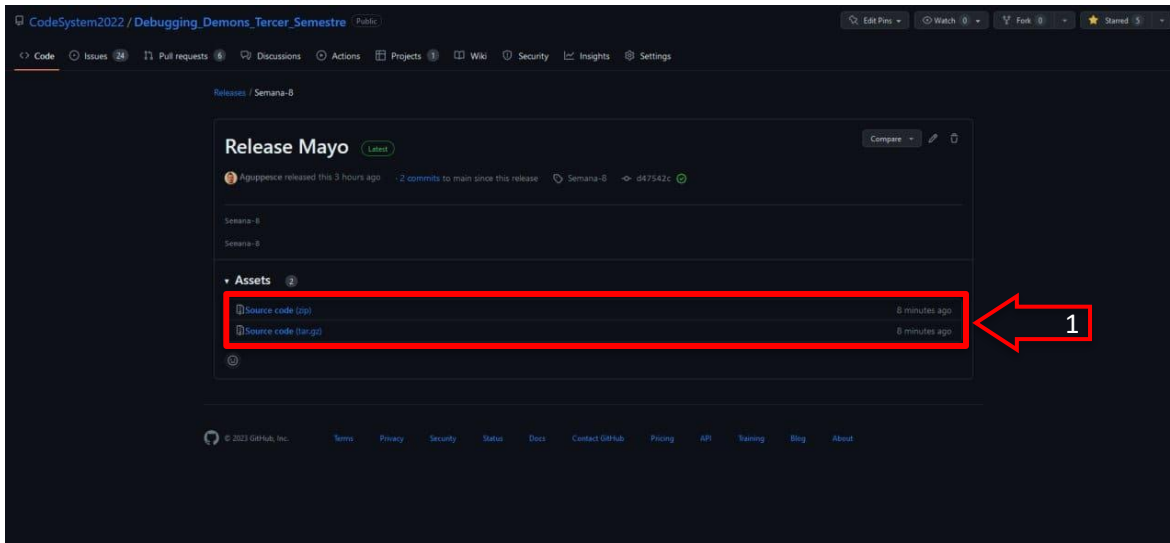


Una vez creado el **RELEASE** del mes, se mostrará en la pantalla de **RELEASES** junto con los demás **RELEASES**.



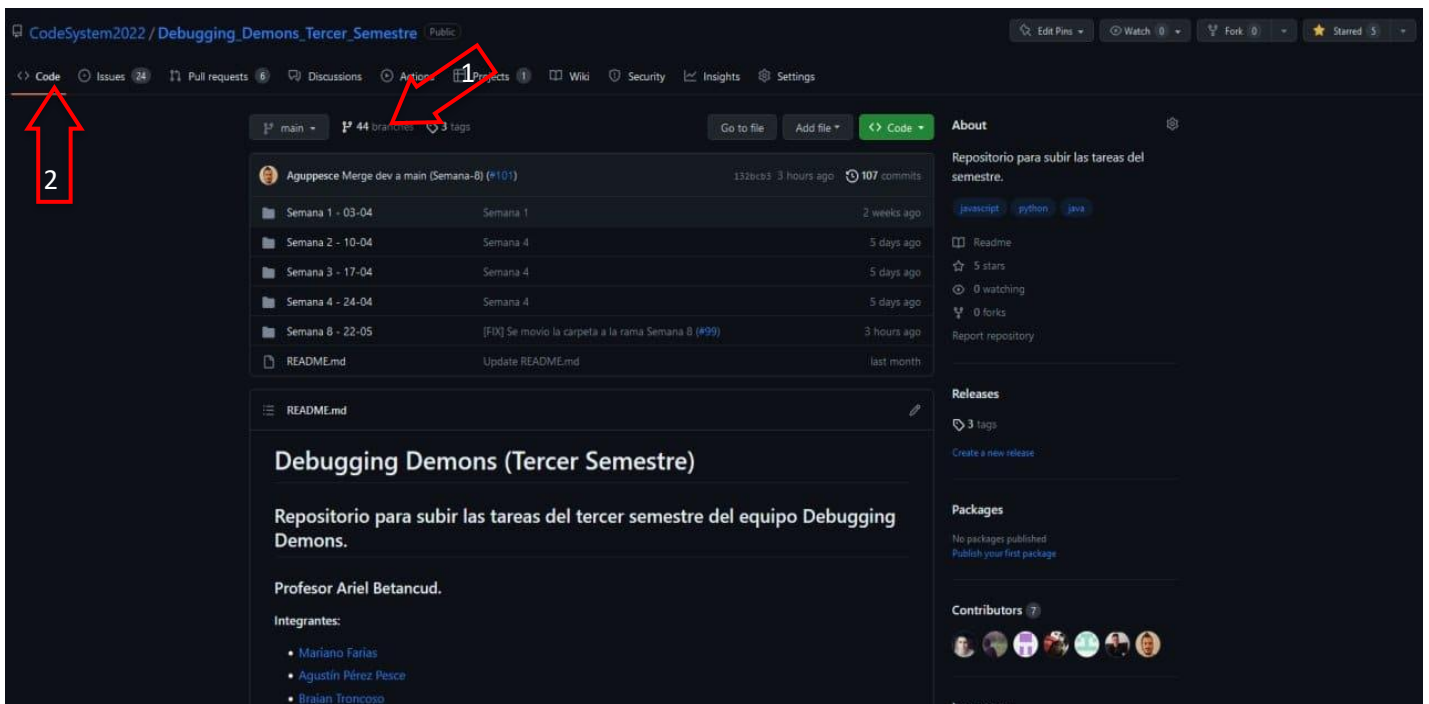


Finalmente si ingresamos al **RELEASE Mayo** podemos observar que se pueden descargar los comprimidos(1) en formato “.zip”, estos comprimidos tienen todo el código del proyecto hecho hasta la **Semana-8**.



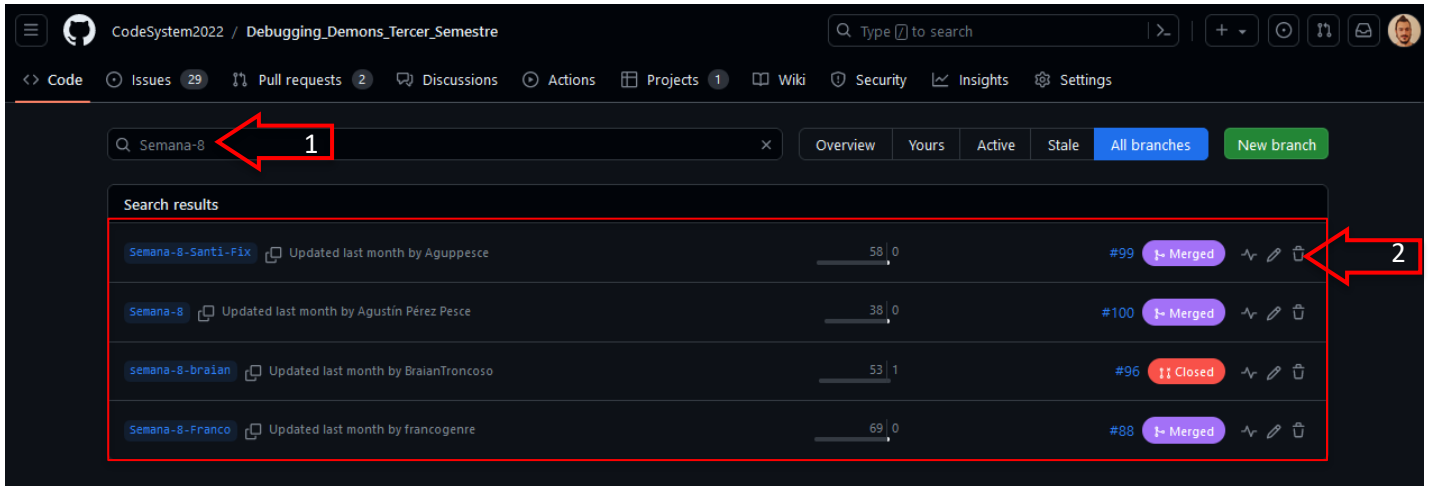
## 11) BORRAR LAS RAMAS QUE NO SE UTILIZAN, SOLO DEBEN QUEDAR LAS DE LAS SEMANAS + DEV + MAIN .

Si bien este es un proceso que se puede realizar al momento de cerrar el **PULL REQUEST**. Otra forma de hacerlo es a través del menú **BRANCHES(1)** ubicado en la pestaña **CODE(2)**.





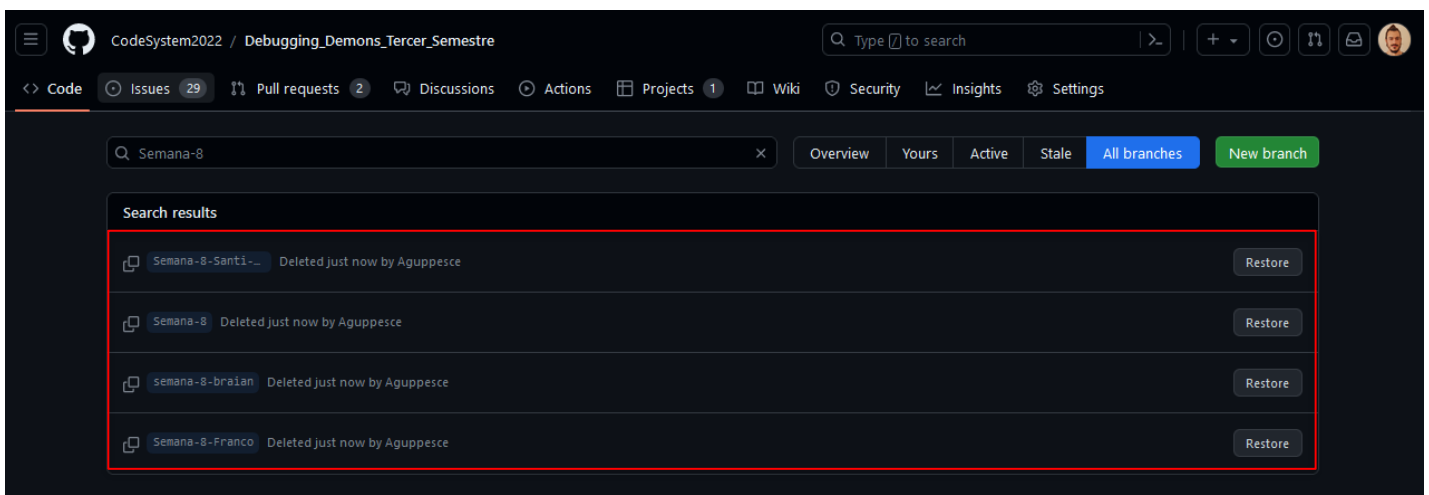
Una vez dentro del menú **BRANCHES**, filtramos las ramas por **Semana-8** y nos fijamos en el listado las ramas que tienen “**Semana-8**” en su nombre. Posteriormente clickeamos en el ícono de eliminar(2) de las ramas que vamos a borrar.



Search results

| Branch Name        | Updated by                                | Commits | Status      | Actions |
|--------------------|---|---------|-------------|---------|
| Semana-8-Santi-Fix | Updated last month by Aguppesce           | 58   0  | #99 Merged  | ✓ ✎ 🗑️  |
| Semana-8           | Updated last month by Agustín Pérez Pesce | 38   0  | #100 Merged | ✓ ✎ 🗑️  |
| semana-8-braian    | Updated last month by BraianTroncoso      | 53   1  | #96 Closed  | ✓ ✎ 🗑️  |
| Semana-8-Franco    | Updated last month by francogenre         | 69   0  | #88 Merged  | ✓ ✎ 🗑️  |

Quedando así eliminadas las ramas que ya no se utilizarán.



Search results

|                    |                               |         |
|--------------------|-------------------------------|---------|
| Semana-8-Santi-... | Deleted just now by Aguppesce | Restore |
| Semana-8           | Deleted just now by Aguppesce | Restore |
| semana-8-braian    | Deleted just now by Aguppesce | Restore |
| Semana-8-Franco    | Deleted just now by Aguppesce | Restore |