

TRABAJO SCRUM

Grupo Gigabyte

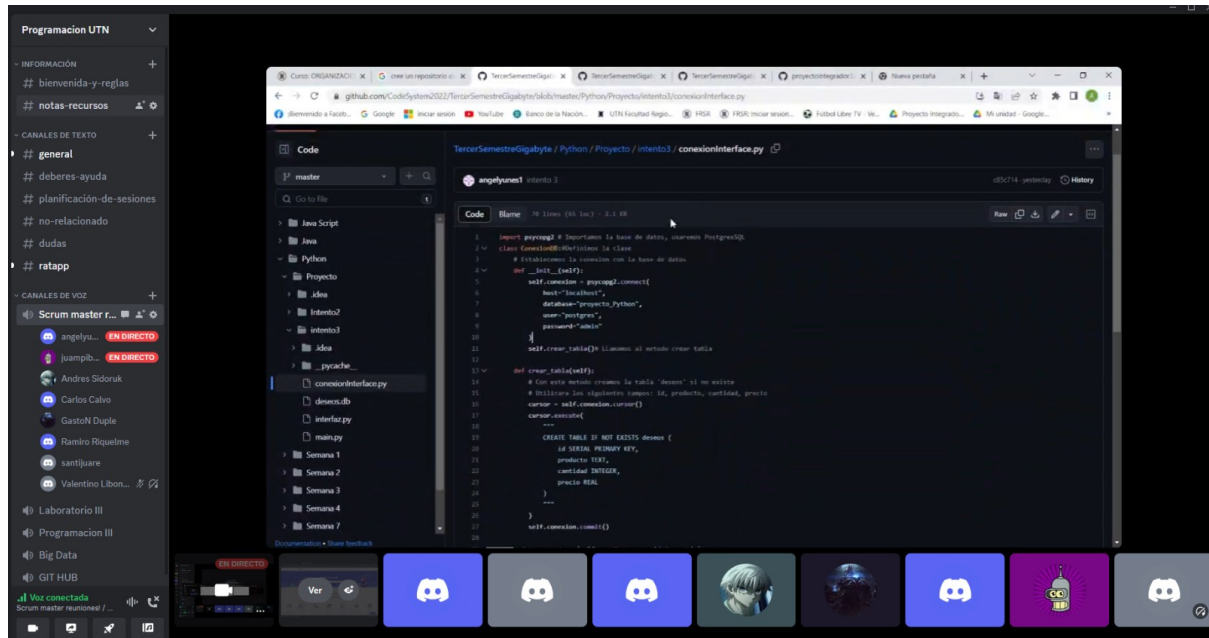
- **Santiago Agustin Juarez** - sjuarez010@gmail.com
- **Juan Pablo Barroso** - juampibarroso164@gmail.com
- **Angel Yunes** - angelyunes1@gmail.com
- **Carlos Alberto Calvo** - ccalvo.ccarlos@gmail.com
- **Ramiro Valentin Riquelme** - ramariquelme3596@gmail.com
- **Andrés Nicolás sidoruk** - 98sidoruk98@gmail.com
- **Gaston Duplessis** - gasty_d@hotmail.com
- **Valentino Libonati** - valentinogrosso1998@gmail.com

INTRODUCCIÓN

Como grupo nos organizamos y dividimos las tareas por vía WhatsApp, por ahí coordinamos fechas para ponernos de acuerdo en la disponibilidad horaria de cada integrante y organizamos una reunión vía Discord, tratando de en lo posible ser la mayor cantidad de integrantes la fecha establecida. En nuestro discord tenemos diferentes canales según la tarea que vayamos a realizar como una forma de organizarnos mejor



FORMA DE TRABAJO:



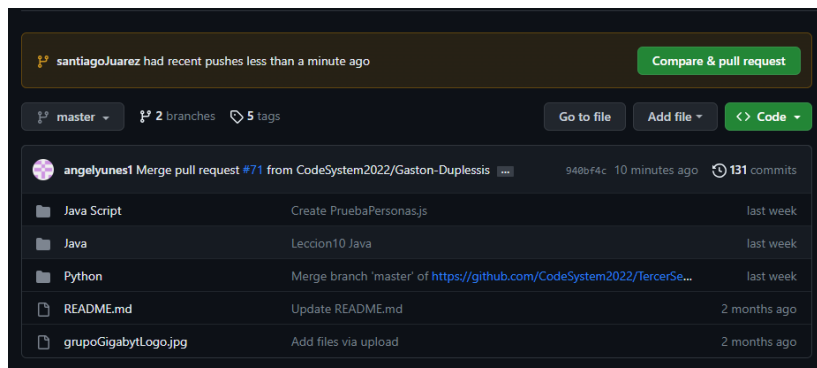
Una vez reunidos, comenzamos a planificar la semana siguiente y asignamos un Scrum Master para cada materia. Estos roles se intercambian a lo largo de las diferentes semanas, lo que significa que en una semana contamos con tres Scrum Masters, uno para cada materia. Esta dinámica nos permite trabajar de manera más organizada y garantizar que todos los miembros tengan roles equilibrados en el equipo.

El trabajo del Scrum Master en nuestro caso implica asignar tareas a los demás miembros según la materia, supervisar la correcta presentación del código y mergear los archivos subidos por los demás compañeros de grupo. De esta manera, mantenemos un control de calidad en el proceso y fomentamos la colaboración efectiva entre los integrantes.

En caso de que el Scrum Master identifique un error en el código o una forma incorrecta de subirlo al repositorio, se comunicará con el integrante correspondiente para informarle sobre el problema y brindarle ayuda para poder corregirlo y completar la tarea de manera adecuada. Si el error es desconocido o no se sabe cómo solucionarlo, el Scrum Master informará al grupo y se intentará resolverlo como equipo, fomentando la colaboración y el intercambio de conocimientos.

En situaciones en las que un integrante no pueda o no esté disponible para subir el código al repositorio debido a alguna razón, el Scrum Master asignará la tarea a otro miembro del equipo. De esta manera, se asegura que el trabajo continúe sin interrupciones y se mantiene el flujo de trabajo establecido.

REPOSITORIO GIGABYTE:



En la imagen superior se puede observar nuestro repositorio.

En el cual hemos estado trabajando en el transcurso del semestre, aportando todos los integrantes los trabajos que debemos realizar. Obteniendo un registro y copia de nuestro trabajo. Permitiéndonos acceder a él desde cualquier lugar con nuestra cuenta.

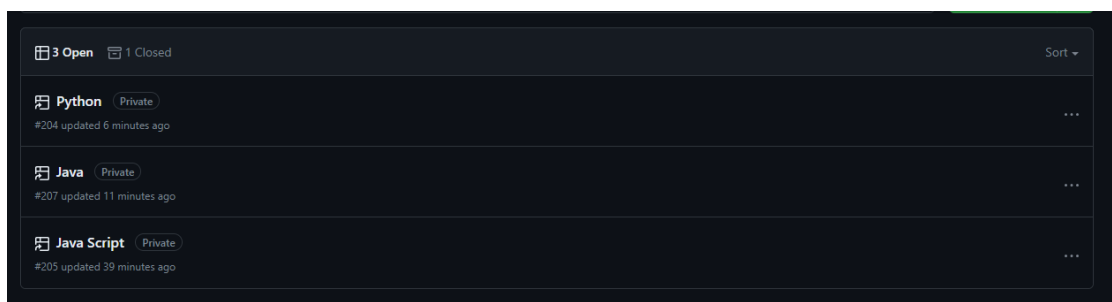
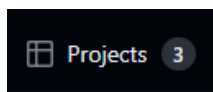
<https://github.com/CodeSystem2022/TercerSemestreGigabyte>

Un repositorio es una estructura web que permite a los programadores organizarse y almacenar proyectos de desarrollo de software. Un repositorio en GitHub contiene todos los archivos y carpetas relacionados con tu proyecto, incluyendo el código fuente, la documentación, las imágenes y cualquier otro recurso necesario.

Puedes utilizar GitHub para realizar un seguimiento de los cambios en los archivos, crear ramas para trabajar en nuevas características o correcciones de errores sin afectar la versión principal del proyecto y fusionar sus cambios con el proyecto principal cuando estén listos.

PROJECTS:

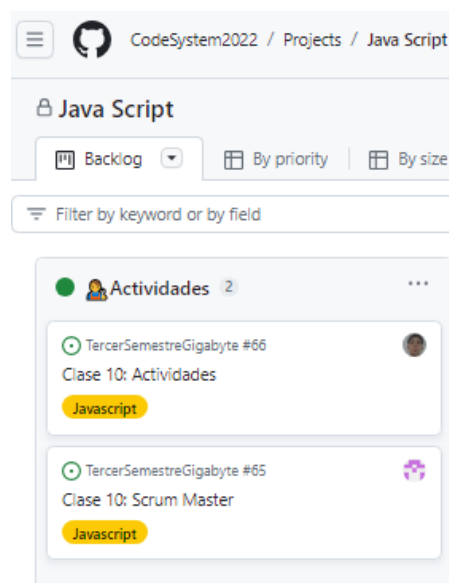
En la parte de proyectos, nosotros dividimos el trabajo por materia y dentro de cada materia sus respectivas tareas divididas por las diferentes clases, en la foto se puede apreciar los 3 proyectos creados (Python, Java y Javascript)



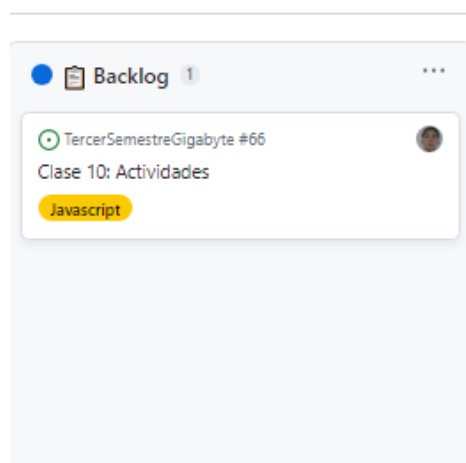
FORMA DE TRABAJO DEL SCRUM MASTER DESIGNADO DE LA SEMANA

Nosotros a lo largo de este semestre teniendo en cuenta la metodología scrum hemos cargado los códigos correspondientes a las clases. Para ello hemos ido seleccionando un scrum master (SM) por semana que se encarga de asignar las actividades de la semana a cada compañero, crear los issues y el milestone en los tableros de cada proyecto y se pacta un tiempo determinado para que estos lo lleven a cabo.

Por ejemplo en el caso de esta semana fue Valentino Libonati el Scrum Master en el tablero del proyecto de JavaScript, crea y carga las actividades de la semana, en este caso la clase se ha asignado a un compañero del grupo Santiago Juarez:



Una vez que comienza a trabajar el encargado de las actividades y mediante una comunicación entre ambos él mueve su issues de **#Actividades** a **#Backlog**, de esta manera el SM sabe que ha comenzado a trabajar.



Cuando el encargado de la actividad termina su trabajo, realiza la carga del ejercicio en donde primero se crea el Commit de esa carga de ejercicios de JavaScript y se designa el autor o co-Autores en este caso Santiago Juarez, una vez hecho el commit se realiza un push, se revisa el codigo para asi luego crear la Pull Request que será supervisada por el Scrum Master.

```
60 + }
61 +
62 +}
63 +
64 + producto1 = new Producto("TV", 2500);
65 + producto2 = new Producto("SmartPhone", 26000);
66 + console.log(producto1.toString());
67 + console.log(producto2.toString());
68 +
69 + orden1 = new Orden();
70 + orden1.agregarProducto(producto1);
71 + orden1.agregarProducto(producto2);
72 + orden1.mostrarOrden();
73 +
74 + producto3 = new Producto("PC", 50000);
75 + producto4 = new Producto("Heladera", 30000);
76 + producto5 = new Producto("Cocina", 60000);
77 + producto6 = new Producto("Cocina2", 52000);
78 + producto7 = new Producto("Cocina3", 63000);
79 + producto8 = new Producto("Cocina5", 61000);
80 + orden2 = new Orden();
81 + orden2.agregarProducto(producto3);
82 + orden2.agregarProducto(producto4);
83 + orden2.agregarProducto(producto5);
84 + orden2.agregarProducto(producto6);
85 + orden2.agregarProducto(producto7);
86 + orden2.agregarProducto(producto8);
87 + orden2.mostrarOrden();
```

Además debe encargarse de correr su issues de actividad de **#Backlog** a **#Terminando**. Sube su actividad en una nueva rama con su nombre y le da aviso al SM de que ha terminado su actividad, y este lo revisará.

Si está todo en orden, el SM hará un merge con la rama master.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base: master ← compare: santiagolopez ✓ Able to merge. These branches can be automatically merged.

🔔 Clase 10 Java Script #72

Se sube la lección de sistema de ventas, carpeta Lección08 🤖 Se reviso el código y esta correcto.

🔗 View pull request

↔ 2 commits

📄 1 file changed

👤 2 contributors

Commits on Jun 23, 2023

Clase 10 Java Script

santiagolopez committed yesterday

c4e0f6 <>

Commits on Jun 24, 2023

Merge branch 'master' into santiagolopez

valegrossol committed 23 minutes ago

d0d2ecf <>

Showing 1 changed file with 87 additions and 0 deletions.

Split Unified

JavaScript/TSLección08/SistemaVentas.js

@@ -0,0 +1,87 @@

1 + class Producto{

2 + static contadorProducto = 0;

3 + constructor(nombre,precio){

4 + this._idProducto = ++Producto.contadorProducto;

5 + this._nombre=nombre;

6 + this._precio=precio;

7 + }

🔗 Open

Clase 10 Java Script #72

valegrossol wants to merge 2 commits into master from santiagolopez

valegrossol self-assigned this 1 minute ago

👤 valegrossol commented now

Member Author ...

Se reviso el código y esta correcto.

📎

Add more commits by pushing to the santiagolopez branch on CodeSystem2022/TercerSemestreGigabyte.

🔍 Review requested

Review has been requested on this pull request. It is not required to merge. [Learn more.](#)

Show all reviewers

🔍 1 pending reviewer

Require approval from specific reviewers before merging

Branch protection rules ensure specific people approve pull requests before they're merged.

Add rule ×

🔍 Continuous integration has not been set up

GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

🔍 This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

You can also open this in GitHub Desktop or view command line instructions.

👤 2 participants

🔒 Lock conversation

Y luego se moverán los issues a **#Finalizado**, donde los cerrará:

🟢 Finalizado 16

🔍 TercerSemestreGigabyte #66

Clase 10: Actividades

JavaScript

🔍 TercerSemestreGigabyte #65 ***

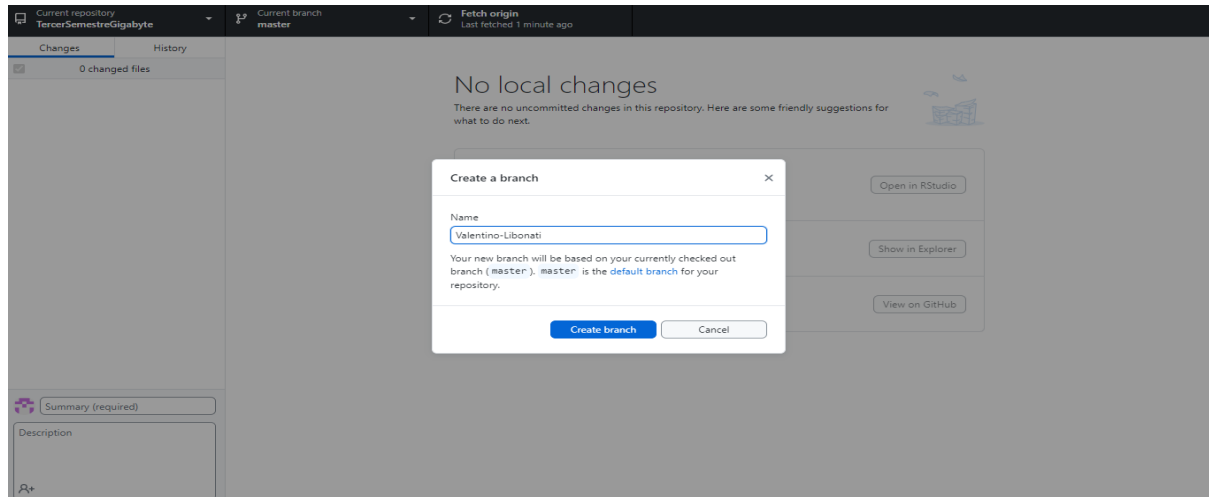
Clase 10: Scrum Master

JavaScript

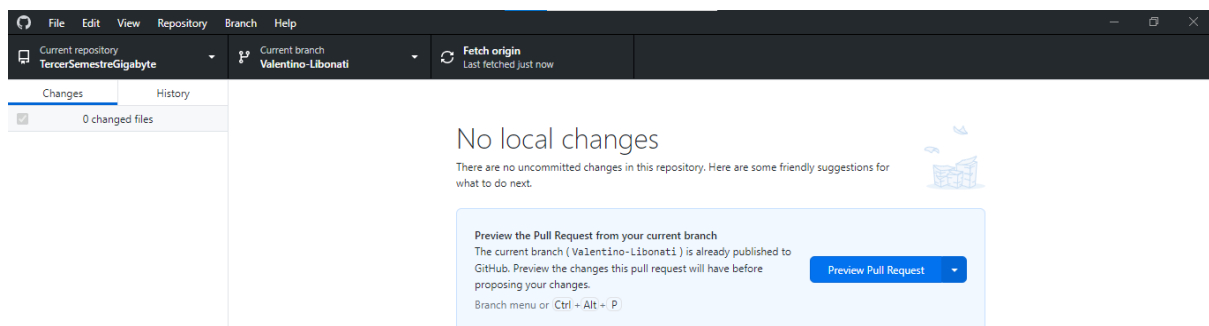
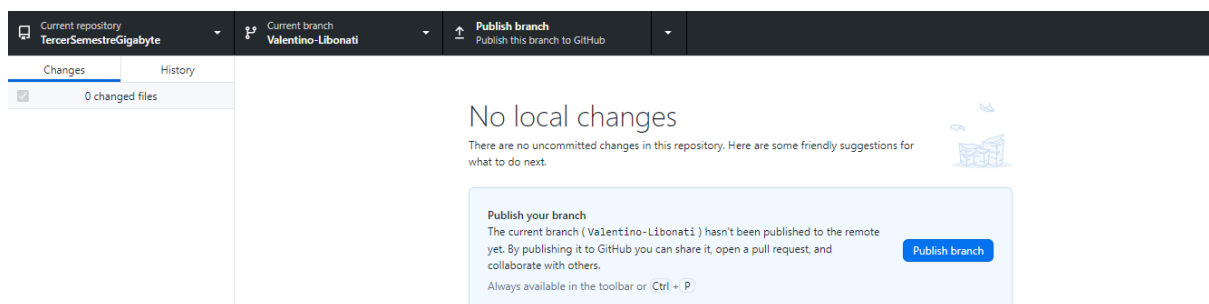
FORMA DE TRABAJO DEL ENCARGADO DE LA TAREA

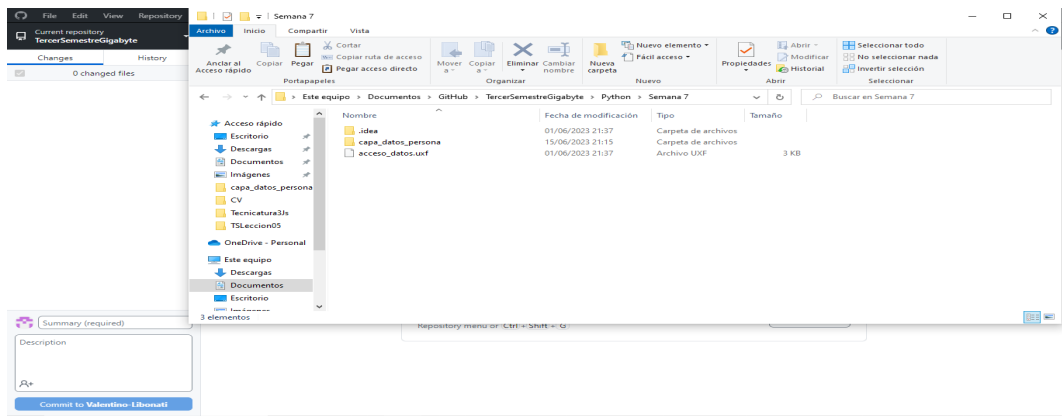
GITHUB DESKOPT:

El encargado de la semana una vez que tiene su código terminado procede a subirlo al repositorio:

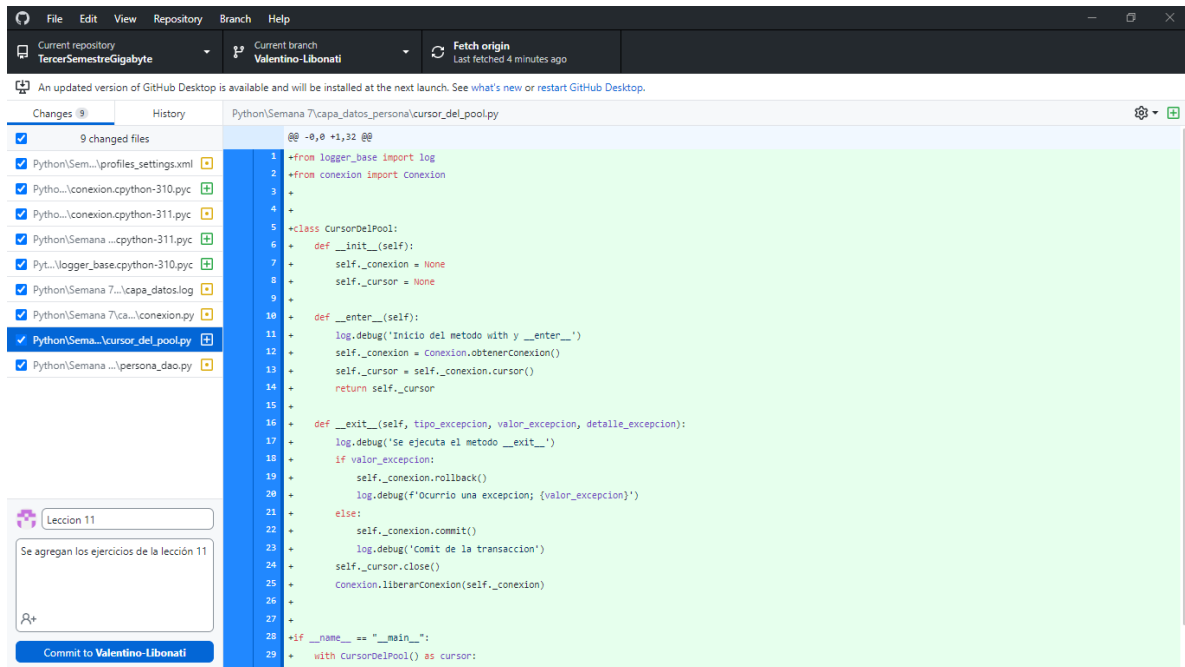


Utilizando GitHub Desktop, el encargado de realizar la actividad, crea una rama con su nombre, desde la cual realizará la carga de su trabajo:

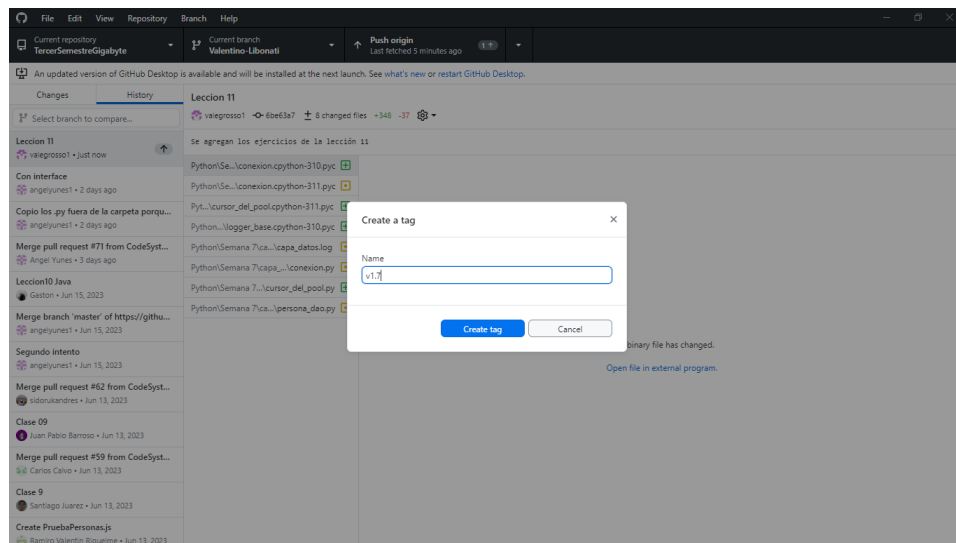




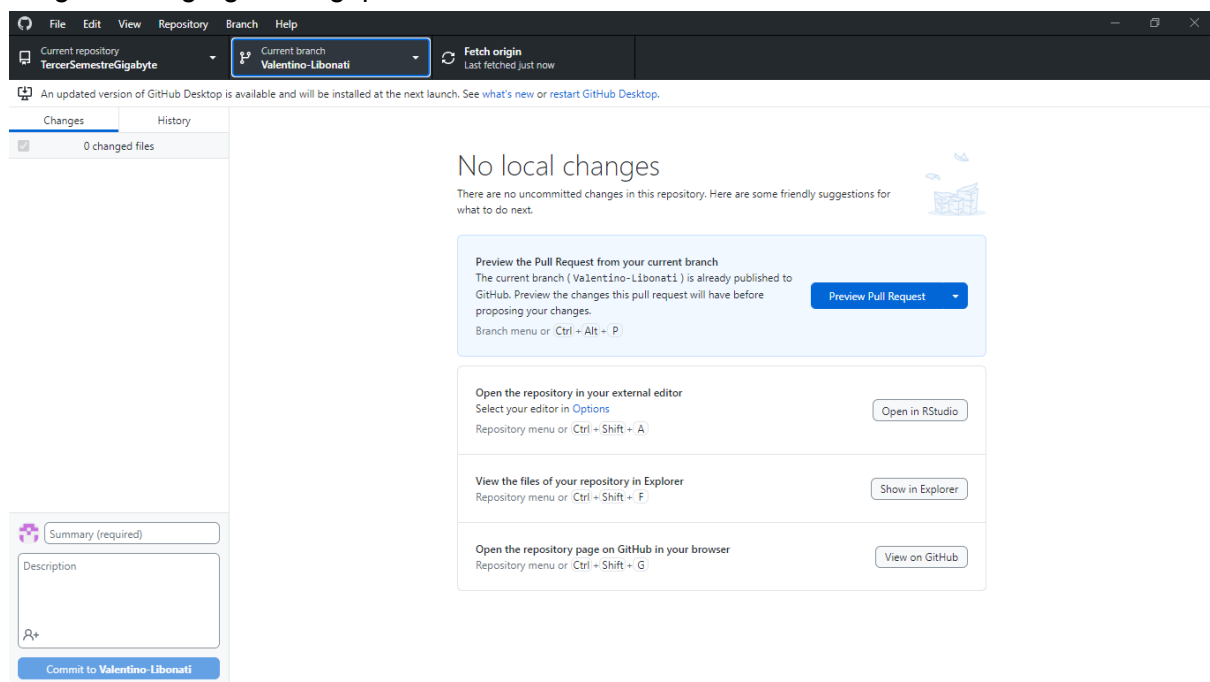
Una vez creada la rama, el encargado de la actividad, crea un proyecto o en este caso continúa con el proyecto ya iniciado. Donde agregara el código de la clase asignada



Una vez terminado su actividad (como se muestra en GitHub, donde podemos apreciar varios cambios en el código con respecto a la versión anterior), se agrega un commit y una descripción a estos cambios



Luego se le agrega un tag, para identificar las diferentes versiones.



Una vez realizado los pasos anteriores se hace un push(es decir, sincroniza los cambios con la nube de git), y queda listo para que el scrum master haga un merge de las ramas si es que todo el proceso y código han sido subidos de forma correcta. Luego de esto se eliminan la rama, quedando únicamente la master.