

Trabajo Scrum: Grupo OldSchool

Integrantes:

Reynoso Yessica

Reynoso Andrea

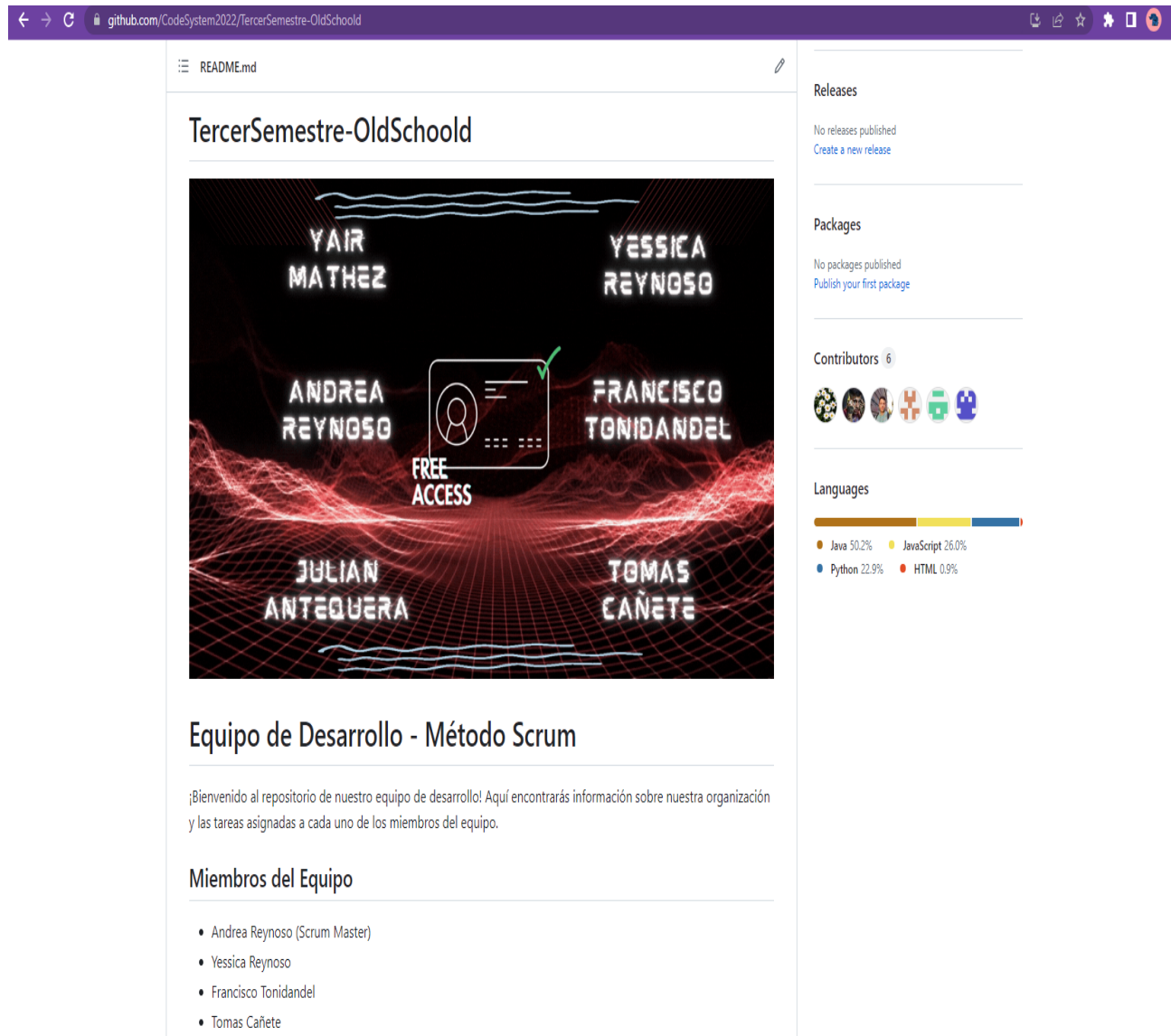
Tonidandel Francisco

Cañete Tomas

Mathez Yair

Antequera Julian

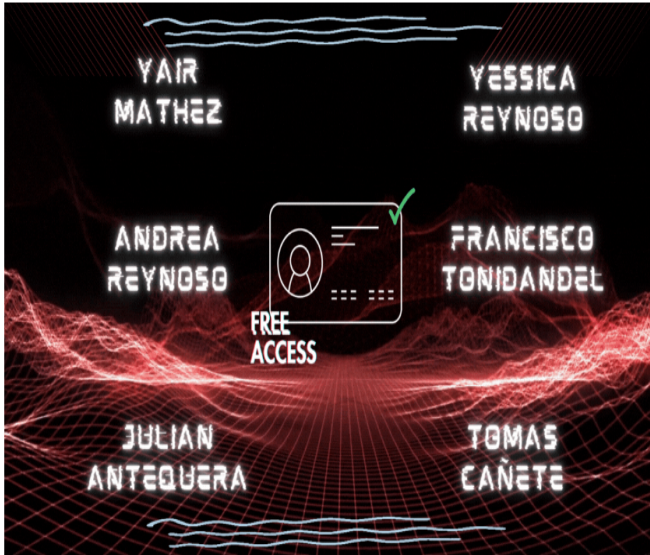
Comenzamos, creando el repositorio **TercerSemestre-OldSchool**, y nuestro primer Readme. En el cual, podemos ver los miembros, las tareas y roles asignados. Como equipo decidimos mantener a nivel de semestre un Scrum Master, por razones organizativas. A su vez, decidimos dividir las tareas por lenguaje de programación, es decir cada miembro debe de subir por semana el lenguaje de programación asignado.



← → C github.com/CodeSystem2022/TercerSemestre-OldSchool

README.md

TercerSemestre-OldSchool



Equipo de Desarrollo - Método Scrum

¡Bienvenido al repositorio de nuestro equipo de desarrollo! Aquí encontrarás información sobre nuestra organización y las tareas asignadas a cada uno de los miembros del equipo.

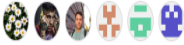
Miembros del Equipo

- Andrea Reynoso (Scrum Master)
- Yessica Reynoso
- Francisco Tonidandel
- Tomas Cañete

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

Contributors 6



Languages

Language	Percentage
Java	50.2%
JavaScript	26.0%
Python	22.9%
HTML	0.9%

README.md

Equipo de Desarrollo - Método Scrum

¡Bienvenido al repositorio de nuestro equipo de desarrollo! Aquí encontrarás información sobre nuestra organización y las tareas asignadas a cada uno de los miembros del equipo.

Miembros del Equipo

- Andrea Reynoso (Scrum Master)
- Yessica Reynoso
- Francisco Tonidandel
- Tomas Cañete
- Yair Mathez
- Julian Antequera

Roles y Tareas

Andrea Reynoso

- Rol: Scrum Master
- Tareas:
 - Coordinar las reuniones diarias del equipo.
 - Facilitar el proceso de desarrollo siguiendo el método Scrum.
 - Ayudar a resolver bloqueos y obstáculos que puedan surgir durante el proyecto.

Yessica Reynoso

- Rol: Desarrolladora
- Tareas:
 - Contribuir al desarrollo del proyecto.
 - Subir actividades relacionadas con Python.

Francisco Tonidandel

- Subir actividades relacionadas con Python.

Francisco Tonidandel

- Rol: Desarrollador
- Tareas:
 - Contribuir al desarrollo del proyecto.
 - Subir actividades relacionadas con JavaScript.

Tomas Cañete

- Rol: Desarrollador
- Tareas:
 - Contribuir al desarrollo del proyecto.
 - Subir actividades relacionadas con Java.

Yair Mathez

- Rol: Desarrollador
- Tareas:
 - Contribuir al desarrollo del proyecto.
 - Subir actividades relacionadas con JavaScript.

Julian Antequera

- Rol: Desarrollador
- Tareas:
 - Contribuir al desarrollo del proyecto.
 - Subir actividades relacionadas con Java.

¡Gracias por visitar nuestro repositorio!

Organización

Semana	Tareas	Responsables
Semana 1	- Redme	Yair Mathez

github.com/CodeSystem2022/tercerSemestre-OldSchool

README.md

- Subir actividades relacionadas con Python.

Francisco Tonidandel

- Rol: Desarrollador
- Tareas:
 - Contribuir al desarrollo del proyecto.
 - Subir actividades relacionadas con JavaScript.

Tomas Cañete

- Rol: Desarrollador
- Tareas:
 - Contribuir al desarrollo del proyecto.
 - Subir actividades relacionadas con Java.

Yair Mathez

- Rol: Desarrollador
- Tareas:
 - Contribuir al desarrollo del proyecto.
 - Subir actividades relacionadas con JavaScript.

Julian Antequera

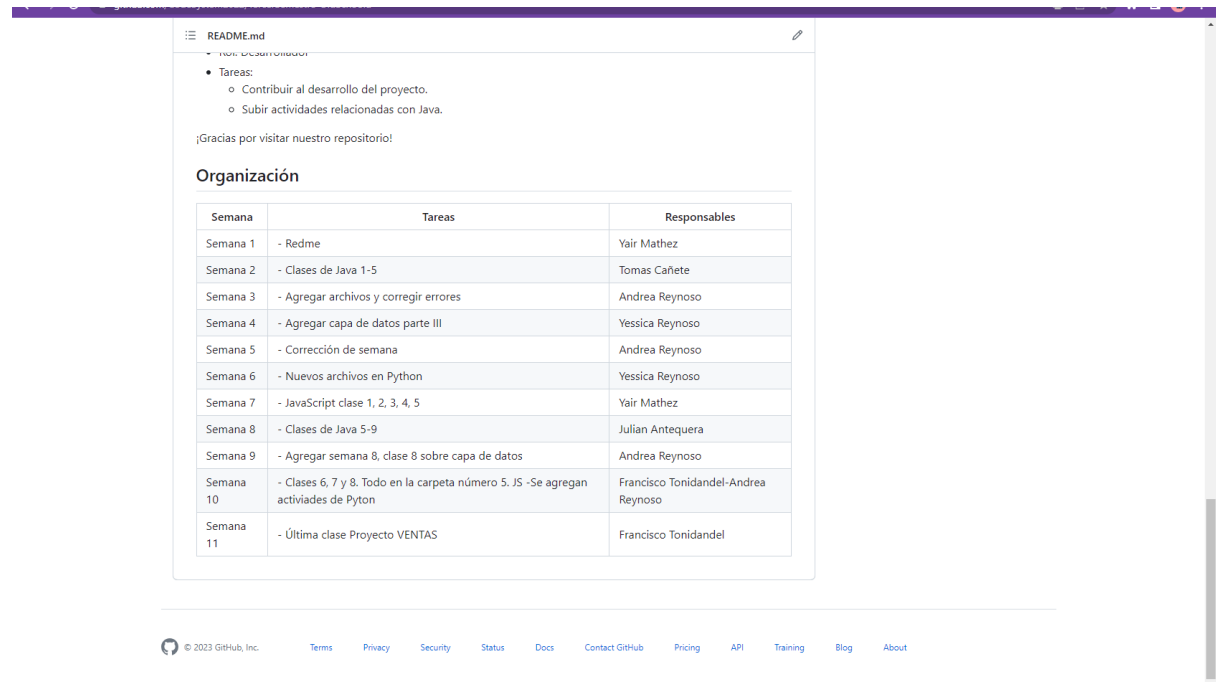
- Rol: Desarrollador
- Tareas:
 - Contribuir al desarrollo del proyecto.
 - Subir actividades relacionadas con Java.

¡Gracias por visitar nuestro repositorio!

Organización

Semana	Tareas	Responsables
Semana 1	- Redme	Yair Mathez

En el siguiente cuadro, organizamos las cargas de trabajo. Con los responsables de cada semana. La idea era ir completando semana tras semana con las tareas asignadas según los temas vistos en cada clase.



The screenshot shows a GitHub repository's README file. It contains a section titled 'Organización' with a table detailing the project schedule. The table has three columns: 'Semana' (Week), 'Tareas' (Tasks), and 'Responsables' (Responsible). The tasks are distributed across 11 weeks, with some weeks having multiple tasks or specific file additions. The responsible parties are listed for each week.

Semana	Tareas	Responsables
Semana 1	- Redme	Yair Mathez
Semana 2	- Clases de Java 1-5	Tomas Cañete
Semana 3	- Agregar archivos y corregir errores	Andrea Reynoso
Semana 4	- Agregar capa de datos parte III	Yessica Reynoso
Semana 5	- Corrección de semana	Andrea Reynoso
Semana 6	- Nuevos archivos en Python	Yessica Reynoso
Semana 7	- JavaScript clase 1, 2, 3, 4, 5	Yair Mathez
Semana 8	- Clases de Java 5-9	Julian Antequera
Semana 9	- Agregar semana 8, clase 8 sobre capa de datos	Andrea Reynoso
Semana 10	- Clases 6, 7 y 8. Todo en la carpeta número 5. JS -Se agregan actividades de Python	Francisco Tonidandel-Andrea Reynoso
Semana 11	- Última clase Proyecto VENTAS	Francisco Tonidandel

SEMANA 6:

- 1) **Objetivo de la semana 6:** Durante la semana 6, el equipo se enfocó en tres tareas principales: "Clase 06.JS", "Clase06 JAVA" y una tarea de Python, "Semana_6". El objetivo es desarrollar y completar las tareas propuestas por cada clase.

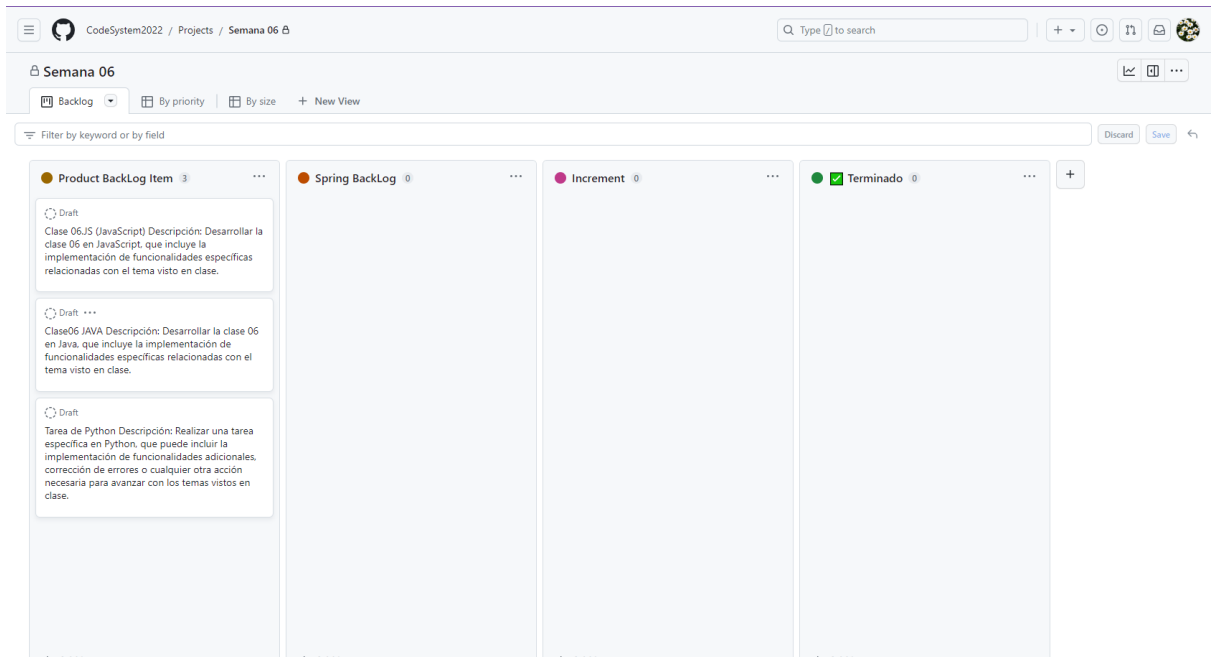
Responsables:

"Clase 06.JS": Tonidandel Francisco.

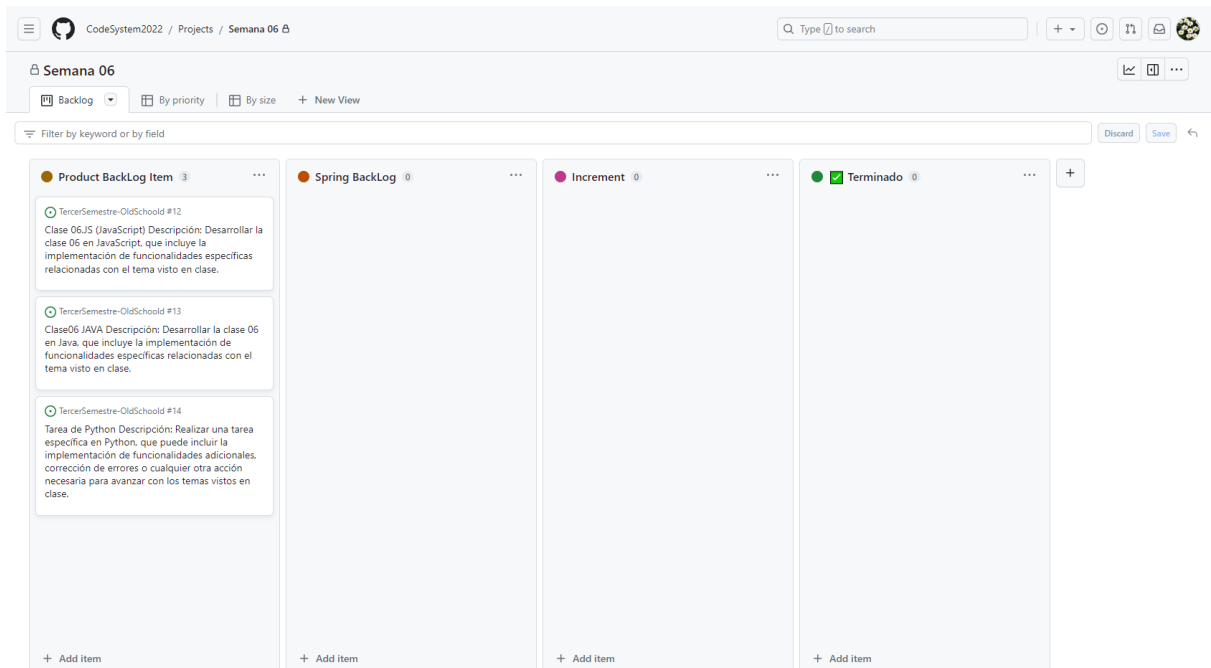
"Clase06 JAVA": Julian Antequera.

"Semana_0": Andrea Reynoso-Yessica Reynoso

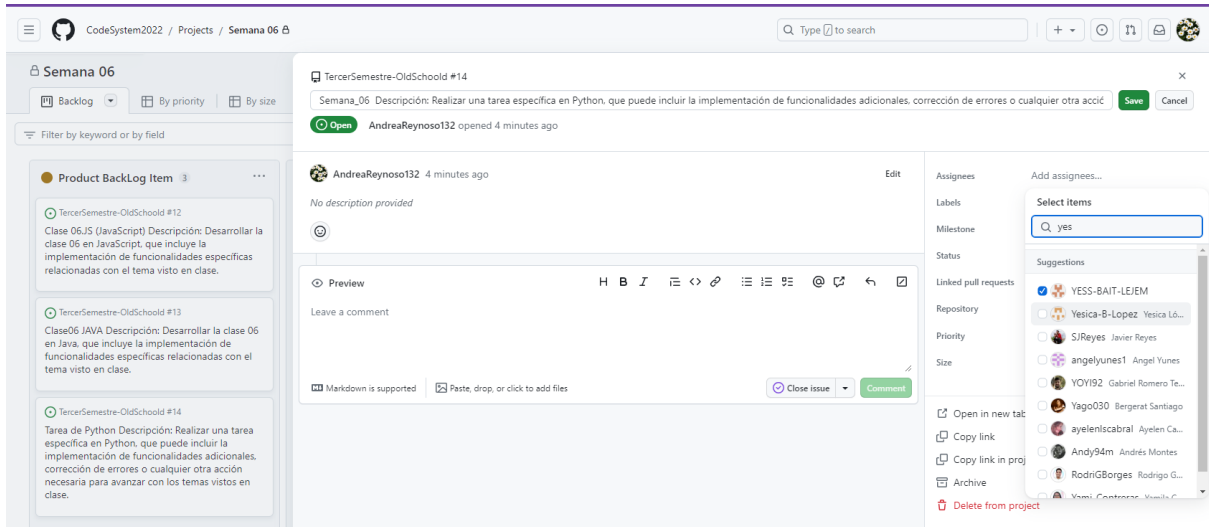
Comenzamos a crear los Product BackLog Items de la semana 6:



Las pasamos a Issues:

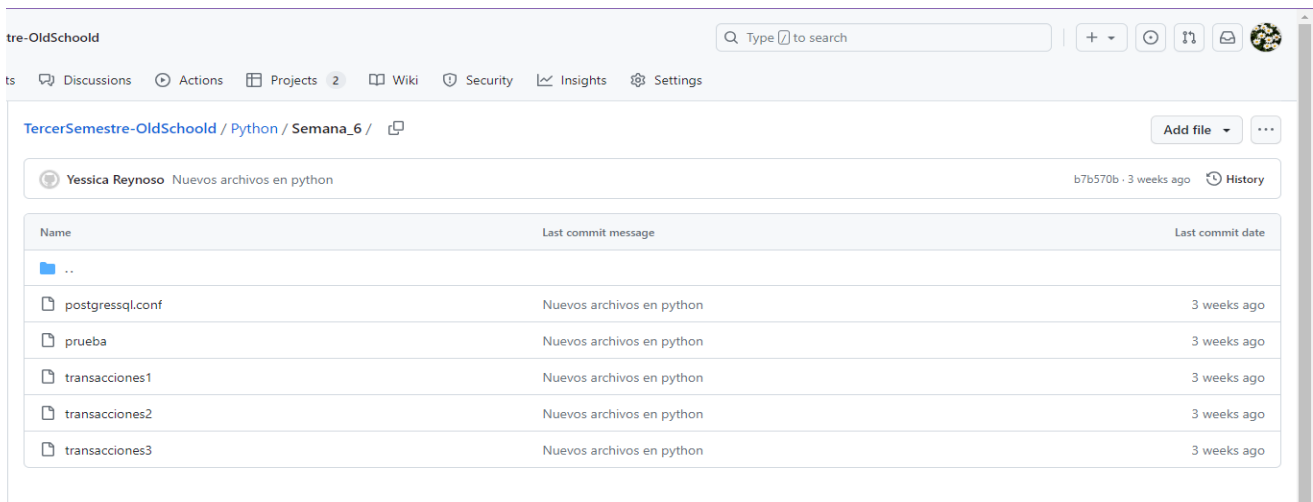


Asignamos las tareas:



Se comienzan a subir las tareas al repositorio:

Python DataBase:



Pasamos la tarjeta a TERMINADO:

CodeSystem2022 / Projects / Semana 06

Q Type to search

+ - 🔍 📄 🌐

Semana 06

Backlog By priority By size + New View

Filter by keyword or by field Discard Save ↩

Product BackLog Item 2

TerceSemestre-OldSchool #12

Clase 06 JS (JavaScript) Descripción: Desarrollar la clase 06 en JavaScript, que incluye la implementación de funcionalidades específicas relacionadas con el tema visto en clase.

TerceSemestre-OldSchool #13

Clase06 JAVA Descripción: Desarrollar la clase 06 en Java, que incluye la implementación de funcionalidades específicas relacionadas con el tema visto en clase.

Spring BackLog 0

Increment 0

Terminado 1

TerceSemestre-OldSchool #14

Semana_06 Descripción: Realizar una tarea específica en Python, que puede incluir la implementación de funcionalidades adicionales, corrección de errores o cualquier otra acción necesaria para avanzar con los temas vistos en clase.

+

Así sucesivamente con las demás tareas:

JAVA INTERFACES:

stre-OldSchool

Q Type to search + - 🔍 📄 🌐

its Discussions Actions Projects 2 Wiki Security Insights Settings

TerceSemestre-OldSchool / Java / Leccion06Interfaces /

Add file ...

Juliantequera Clases java 5-9

16d3389 - yesterday History

Name	Last commit message	Last commit date
..		
out/production/Clase6Interfaces	Clases java 5-9	yesterday
src	Clases java 5-9	yesterday
Clase6Interfaces.iml	Clases java 5-9	yesterday

CodeSystem2022 / Projects / Semana 06

Search: Type to search

Semana 06

Backlog By priority By size + New View

Filter by keyword or by field Discard Save

Product BackLog Item 1

TerceSemestre-OldSchool #12

Clase 06 JS (JavaScript) Descripción: Desarrollar la clase 06 en JavaScript, que incluye la implementación de funcionalidades específicas relacionadas con el tema visto en clase.

Spring BackLog 0

Increment 0

Terminado 2

TerceSemestre-OldSchool #14

Semana_06 Descripción: Realizar una tarea específica en Python, que puede incluir la implementación de funcionalidades adicionales, corrección de errores o cualquier otra acción necesaria para avanzar con los temas vistos en clase.

TerceSemestre-OldSchool #13

Clase06 JAVA Descripción: Desarrollar la clase 06 en Java, que incluye la implementación de funcionalidades específicas relacionadas con el tema visto en clase.

JAVASCRIPT

panchotoni committed 3 weeks ago 1 parent 78b4568 commit a6c2b2a

Showing 2 changed files with 144 additions and 0 deletions. Split Unified

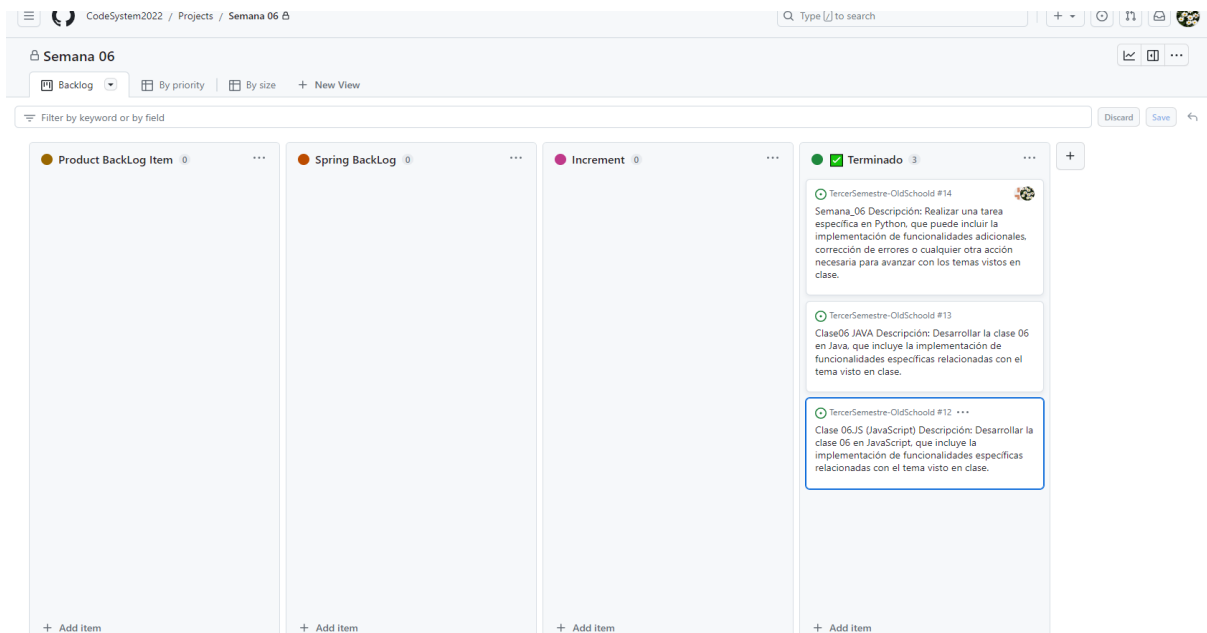
Filter changed files

- JavaScript/Leccion05Clases
 - 05-03-clases.js
 - index.html

```

132 JavaScript/Leccion05Clases/05-03-clases.js
...
@@ -0,0 +1,132 @@
1 + class Persona{ //Clase padre
2 +
3 +   static contadorPersonas = 0; //Atributo estatico
4 +   //email = 'Valor default email'; //Atributo no estatico
5 +
6 +   static get MAX_OBJ(){ //Este metodo simula una constante
7 +     return 5
8 +   }
9 +
10 +   constructor(nombre, apellido){
11 +     this._nombre = nombre;
12 +     this._apellido = apellido;
13 +     if(Persona.contadorPersonas < Persona.MAX_OBJ){
14 +       this.idPersona = ++Persona.contadorPersonas;
15 +     } else {
16 +       console.log("Se ha superado el maximo de Objetos permitidos");
17 +     }
18 +     //console.log("Se incremento el contador: " +Persona.contadorObjetosPersona);
19 +   }
20 +
21 +   get nombre() {
22 +     return this._nombre;
23 +   }
24 +
25 +   set nombre(nombre){
26 +     this._nombre = nombre;
27 +   }
28 +
29 +   get apellido() {
30 +     return this._apellido;
31 +   }
32 +
33 +

```



Explicación de la metodología de trabajo:

Durante la semana 6 (y todo el semestre), nos organizamos siguiendo el método Scrum para llevar a cabo nuestras tareas. Como equipo, nos reunimos al comienzo de la semana para revisar el Product Backlog y determinar las tareas que se llevarían a cabo durante esa semana, según lo visto en cada clase.

Utilizamos GitBash como herramienta para gestionar nuestro repositorio de código. Como parte de nuestra metodología, acordamos que todos los integrantes subiríamos nuestros avances a la rama principal (main) del repositorio. Sin embargo, antes de subir los cambios de manera remota, nos aseguramos de revisar y validar el código que estaba siendo subido por otros miembros del equipo.

Por ejemplo, si un compañero subía un ejercicio en Java, otro miembro del equipo se encargaba de revisar el código y ejecutarlo localmente para asegurarse de que funcionaba correctamente. De esta manera, nos aseguramos de que todos los ejercicios estuvieran cumpliendo con los requisitos establecidos.

Si bien utilizamos la rama principal para gestionar nuestras actualizaciones, reconocemos la importancia de utilizar ramas individuales en un proyecto real. Próximamente, planeamos implementar el uso de

ramas individuales para que cada miembro del equipo pueda trabajar en su propia rama y luego fusionar los cambios en la rama principal una vez que se hayan validado y probado correctamente.

Entendemos que el uso de ramas individuales es una buena práctica en proyectos más grandes y complejos, ya que permite un mayor control sobre los cambios y facilita la colaboración simultánea en diferentes partes del proyecto sin interferir con el trabajo de los demás.