

# SEGUNDO SPRINT, OBJETOS Y TXT

Eduardo Gutierrez, Emiliano Luna, Alejandro  
Nava, Gabriel Alvarez

## Organización (Cómo nos organizamos)

Similar a sprint anterior

Turnandonos código, usando trello

Instrucciones, las seguimos y aportamos ideas

USAMOS WHATSAPP



Desarrollo del código (aquí se pega código actualizado)  
en la lista Doing

Miembros Etiquetas

AD + Segundo Spint Programación CODIGO +

Notificaciones Vencimiento

Seguindo 23 de abr. a las 23:59 Cumplida

Descripción

PEGAR EN COMENTARIOS CODIGO NUEVO

Continuar funciones, centrándonos en el almacenamiento de datos haciendo uso de un txt y haciendo todo en torno a objetos.

Adjuntos

py PROYECTO\_ASESORIAS.py  
Añadido: hace 12 minutos • Comentario • Descargar • Eliminar • Editar

py Sprint 2, COMENTADO.py  
Añadido: 17 de abr. a las 21:07 • Comentario • Descargar • Eliminar • Editar

py Sprint 2, HOY LO TRABAJO.py  
Añadido: 17 de abr. a las 7:33 • Comentario • Descargar • Eliminar • Editar

py Sprint.py  
Añadido: 1 de abr. a las 12:13 • Comentario • Descargar • Eliminar • Editar

Done

Segundo Spint Programación

Empezar a usar txt (YA)

Segundo Spint Definicion

Crear tablas de como se van a guardar datos

Segundo Spint Programación CODIGO

Refactorizar todo el codigo a orientada a objetos

Segundo Spint Programación CODIGO

Impresion de las asesorias de los alumnos

Segundo Spint Programación CODIGO

Impresion de la lista de los alumnos del profesor

Segundo Spint Definicion

Diagrama de la base de datos sql (ya se hicieron las tablas)

TRELLO

Completamos las tareas, pero perdimos terreno por el remake

# Reformatado

Código convertido a objetos

Clase principal (usuarios)

Clases secundarias (Estudiantes y Profesores, futuramente administradores)

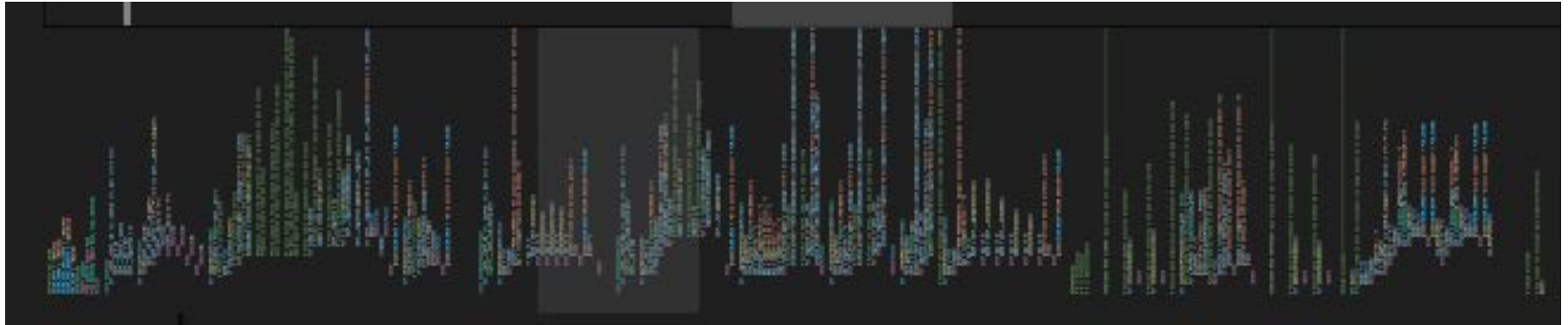
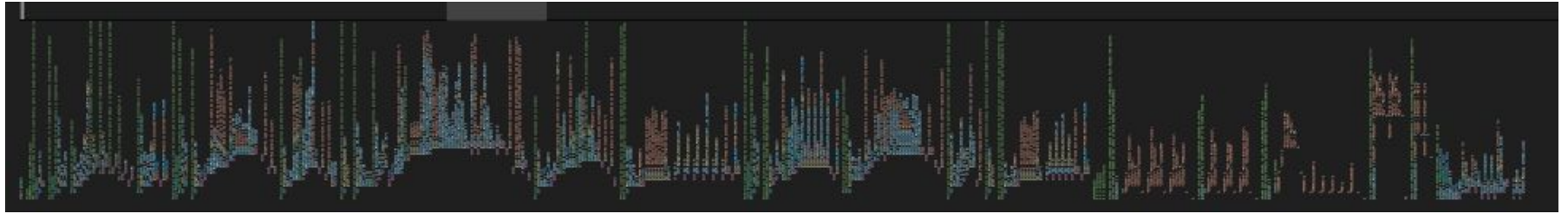
```
class usuarios:
    def __init__(self, id: str, contra: str, dir: str) -> None:
        self.__id = id
        self.__contra = contra
        self.__dir = dir
        self.__registros = []
```

```
class estudiante(usuarios):
    def __init__(self, id: str, contra: str, dir: str) -> None:
        super().__init__(id, contra, dir)
```

```
class profesor(usuarios):
    def __init__(self, idp: str, contra: str, dir: str) -> None:
        super().__init__(idp, contra, dir)
        self.__id = idp
```

# REMAKE DEL CÓDIGO

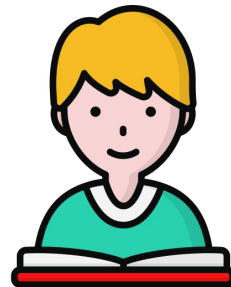
No estaba tan bonito, así que se borraron funciones/se están volviendo a hacer (viejo ➞ nuevo)



# TXT

Ya empezamos a utilizar txt's (falta crear el que guarda los comentarios)

```
ID|Nombre|Carrera|Semestre|Contrasena
0261445E|Juan Manuel Gonzalez|Ciber|6|password123
0172114E|Rafael Castellanos Frisbie|Comunicacion|2|juanpepe2004
0236784E|Paulina Mendez Flores|Finanzas|5|9900almandez
```



```
ID|Nombre|Contrasena
0255612P|Federica Mirales Morales|contrareal90
0255613P|Pepe Pepe Zumiga|12345678910
0255614P|Mariana Rodriguez Ramirez|aminomegustaelpollo
```



```
ID|Materia|Hora de inicio|Hora de Fin|Id Profesor|Fecha|Cupo
23F|Mates|2|5|0255613P|25/04/2024|12
23A|Mates|2|5|0255613P|22/04/2024|12
```



## Mejoras/Problemas

Vacaciones (se rompió el esquema)

Exámenes

Conocimiento que se adquirió a medio sprint  
(txts, más profundidad en objetos

Falta funciones de mandar mensajes y crear  
objetos al momento de leer txt





**GRACIAS POR SU  
ATENCIÓN,  
PREGUNTAS?**