

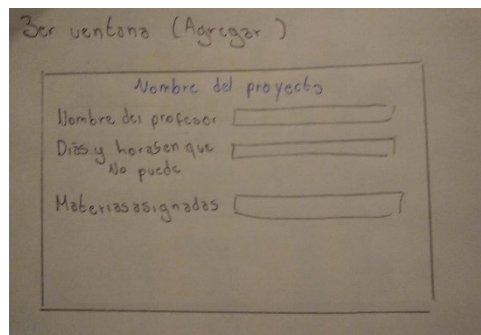
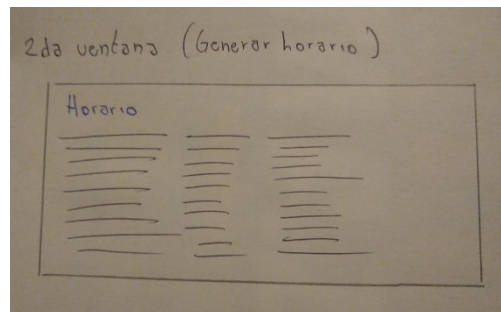
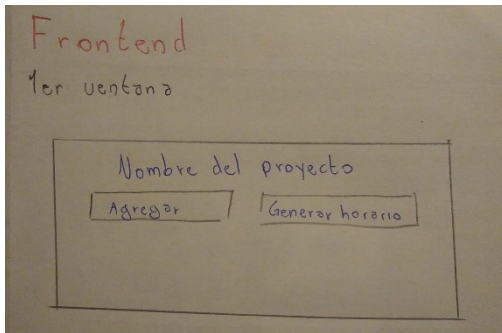
PAULINA AMEZCUA

PROCESO

Lo que hice:

En este proyecto he realizado las cosas básicas primeramente del frontend para darle una mano, por el mismo hecho de que al principio no tenía un rol específico me coloqué en uno generalista ayudando a algunos de los integrantes con partes que se les dificultasen ya sea por falta de tiempo o problemas para resolverlo.

Entonces realicé los bocetos de cómo se verían terminadas las ventanas de la interfaz, primero planteando la idea como borrador en una hoja de papel, quedando con el equipo de acuerdo con que todos estuviéramos satisfechos con la idea principal de la vista y después haciéndolo en un programa ya de una manera más visualmente agradable y limpia.



Después de que me lo confirmaran, realicé los mismos bocetos, pero en un programa para que se vieran más bonitos y entendibles.

PRIMER VENTANA:

ASIGNADOR DE HORARIOS

AGREGAR PROFESOR
GENERAR HORARIO

¡¡Esta es una herramienta para directores y administrativos, te ayudaremos a asignarle horarios a tus profesores!!

TERCERA VENTANA (AGREGAR PROFESOR):

ASIGNADOR DE HORARIOS

AREA PARA AGREGAR A TUS PROFESORES

NOMBRE DEL PROFESOR:
HORAS QUE NO TIENE DISPONIBLES:
MATERIA(S) ASIGNADA(S):

GUARDAR

SEGUNDA VENTANA (GENERAR HORARIO):

MATERIA	PROFESOR	DIA	HORARIO	AULA
Análisis de algoritmos	Jorge Ernesto López	Lunes	09:00 am - 10:55 am	1
Análisis de algoritmos	Jorge Ernesto López	Lunes	11:00 am - 12:55 pm	2
Análisis de algoritmos	Jorge Ernesto López	Martes	09:00 am - 10:55 am	1
Análisis de algoritmos	Lorena Sánchez	Lunes	09:00 am - 10:55 am	3
Lógica matemática	José Luis Gutiérrez	Lunes	09:00 am - 10:55 am	4
Lógica matemática	José Luis Gutiérrez	Lunes	11:00 am - 12:55 pm	5
Lógica matemática	María Elena Suarez	Martes	09:00 am - 10:55 am	2
Lógica matemática	Lorena Sánchez	Martes	09:00 am - 10:55 am	3
Lógica matemática	Lorena Sánchez	Miércoles	09:00 am - 10:55 am	1
Mecánica	Jorge Ernesto López	Lunes	03:00 pm - 04:55 pm	6
Mecánica	Jorge Ernesto López	Martes	11:00 am - 12:55 pm	4
Mecánica	Jorge Ernesto López	Miércoles	09:00 am - 10:55 am	2
Mecánica	Lorena Sánchez	Miércoles	09:00 am - 10:55 am	3
Mecánica	José Luis Gutiérrez	Jueves	09:00 am - 10:55 am	1

Después me dí a la tarea de investigar que librerías o aplicaciones usar para la realización de una interfaz grafica en Python descubrí mucha información de diversos entornos, y al final me gué por PyQt5 el cuál elegí principalmente ya que había más información y tutoriales sobre cómo manejarlo que los demás.

Descargué Python 3.9, PyQt5 y QTDesigner como app para la elaboración de la vista. Una vez sabiendo que era funcionable le pasé la información de este entorno al frintend para que comenzara con la elaboración de las ventanas diciéndole el funcionamiento detallado de estas.

La imagen siguiente fué el ejemplo de la realización de la pantalla inicial que realicé como prueba básica para comprobar que todo lo instalado estuviera corriendo en perfecto orden y sobre la cuál se iba a editar:

COMIENZO INTERFAZ

```
//Comenzamos importando las librerías necesarias
//Librería de sistema
import sys
//Librería de PyQt5, Importar los módulos
from PyQt5 import QtWidgets, QtCore

//Creamos una aplicación con una clase llamada QApplication
app = QtWidgets.QApplication(sys.argv) //Espera una lista de argumentos que pasaremos mediante sys
//Crearemos una ventana
ventana = QtWidgets.QWidget() //Representa una ventana grafica especifica

//Ingresamos metodos basicos a modificar
//Tamaño
ventana.resize(400, 200)
//Titulo de la ventana
ventana.setWindowTitle("Proyecto")
//Mostrar ventana con el metodo show
ventana.show()

//Para ejecutar
exit(app.exec())
```

Después realicé un manual de usuario para que así todos tuviéramos una idea más clara sobre el uso de la aplicación principalmente por la necesidad de saber las partes técnicas de este y así implementarlo de manera más rápida y optima, así como también me esmeré en la realización del diagrama de usuario para tener una vista grafica de los accesos que se realizarán al interactuar con cada botón de las ventanas:

MANUAL DE USUARIO BÁSICO PARA GENERADOR DE HORARIOS

Manual de usuario:

Para utilizar el programa con total eficiencia creamos esta sección llamada manual de usuario, dedicada en hacer la experiencia de este programa más sencilla y confortante

Al iniciar el sistema te aparecerán dos botones, cada uno con su temática principal, el primer botón llamado "Agregar profesor"

Y el segundo botón llamado

"Generar horario"

Al hacer click en el primer botón nos lanza a una segunda pestaña o pantalla

La cual sirve para efectivamente agregar a un nuevo profesor

En esta sección podemos realizar 3 cosas

Primero colocar el nombre completo del profesor que se agregará, esto es dando click en el recuadro blanco de la derecha

Después podemos también seleccionar los horarios en los días que el profesor no podrá, se realiza igual dando click en el primer recuadro

En este aparecerá una lista desplegable del día de la semana en el cual seleccionará el horario

Y en el recuadro que le sigue justamente el horario que no puede

(Se pueden agregar diferentes horarios y días según las necesidades del profesor)

Y la tercera cosa realizable es asignarle una o varias materias al profesor

Al darle click al recuadro blanco de la derecha de "Materia(s) asignada(s)"

Se desplegará igual una lista con dos nombres de carreras

"ICOM"

e

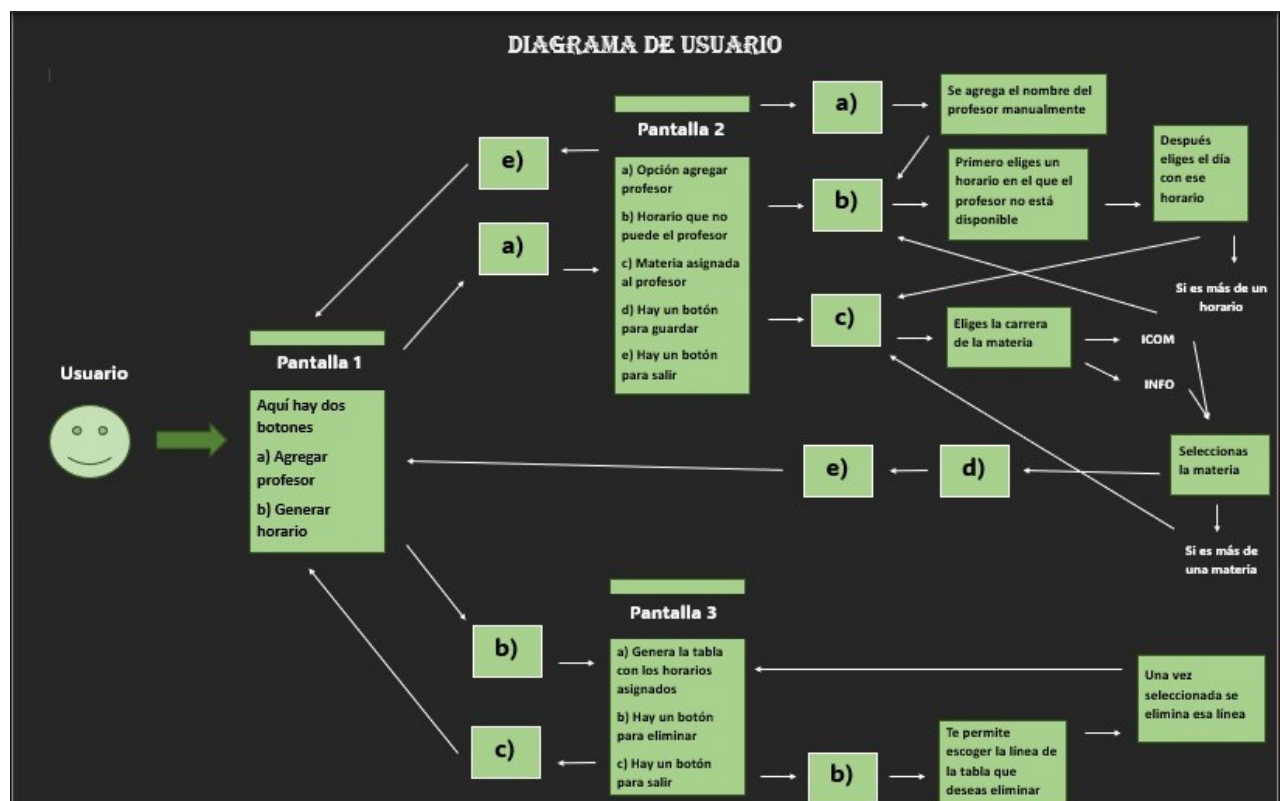
"INFO"

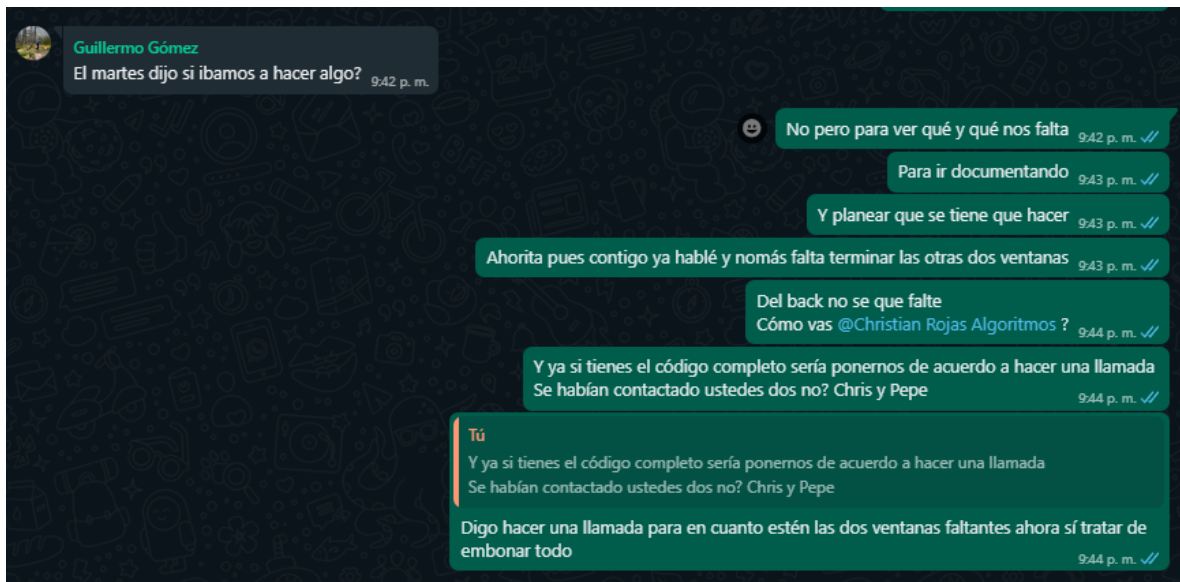
Usted cómo usuario elegirá la carrera que imparte el profesor y al seleccionarla volverá a aparecer otra lista desplegable de todas las materias de esa carrera, seleccionando así usted la que su profesor imparte

Una vez finalizado el llenado de este pequeño formulario dará click en el botón de más abajo

Llamado "Guardar"

Para poder agregar al profesor a el horario





Además de que ayudé un poco en el rol de ejercer presión a los demás integrantes para poder estar todos al tanto de cada cambio realizado y el cómo va el proceso de cada sección intentando dar entrada para poder ayudar en cada una de sus específicas partes.

Una vez solucionado todo esto aún sentía que no avanzábamos lo suficiente por lo cual acudimos al profesor, y tener una idea más clara de que otra sección podía cumplir en mi rol sin quitarle trabajo a los demás, este nos dió la idea de que fueran dos secciones de backend, uno creado por mi compañero Christian y el otro directamente mío, pero con un método diferente al primero, el cual sería con el método de fuerza bruta. Así al final el tester vería cual es el más optimo a implementar.

Una vez sabiendo esto, le dí un repaso a todo lo referente a algoritmos de fuerza bruta para tener un ejemplo claro de el cómo comenzaría realizando. Empecé igual que todos basándose ya en los bocetos y avances dados con antelación la semana pasada, creando un algoritmo a mi parecer más simple y usando un documento de Excel como base de datos.

Cree un prototipo demasiado básico de la creación del algoritmo, básicamente de la estructura que debería llevar con más comentarios que código que me ayudarían para comenzar con la realización de esta sección.

Busqué información sobre como es el funcionamiento de los algoritmos de fuerza bruta y conocí las diferencias de estas a las de los algoritmos de backtracking, aprendí que la fuerza bruta en los algoritmos implica probar todas las combinaciones posibles sin considerar mucho los problemas que podrían intervenir después, en cambio, el backtracking realiza una búsqueda más inteligente, deteniéndose en soluciones parciales que no pueden llevar a una solución válida y retrocediendo para probar otras opciones. Además, para ser más precisos en cuánto al código, el de fuerza bruta suele llevarse a cabo con bucles anidados mientras que el otro utiliza búsquedas recursivas. El backtracking es más elaborado, pero más eficiente.

Hice un texto simple del cómo funcionaría el programa para darme la clara idea de cómo inicializarlo:

Lo empezaría como la mayoría de los códigos que me han agradado realizar, iniciando con un menú, seguramente durante el proceso cambiaré algunos datos, cosas o haré movimientos extras además de que le agregaré la parte de los bucles (Cabe aclarar que lo hice mientras trabajaba):

Idea basica del código para generar horarios

- Primeramente es agregar las librerias esenciales para poder trabajar con este algoritmo el cual se implementará con el metodo visto anteriormente en clase de: fuerza bruta.
- Después agregaremos la liga del o los documentos que usaremos de excel como base de datos para poder agregar la información.
- La información dentro de el/los documentos de excel será: La lista de materias divididas en dos secciones, INFO con 20 materias e ICOM con 43 materias, con la especificación de cuales requieren uso de computadoras y cuales no. Y como segunda lista será una de salones, llenando de X1 a X22 igual con las especificaciones de cuales tienen y no tienen computadoras.

#Hacer un menú con las dos opciones de la pantalla principal

GENERADOR DE HORARIOS

MENÚ

1- REGISTRAR PROFESOR
2- GENERAR HORARIO

if 1

Print("Ingrese el nombre del profesor: ")

Print("Ingrese horario en que el profesor no puede asistir: ")

#Aquí aparecerá una lista de horarios desplegable para poder elegir

09:00 - 10:55

11:00 - 12:55

01:00 - 02:55

Print("Ingrese día que el profesor no puede asistir en el horario elegido: ")

#Aquí aparecerá una lista de días desplegable para poder elegir

Lunes

Martes

Miercoles

Jueves

Viernes

Sabado

Print("Si aún tiene más horarios para no asistir presione 1, si no, presione 2: ")

Al precionar 1 habrá un return para volver a escoger otro horario extra

Al precionar 2 podrá pasar a la siguiente pregunta

Print("Seleccione la carrera que el profesor imparte: ")

#Aquí aparecerán dos opciones: 1- ICOM 2- INFO

#Una vez seleccionada la carrera se desplegará la lista de materias

Print("Seleccione la(s) materia(s) que el profesor imparte: ")

#Podrá seleccionar las que se deseen

#Una vez finalizado el llenado de formulario podrá guardar los cambios

else 2

Aquí simplemente se generará el horario, habrá un botón para poder eliminar una línea de la tabla de cualquier horario.

Avancé con gran desarrollo del código llegando a esta parte:

```
"""
GENERADOR DE HORARIOS
EQUIPO 3 BACK 2
"""

profesores = []

def agregar_profesor():
    nombre = input("Ingrese el nombre del profesor: ")
    horas_no_trabajo = input("Ingrese las horas que el profesor no puede
trabajar: ")
    dia_no_trabajo = input("Ingrese el día sobre el horario que el
profesor no puede trabajar: ")
    materia = input("Ingrese la materia que imparte el profesor: ")

    profesor = {
        "Nombre": nombre,
        "Horas_no_trabajo": horas_no_trabajo,
        "Dia_no_trabajo": dia_no_trabajo,
        "Materia": materia
    }

    profesores.append(profesor)
    print("Profesor agregado exitosamente.")

def generar_tabla_horarios():
    print("\nTabla de Horarios para Profesores:")
    print("{:<20} {:<20} {:<20} {:<20}".format("Nombre", "Horas no
trabajo", "Día no trabajo", "Materia"))
    for profesor in profesores:
        print("{:<20} {:<20} {:<20} {:<20}".format(profesor["Nombre"],
profesor["Horas_no_trabajo"], profesor["Dia_no_trabajo"],
profesor["Materia"]))

def main():
    while True:
        print("\nMenú:")
        print("1. Agregar profesor")
        print("2. Generar tabla de horarios")
        print("3. Salir")

        opcion_menu_principal = input("Seleccione una opción: ")

        if opcion_menu_principal == "1":
            # Submenú para agregar profesor
            while True:
                print("\nSubmenú - Agregar Profesor:")
                print("1. Nombre del profesor")
                print("2. Horas que no puede trabajar")
                print("3. Día establecido sobre el horario que no puede
trabajar")

                print("4. Materia que imparte")
                print("5. Volver al menú principal")

                opcion_submenu_agregar = input("Seleccione una opción: ")
```



```

        elif opcion_submenu_agregar == "1":
            agregar_profesor()
        elif opcion_submenu_agregar == "2":
            agregar_profesor()
        elif opcion_submenu_agregar == "3":
            agregar_profesor()
        elif opcion_submenu_agregar == "4":
            agregar_profesor()
        elif opcion_submenu_agregar == "5":
            break
        else:
            print("Opción no válida. Intente de nuevo.")

    elif opcion_menu_principal == "2":
        generar_tabla_horarios()
    elif opcion_menu_principal == "3":
        print("Saliendo del programa. ¡Hasta luego!")
        break
    else:
        print("Opción no válida. Intente de nuevo.")

if __name__ == "__main__":
    main()

```

Después de esto, lo dejé un poco de lado para enfocarme en una de las cosas que principalmente hacían falta, el cual es el uso de archivos, me dediqué a realizar los archivos con las especificaciones de cada uno, en este caso serían los 4 archivos, el primero con el horario justamente acomodado de 9:00 a 4:55 con horarios de 2 en 2 como lo habíamos especificado al principio, se supone que el programa será capaz de lanzar una lista desplegable de estos horarios ya que es el contenido del archivo, el segundo archivo será con los días de lunes a viernes al igual que el anterior siendo una lista desplegable, el tercero con las carreras que se impartirán con la asignación de su materia, con su respectiva respuesta sobre cuales necesitan computadora, llegando finalmente al cuarto archivo el cuál queda definido con la información interna de las aulas y cuales tienen computadora.

Al final cambié de idea, y para facilitar su creación aún más, implementé un archivo extra, el cual es básicamente el de materias, pero serían dos, un archivo únicamente para INFO y uno específicamente con las materias de ICOM.

HORARIOS

```

09:00 am - 10:55 am
11:00 am - 12:55 pm
01:00 pm - 02:55 pm
03:00 pm - 04:55 pm

```

ARCHIVO 1

DIAS

LUNES
MARTES
MIERCOLES
JUEVES
VIERNES
SABADO |

ARCHIVO 2

ICOM

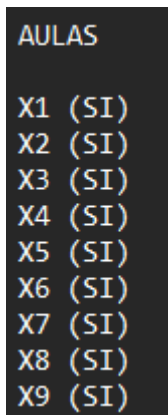
1. FUNDAMENTOS DE LA PROGRAMACION (SI)
2. ETICA Y LEGISLACION
3. LOGICA MATEMATICA
4. PRECALCULO|
5. FUNDAMENTOS DE FISICA
6. INTRODUCCION A LA INGENIERIA
7. PROGRAMACION ESTRUCTURADA (SI)
8. MATEMATICA DISCRETA
9. CALCULO DIFERENCIAL E INTEGRAL
10. MECANICA
11. ADMINISTRACION DE PROYECTOS TECNOLOGICOS
12. EXPRESION ORAL Y ESCRITA
13. PROGRAMACION ORIENTADA A OBJETOS (SI)

ARCHIVO 3 (El “si” de a un lado es que necesita uso de computadora).

INFO|

1. LOGICA Y CONJUNTOS
2. INTRODUCCION A LA FISICA
3. TALLER DE INTRODUCCION A LA COMPUTACION
4. TALLER DE COMUNICACION ORAL Y ESCRITA
5. INTRODUCCION A LA COMPUTACION (SI)
6. TALLER DE REDACCION
7. INTRODUCCION A LA PROGRAMACION (SI)
8. ELEMENTOS DE PROBABILIDAD Y ESTADISTICA
9. TALLER DE PROGRAMACION ORIENTADA A OBJETOS (SI)
10. AUDITORIA DE SISTEMAS (SI)
11. SISTEMAS DE INFORMACION FINANCIEROS
12. SISTEMAS DE INFORMACION PARA LA MANUFACTURA

ARCHIVO 4 (El “si” de a un lado es que necesita uso de computadora).



ARCHIVO 5 (El “si” de al lado es que el aula si cuenta con computadora, 20 si tienen y 22 no)

.idea	26/11/2023 06:56 p. m.	Carpeta de archivos	
aulas	26/11/2023 07:35 p. m.	Documento de te...	1 KB
dias	26/11/2023 07:32 p. m.	Documento de te...	1 KB
horarios	26/11/2023 07:31 p. m.	Documento de te...	1 KB
icom	26/11/2023 07:35 p. m.	Documento de te...	2 KB
info	26/11/2023 07:35 p. m.	Documento de te...	1 KB
main.py	26/11/2023 06:56 p. m.	PyCharmCE2023.2	3 KB

Así se van viendo los archivos ya guardados c:

Lo que estoy haciendo:

Estoy logrando que el código pueda leer correctamente los archivos, ya que el código que tengo implementado por el momento solo puede ingresar todo de manera manual, la lectura correcta de los archivos seguramente se me complicará un poco, pero en seguida investigaré como se debe realizar.

También estoy trabajando en implementar la sección de eliminar uno de los acomodos de horario generado.

Lo que falta:

Debo tener un diagrama de algoritmo en el cuál se muestre su funcionamiento, además de el manual de algoritmo, el cual realizaré con el código una vez terminado para poder basarme 100% en el y poder tener comentarios de cada bloque de líneas en los cuales se especifique su funcionamiento.

También hace falta terminarlo a manera de que sea capaz de leer los archivos.

Dificultades presentadas:

Principalmente la dificultad técnica que nos atrasó un poco al inicio fue el conjunto de versiones, me generó un poco de problemas que necesariamente se necesitase cierta versión de un lenguaje para poder instalar las herramientas necesarias.

El echo también de que yo no tenía un rol fijo desbalanceaba un poco las actividades individuales de los demás y me quedaba a la incertidumbre de que debía hacer para poder alcanzar un avance eficiente.

La cantidad de tiempo presentada para el proceso de creación fue otra de las problemáticas que enfrenté ya que el rol específico que se me asignó fue a una semana de la entrega lo cual me acortó el plazo tanto a mi con mi código como a los de mi equipo ya que se los mandaría en un tiempo muy pequeño para concluir con el testing y el ensamblado.

Además de que la comunicación no fue la óptima de parte de algunos integrantes en específico para poder avanzar de manera eficiente en el proceso por partes. Pese a que el administrador y yo tratábamos de comunicarnos, por días no daban señales, ni cooperación, y los avances que iban teniendo no eran los que se esperaban.

Fue muy complicado lograr que el código leyera los archivos por la confusa información que venía dentro de estos, principalmente el tener que separar cada bloque como debería.

Otro de mis problemas presentados fue el manejo de mi entorno de desarrollo, o sea, de mi computadora, por que en un momento necesité instalar Tkinter por que al final decidimos no usar el PyQT debido a que mi compañero encargado al front no lo terminó de realizar y se encargó de esto el administrador, y al intentar instalar desde el cmd tkinter no me lo permitía, era ya muy tarde para ello y solo me enfoqué en mi parte del backend.