



ACME

PROYECTO ACME

Sistema de Gestión de Asistencia
Académica

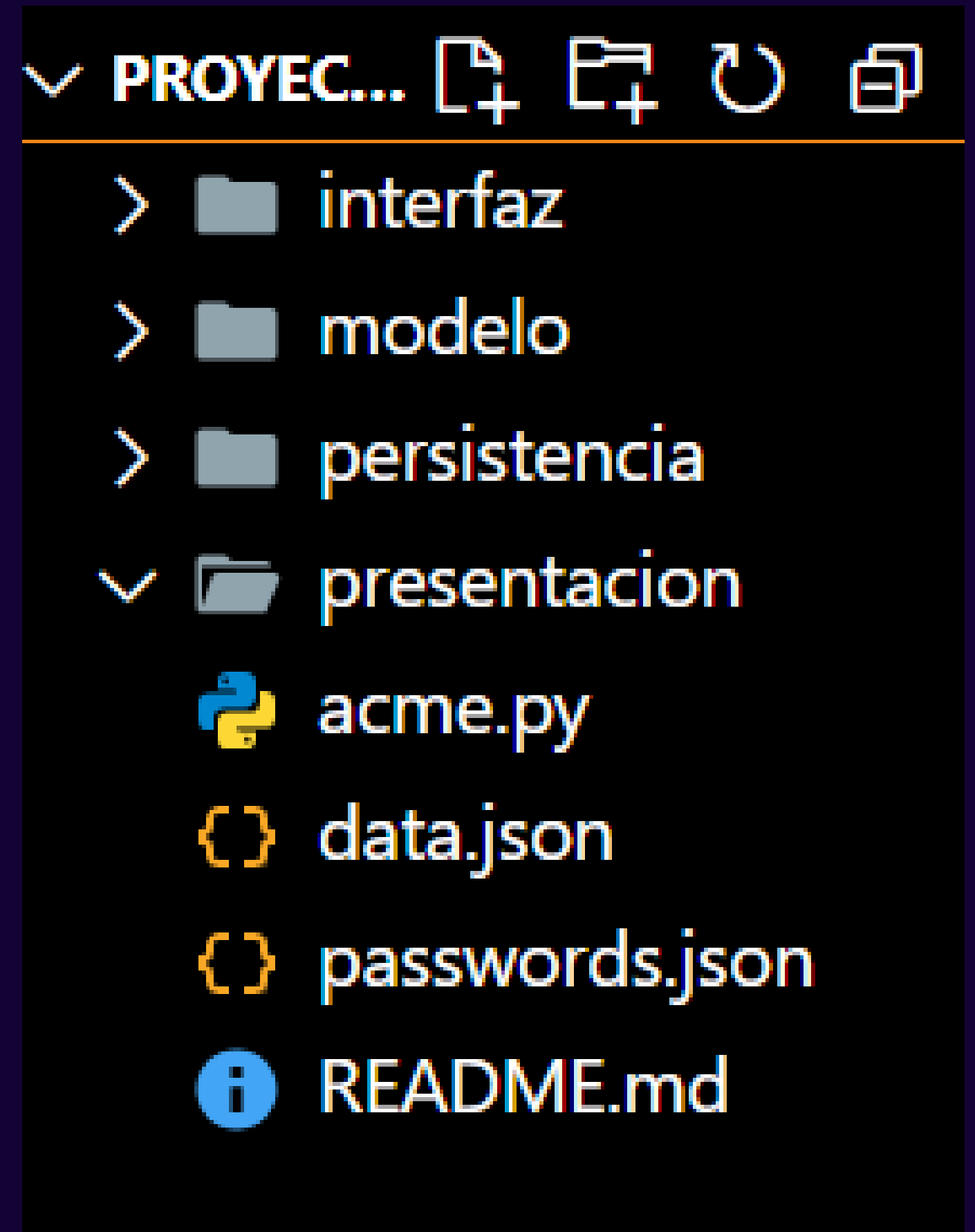


Por Eimer Santiago. martes, 24 de septiembre de
2024.

RESUMEN

Primero procedimos analizar bien el proyecto, todos los requerimientos, y realizar posteriormente el esquema de como iniciar el proyecto, por medio de su arquitectura.

Se especifica las carpetas para la interfaz, modelo, persistencia y la presentación, y por fuera queda el archivo principal y los .json



ESTRUCTURAS UTILIZADAS:



```
while True: # Entramos al menú
    dDataAll = cargarDatos(pathData)
    menu()
    opcion = input("Introduce tu opción (A, B, C, D, etc): ").strip().upper() # quita espacios y pasa a mayuscula
    if opcion == 'A':
        registrarGrupo(dDataAll,pathData)
    elif opcion == 'B':
        registrarModulo(dDataAll,pathData)
    elif opcion == 'C':
        registrarEstudiante(dDataAll,pathData)
    elif opcion == 'D':
        registrarDocente(dDataAll,pathData)
```

ubicada en el **acme.py**

Este while lo utilizamos para entrar al menú, dentro de él carga los datos y luego procede a cada sección.

ESTRUCTURA EN EL LOGIN

Estructura utilizada para la contraseña encriptada que nos estan solicitando con el hashlib y utilizamos un json para guardar.


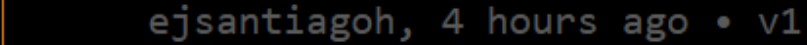
interfaz >  login.py >  login

You, 38 minutes ago | 2 authors (ejsantiagoh and one other)


```
1  import json
2  import hashlib
3
4  # Definimos un usuario y una contraseña predeterminada
5  USERNAME = 'usuario'
6  PASSWORD = 'SISGESA'
7
8  # Función para encriptar la contraseña con SHA-256
9  def hash_password(password):
10     return hashlib.sha256(password.encode()).hexdigest()
11
12  # Guardamos el usuario y la contraseña en un archivo JSON
13  def guardarCredenciales():
14     credentials = {
15         'username': USERNAME,
16         'password': hash_password(PASSWORD)
17     }
18     with open('passwords.json', 'w') as f:
19         json.dump(credentials, f)
20
21
```


persistencia

Tenemos todo lo pertinente a cargar y guardar datos, utilizamos la importación del path y de json para finalmente tener almacenados los registros.

```
persistencia >  persistencia.py > ...  
You, 43 minutes ago | 2 authors (ejsantiagoh and one other)  
1 import json  
2 from pathlib import Path  
3  
4 pathData = "data.json"  
5  ejsantiagoh, 4 hours ago • v1  
6 def guardarDatos():  
7     datos = {}  
8     with open('data.json', 'w') as archivo:  
9         json.dump(datos, archivo)  
10  
11  
12 def cargarDatos(arch):  
13     archivo = Path(arch)  
14     grupo = {}  
15     if archivo.is_file(): #True si existe y es un archivo  
16         try:  
17             with open(arch, "r") as fd: #Abrir el archivo  
18                 grupo = json.load(fd)  
19                 if not fd.closed:  
20                     fd.close()  
21         except Exception as e:  
22             print("Error al abrir el archivo data\n", e)  
23     else:  
24         print("Error el archivo no existe")  
25         input("Presione cualquier tecla para continuar...")  
26     return grupo  
27
```

ARQUITECTURA UTILIZADA

✓  interfaz

>  modelo

✓  persistencia

 acme.py

{ } data.json M

{ } passwords.js... M

A stylized world map is visible in the background, rendered in a light teal color. The map shows the continents of North and South America. Two large, solid circles are positioned on the map: a teal circle in the upper left quadrant and an orange circle in the lower left quadrant. The text '¡GRACIAS!' is centered horizontally across the middle of the image.

¡GRACIAS!