

PROYECTO #1

Sistema de gestión de asistencia académica

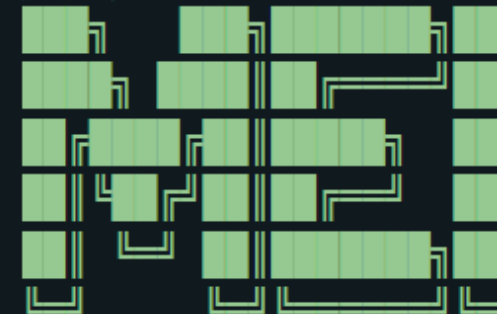
Laia Carrillo
Python



Estructura del sistema



Creación Menu Opciones

```
if login():
    sw = True
    while sw:
        print('

        ')
        opc = menu()
        match opc:
            case 1:
```

```
case 'a':
    limpiarPantalla()
    grupos = {}
    archGrupos = 'ACME/json/Grupos.json'
    grupos = cargarGrupos(archGrupos)
    grupos = registroGrupos(grupos)
    limpiarPantalla()
```

```
case 'f':
    estudiantes = {}
    archEstudiantes = 'ACME/json/Estudiantes.json'
    estudiantes = cargarEstudiantes(archEstudiantes)
    modulos = {}
    archModulos = 'ACME/json/Modulos.json'
    modulos = cargarModulos(archModulos)
    grupos = {}
    archGrupos = 'ACME/json/Grupos.json'
    grupos = cargarGrupos(archGrupos)
    docentes = {}
    archDocentes = 'ACME/json/Docentes.json'
    docentes = cargarDocentes(archDocentes)
    limpiarPantalla()
    consultaInformacion(estudiantes, modulos, grupos, docentes)
    limpiarPantalla()
```

Decisiones de programación

Funciones Clave:

- Función de lectura y escritura en JSON de manera eficiente para manejar grandes volúmenes de datos.
- Módulo de autenticación de usuarios, que incluirá roles y permisos según el tipo de usuario.

Manejo de Usuarios:

- Creación de perfiles de usuarios con diferentes niveles de acceso a la información.
- Control de acceso basado en roles: administradores, usuarios regulares, y usuarios de solo lectura.

```
> def registroGrupos(grupos): ...

> def registroModulos(modulos): ...

> def registroEstudiantes(estudiantes): ...

> def registroDocentes(docentes): ...

    # Función para convertir objetos datetime a cadenas
> def convertir_tiempos_a_str(asistencia): ...

> def registroAsistencia(asistencia): ...

> def asignarGruposyModulos(estudiantes_json, grupos_json, modulos_json): ...

> def consulta_estudiantes_Por_Grupo(estudiantes, grupos): ...

> def consulta_estudiantes_Por_Modulo(modulos, estudiantes): ...

> def consulta_Docentes_Por_Modulo(modulos, docentes): ...

> def consulta_Estudiantes_Por_Docente(estudiantes, docentes, modulos): ...

> def consultaInformacion(estudiantes, modulos, grupos, docentes): ...

> def generacionInformes(asistencia, estudiantes, modulos): ...
|
> def informe_tardanza(asistencia): ...

> def informe_salida_temprana(asistencia): ...

> def informe_asistencia_perfecta(asistencia, estudiantes): ...

> def informe_porcentaje_asistencia(asistencia, estudiantes, modulos): ...
```