<!--Acme Education-->

{Proyecto Python}

Sistema de gestión de asistencia académica

Print("Marlon Chacón")
Grupo = “C4”
Campuslands

Contenidos

Introducción {

Contexto del Problema {

Acme Education necesita modernizar su sistema de gestión de asistencia debido a las ineficiencias del registro manual, que afectan la precisión y los procesos administrativos

}

Objetivo {

Automatizar el registro de asistencia para mejorar la precisión, eficiencia y generacion de informes claros en apoyo a los procesos institucionales

}

DESCRIPCIÓN DEL PROYECTO {

Carateristicas claves{

El programa cuenta con un registro de:

- Grupos
- Módulos
- Estudiantes
- Docentes

También permite:

- Registro de Asistencia

Y brinda consultas e informes de cada uno de los items anteriores }

Requisitos Cumplidos{

El Programa brinda la posibilidad de registrar asistencia automáticamente, con el uso del código asignado a cada estudiante, de esta forma los informes son mas precisos }

SOLUCIÓN TÉCNICA {

```python
def encriptar_contraseña(contraseña):
    return hashlib.sha256(contraseña.encode('utf-8')).hexdigest()


def verificar_contraseña(contraseña_ingresada, hash_guardado):
    # Generar el hash de la contraseña ingresada
    hash_ingresado = encriptar_contraseña(contraseña_ingresada)
    # Comparar con el hash guardado
    return hash_ingresado == hash_guardado
```

```python
def leerPrecio():
    while True:
        try:
            precio = int(input("Precio del libro: "))
            if precio < 500:
                print(">>> Error. Precio incorrecto")
                continue
            return precio
        except ValueError:
            print(">>>> Error. Precio inválido")
```

```python
f Menu2():
    while True:

        print("")
        print(">>> MENU <<<".center(50))
        print("*" * 60)
        print("1. Registro de Grupos")
        print("2. Registro de Módulos")
        print("3. Registro de Estudiantes")
        print("4. Registro de Docentes")
        print("5. Registro de Asistencia")
        print("6. Consultas de Información")
        print("7. Generar Informes")
        print("8. Cambiar Contraseña")
        print("9. Salir")
        print("*" * 60)

        print("Opcion? >>> ", end="")
        try:
            opcion = int(input())
            if opcion < 1 or opcion > 9:
                print("ERROR. Opción NO válida")
                input("Presione cualquier tecla para volver al menu...")
            return opcion

        except ValueError:
            print("ERROR. Opción NO válida")
            input("Presione cualquier tecla para volver al menu...")
```

Estrategias Implementadas:

- Uso de un menú de opciones para la interacción
- Manejo de Errores para evitar que el programa se cierre inesperadamente
- Uso de SHA-256 para la seguridad de las contraseñas

}

SOLUCIÓN TÉCNICA {

```
"Modulos": {
    "01": {
        "Nombre": "Ingles",
        "Duracion Semanas": 4,
        "Fecha Inicio": "01/10/2024",
        "Fecha Fin": "30/10/2024",
        "Hora Inicio": "09:00",
        "Hora Fin": "11:00",
        "Integrantes": {
            "01": "Marlon",
            "02": "Sara",
            "03": "Jorge",
            "04": "Jose"
        },
        "Docente/s": {
            "37705895": {
                "Nombre": "Vannesa",
                "Estudiantes_Asignados": 4,
                "Estudiantes_Asignados_List": [
                    "01",
                    "02",
                    "03",
                    "04"
                ]
            }
        },
        "Asistencia": {
            "01/10/2024": {
                "01": {
                    "Hora de Llegada": "09:10",
                    "Hora de Salida": "10:50"
                },
                "02": {
                    "Hora de Llegada": "09:05",
                    "Hora de Salida": "11:00"
                }
            },
            "02/10/2024": {
```

```
    "Informes": {
        "EstudiantesTarde": [
            "01",
            "03"
        ],
        "EstudiantesRetiradosAntes": [
            "01"
        ],
        "EstudiantesSinFaltas": [
            "02",
            "04"
        ]
    }
,
02": {
    "Nombre": "Matem\u00c3\u00a1ticas",
    "Duracion Semanas": 4,
    "Fecha Inicio": "01/10/2024",
    "Fecha Fin": "30/10/2024",
    "Hora Inicio": "11:30",
    "Hora Fin": "13:30",
    "Integrantes": {
        "01": "Marlon",
        "05": "Luis",
        "06": "Ana"
    },
    "Docente/s": {
        "1098407324": {
            "Nombre": "Orbin",
            "Estudiantes_Asignados": 3,
            "Estudiantes_Asignados_List": [
```

Estructura de datos utilizada:

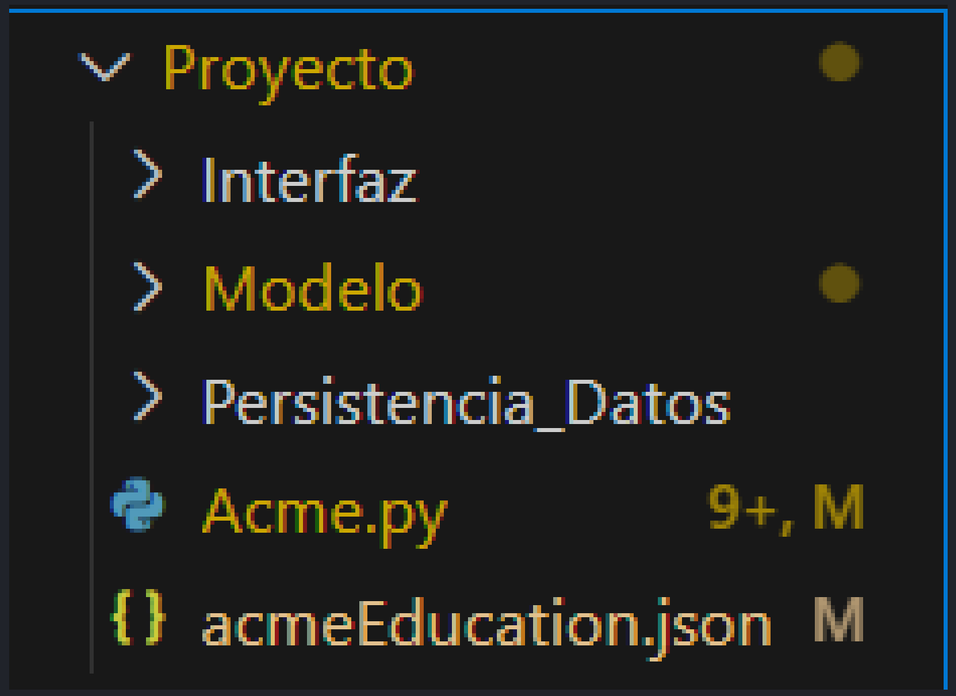
- Los diferentes datos registrados en el sistema se fueron almacenando en un JSON que contenía un diccionario, con diferentes claves, valores, listas etc
- El registro de estos Datos dentro del JSON permitia la persistencia de los datos, para dar los difrentes informes y consultas solicitados.

}

Estructura del proyecto {

Diseño modular

Se eligió un diseño modular, organizando la persistencia, lecturas, modelo, menú y archivo principal en subcarpetas separadas para una mejor gestión de las funciones del proyecto.



}

Estructura del proyecto {

Manejo de persistencia

Para facilitar la consulta de información se utilizó un JSON que almacenó todos los datos registrados de forma permanente utilizando funciones para guardar y cargar para posteriores consultas e informes

```python
import json
from pathlib import Path

def guardar(lib, arch):
    with open(arch, "w") as fd:
        json.dump(lib, fd)

    if not fd.closed:
        fd.close()


def cargar(arch):
    archivo = Path(arch)
    lib = {}
    if archivo.is_file(): # True: si existe y es un archivo
        try:
            with open(arch, "r") as fd:
                lib = json.load(fd)

            if not fd.closed:
                fd.close()
        except Exception as e:
            print(">>> Error al abrir el archivo.\n" + e)

    else:
        print(">>> Error. El archivo no existe")
        input(">>> Presione cualquier tecla para continuar")

    return lib
```

}

Arquitectura {

Consultas

CÓDIGO

ACME
EDUCATION

Datos
Registrados

AUTENTICACION

Informes

}

Principales Funciones {

```python
from Persistencia_Datos.Persistencia import guardar, cargar
from Modelo.lecturas import *
from datetime import datetime, timedelta

def registrarAsisSalida(lib, arch): ...

def registrarAsisLlegada(lib, arch): ...

def asignarGrupo(lib): ...

def insertDocAsigEst(lib, arch): ...

def insertarGrup(lib, arch): ...

def insertarMod(lib, arch): ...

def insertarEst(lib, arch): ...

def consultarGroup(lib): ...

def consultarMod(lib): ...

def consultarDoc(lib): ...

def consultarDocImpar(lib): ...

def consultarEstudiantesTarde(lib): ...
```

```python
def eleccionGt(msg): ...

def eleccion(msg): ...

def leerHora(): ...

def leerFecha(): ...

def nomDocen(): ...

def leerCedul(): ...

def siglaGrup(): ...

def duraSem(): ...

def leerEdad(): ...

def leerSexo(): ...

def nomGrup(): ...

def nomMod(): ...

def nomEst(): ...
```

}

Informes {

```
Los estudiantes que llegaron tarde al Módulo:
Código: 01  >>>>  Nombre: Ingles son:

Código del Estudiante          Nombre del Estudiante
=============================================
01                             Marlon
03                             Jorge

Presione cualquier tecla para volver al menú...
```

Consultas {

```
Los estudiantes matriculados en el Grupo:
Codigo: c4 >>>>  Nombre: Python Lovers  son:

Codigo del Estudiante          Nombre del Estudiante
=============================================
01                             Marlon
02                             Sara
03                             Jorge
04                             Jose
05                             Luis
06                             Ana
07                             Carlos
08                             Lucia
09                             Diego
10                             Elena
11                             Mario
12                             Andrea
13                             Sofia
14                             Pablo
15                             Gabriela
16                             Ricardo
17                             Diana
18                             Felipe
19                             Isabel
20                             Sergio

Presione cualquier tecla para volver al menu...
```

}

Conclusión {

El programa cumplió con el objetivo inicial, mediante una estrategia y unas metodologías desarrolladas en el transcurso de la codificación, y está puesto para su utilización

EJECUCIÓN -->

<!--Estudio Shonos-->

Gracias {

<Por="MARLON CHACÓN/>

}