**APLICATIVO DE CONSOLA**

crear un aplicativo de consola, en python, con un microsistema, una simulación de inicio de sesión y registro.

ROLES (administrador, usuario).

todo trabajado con arrays, listado de usuarios, con la capacidad de ordenarlo de forma ascendente y descendente.

REGISTRO: nombre, apellido, rol, correo, contraseña.

USUARIO: correo y contraseña.

rol administrador: lista de usuarios, ordenarlos, crear y listar roles.

Rol usuario: listar usuarios y roles.

1. Primero lo que se va a realizar es: crear un array para almacenar a los usuarios registrados.

usuarios = []

1. Luego creamos un array para almacenar a todos roles disponibles. Tener mucho cuidado a la hora de ingresar el rol, lo cual conlleva a escribir exactamente como se está pidiendo.

roles = ["administrador", "usuario"]

1. BIEN, UNA VEZ CREADOS LOS ARRAYS, PASAMOS A CREAR UNA FUNCIÓN PARA REGISTRAR UN NUEVO USUARIO.

def registrar\_usuario():

1. Pasamos a pedir al usuario que por favor ingrese sus datos.

nombre=input("Ingrese el Nombre del Usuario: ")

apellido= input ("Ingrese Apellido: ")

email=input("ingrese Email: ")

password=input("Ingrese Password: ")

rol = input("ingrese su rol(administrador o usuario): ")

1. Pasamos a validar el rol ingresado.

 if rol not in roles:

     print("rol no válido. intente de nuevo")

     return

1. Validamos también, que el correo no esté registrado.

for usuario in usuarios:

    if usuario["email"]==email:

        print("email ya existe, ingrese uno diferente.")

        return

1. Creamos un diccionario con los datos del usuario.

usuario={"nombre":nombre,

          "apellido": apellido,

          "email":email,

           "password":password,

           "rol": rol}

1. Agregamos el usuario al array de usuarios.

usuarios.append(usuario)

1. Luego de registrarse, vamos a mostrar un mensaje de éxito

print("\n Usuario Registrado con exito!.")

1. CREAMOS UNA FUNCIÓN PARA INICIAR SESIÓN.

def iniciar\_sesion ():

1. Pasamos a pedir al usuario que ingrese su correo y contraseña

 email = input("ingrese su email: ")

 password = input("ingrese su password: ")

1. Buscamos al usuario en el array de usuarios

 for usuario in usuarios:

        if usuario["email"]==email and usuario["password"]==password:

1. Mostramos un mensaje de bienvenida

print(f"bienvenido {usuario['nombre']} {usuario['apellido']} .tu rol es -> {usuario['rol']}.")

1. Retornamos al usuario encontrado

return usuario

1. Si no se encuentra el usuario, mostrar un mensaje de error.

print("email o password incorrectos. intente de nuevo")

1. Retornar none

return None

1. CREAMOS UNA FUNCIÓN PARA LISTAR LOS USUARIOS REGISTRADOS

def listar\_usuarios():

1. Verificamos que haya usuarios registrados

if len(usuarios) == 0:

    print("No hay Usuarios registrados. ")

    return

1. Mostrar los datos de cada usuario en una tabla

print("lista de usuarios regitrados: ")

print("-"\*75)

print(f"{'nombre':<20}{'apellido':<20} {'email':<20}{'rol':<20}")

print("-"\*75)

for usuario in usuarios:

  print(f"{usuario['nombre']:<20}{usuario['apellido']:<20}{usuario[ 'email']:<20}{usuario['rol']:<20}")

    print("-"\*75)

1. CREAR UNA FUNCIÓN PARA ORDENAR LOS USUARIOS POR NOMBRE O APELLIDO

def ordenar\_usuarios():

1. Pedir al usuario que ingrese el criterio de ordenamiento (nombre o apellido)

criterio = input("ingrese el criterio de orden (nombre o apellido): ")

1. Validar que el criterio sea válido

if criterio not in ["nombre", "apellido"]:

    print ("criterio no válido. inténtelo de nuevo.")

    return

1. Pedir al usuario que ingrese el sentido de ordenamiento (ascendente o descendente)

sentido = input("ingrese el sentido de ordenamiento(ascendente o descendente)")

1. Validar que el sentido sea válido.

if sentido not in ["ascendente", "descendente"]:

      print("sentido no válido, intente nuevamente.")

      return

1. Ordenar el array de usuarios según el criterio y el sentido elegidos

usuarios.sort(key=lambda x:x[criterio], reverse=(sentido=="descendente"))

1. Mostramos un mensaje de éxito

print(f"usuarios ordenados por {criterio} en sentido {sentido}.")

1. CREAR UNA FUNCIÓN PARA CREAR UN NUEVO ROL

def crear\_rol():

1. Pedir al usuario que ingrese el nuevo rol

nuevo\_rol = input("ingrese el nombre del nuevo rol: ")

1. Validar que el rol no exista

if nuevo\_rol in roles:

  print("el rol agregado ya existe, inténtelo de nuevo.")

  return

1. Agregar el rol al array de roles

 roles.append(nuevo\_rol)

1. Luego pasamos a mostrar un mensaje de éxito

print("rol agregado satisfactoriamente.")

1. CREAMOS UNA FUNCIÓN PARA LISTAR LOS ROLES DISPONIBLES.

def listar\_roles():

1. Verificamos que haya roles disponibles

if len(roles) == 0:

 print("no hay roles dsiponibles.")

 return

1. Mostramos los roles en una lista

print("Lista de roles disponibles:")

for rol in roles:

    print(f"-{rol}")

1. CREAR UNA FUNCIÓN PARA MOSTRAR EL MENÚ DE OPCIONES SEGÚN EL ROL DEL USUARIO

def mostrar\_menu(usuario):

1. Mostrar el menú según el rol de usuario

if usuario["rol"] == "administrador":

        print("menú de opciones para administrador.")

        print("1. listar usuarios")

        print("2. ordenar usuarios")

        print("3. crear rol")

        print("4. listar roles")

        print("5. salir")

elif usuario["rol"] =="usuario":

        print("menú de opciones para usuario.")

        print("1. listar usuarios.")

        print("2. listar roles")

        print("3. salir")

else:

        print(f"no hay opciones para el rol {usuario['rol']}.")

1. CREAR UNA FUNCIÓN PARA EJECUTAR LA OPCIÓN ELEGIDA POR EL USUARIO

def ejecutar\_opcion(opcion,usuario):

1. Ejecutar la opción según el rol de usuario

if usuario["rol"] == "administrador":

      if opcion == "1":

        listar\_usuarios()

      elif opcion == "2":

        ordenar\_usuarios()

      elif opcion == "3":

        crear\_rol()

      elif opcion == "4":

        listar\_roles()

      elif opcion == "5":

       return False

      else:

       print("Opción inválida. Intente de nuevo.")

    elif usuario["rol"] == "usuario":

      if opcion == "1":

          listar\_usuarios()

      elif opcion == "2":

          listar\_roles()

      elif opcion == "3":

         return

      else:

         print("Opción inválida. Intente de nuevo.")

    else:

       print(f"no hay opciones para el rol {usuario['rol']}.")

1. Retornar true si se quiere continuar con el programa

return True

1. Creamos una variable para controlar el ciclo principal del programa

continuar = True

1. Mostrar un mensaje de bienvenida

print("bienvenidos al aplicativo desarrollado por jack\_luna.")

1. Iniciar el ciclo principal del programa

while continuar :

1. Mostrar las opciones de inicio

print("opciones de inicio: ")

print("1. REGISTRARSE")

print("2. INICIAR SESIÓN")

print("3. SALIR")

1. Pedir al usuario que ingrese una opción

opcion = input("ingrese una opción: ")

1. Ejecutar la opción elegida

if opcion == "1":

      registrar\_usuario()

elif opcion == "2":

      usuario = iniciar\_sesion()

      if usuario is not None:

1. Iniciar un ciclo secundario para mostrar el menú según el rol del usuario

while True:

   mostrar\_menu(usuario)

   opcion = input("ingrese una opción: ")

   continuar = ejecutar\_opcion(opcion, usuario)

   if not continuar:

      break

elif opcion == "3":

  continuar = False

else:

      print ("la opción ingresada no es válida!, intente de nuevo.")

1. Mostrar un mensaje de despedida

print("GRACIAS POR USAR EL APLICATIVO JACK\_LUNA. HASTA PRONTO")