

Python EJ 19

CRUD Centro deportivo

En este ejercicio hay que crear una base de datos para un centro deportivo en la que se puedan registrar clientes, ver a los clientes registrados, actualizar o borrar clientes, registrar, visualizar, actualizar o borrar actividades, registrar, visualizar, actualizar o borrar entrenadores y registrar, visualizar y borrar inscripciones.

- Crear una base de datos
- Conectarse a la base de datos
- Definir funciones para las opciones del menú
- Definir funciones para crear los menús
- Crear el menú principal

bdd.py

```
import mysql.connector

def conectar(): #Conecta con la base de datos
    conexion = mysql.connector.connect(
        host="localhost",    # Cambia según tu configuración
        user="root",         # Cambia según tu configuración
        password="curso",    # Cambia según tu configuración
        database="centro_deportivo"
    )
    return conexion
```

clientes.py

```
from bdd import conectar

def crear_cliente(nombre, edad, tipo_membresia):
    conexion = conectar()
    cursor = conexion.cursor()
    consulta = "INSERT INTO clientes (nombre, edad, tipo_membresia)
VALUES (%s, %s, %s)"
    cursor.execute(consulta, (nombre, edad, tipo_membresia))
    conexion.commit()
    cursor.close()
    conexion.close()

def leer_clientes():
    conexion = conectar()
    cursor = conexion.cursor()
```

```

        cursor.execute("select * from clientes")
        clientes = cursor.fetchall()
        cursor.close
        conexion.close
        return clientes

def actualizar_cliente(id_cliente, nuevo_nombre, nueva_edad,
nuevo_tipo_membresia):
    conexion = conectar()
    cursor = conexion.cursor()
    consulta = "UPDATE clientes SET nombre = %s, edad = %s,
tipo_membresia = %s where id_cliente = %s"
    cursor.execute(consulta, (nuevo_nombre, nueva_edad,
nuevo_tipo_membresia, id_cliente))
    conexion.commit()
    cursor.close()
    conexion.close()

def borrar_cliente(id_cliente):
    conexion = conectar()
    cursor = conexion.cursor()
    consulta = "delete from clientes where id_cliente = %s"
    cursor.execute(consulta, (id_cliente,))
    conexion.commit()
    consulta = "delete from inscripciones where id_cliente = %s"
    cursor.execute(consulta, (id_cliente,))
    conexion.commit()
    cursor.close()
    conexion.close()

```

actividades.py

```

from bdd import conectar

def crear_actividad(nombre_actividad, horario, duracion, id_entrenador):
    conexion = conectar()
    cursor = conexion.cursor()
    consulta = "INSERT INTO actividades (nombre_actividad, horario,
duracion, id_entrenador) VALUES (%s, %s, %s, %s)"
    cursor.execute(consulta, (nombre_actividad, horario, duracion,
id_entrenador))
    conexion.commit()
    cursor.close()
    conexion.close()

def leer_actividades():
    conexion = conectar()
    cursor = conexion.cursor()

```

```

        cursor.execute("select*from actividades")
        actividades = cursor.fetchall()
        cursor.close
        conexion.close
        return actividades

def actualizar_actividad(id_actividad, nuevo_nombre_actividad,
nuevo_horario, nueva_duracion, nuevo_id_entrenador):
    conexion = conectar()
    cursor = conexion.cursor()
    consulta = "UPDATE actividades SET nombre_actividad = %s, horario =
%s, duracion = %s, id_entrenador = %s WHERE id_actividad = %s"
    cursor.execute(consulta, (nuevo_nombre_actividad, nuevo_horario,
nueva_duracion, nuevo_id_entrenador, id_actividad))
    conexion.commit()
    cursor.close()
    conexion.close()

def borrar_actividad(id_actividad):
    conexion = conectar()
    cursor = conexion.cursor()
    consulta = "delete from actividades where id_actividad = %s"
    cursor.execute(consulta, (id_actividad,))
    conexion.commit()
    consulta = "delete from inscripciones where id_actividad = %s"
    cursor.execute(consulta, (id_actividad,))
    conexion.commit()
    cursor.close()
    conexion.close()

```

entrenadores.py

```

from bdd import conectar

def crear_entrenador(nombre_entrenador, especialidad):
    conexion = conectar()
    cursor = conexion.cursor()
    consulta = "INSERT INTO entrenadores (nombre_entrenador,
especialidad) VALUES (%s, %s)"
    cursor.execute(consulta, (nombre_entrenador, especialidad))
    conexion.commit()
    cursor.close()
    conexion.close()

def leer_entrenadores():
    conexion = conectar()
    cursor = conexion.cursor()
    cursor.execute("select*from entrenadores")

```

```

    entrenadores = cursor.fetchall()
    cursor.close()
    conexion.close()
    return entrenadores

def actualizar_entrenador(id_entrenador, nuevo_nombre_entrenador,
nueva_especialidad):
    conexion = conectar()
    cursor = conexion.cursor()
    consulta = "UPDATE entrenadores SET nombre_entrenador = %s,
especialidad = %s WHERE id_entrenador = %s"
    cursor.execute(consulta, (nuevo_nombre_entrenador,
nueva_especialidad, id_entrenador))
    conexion.commit()
    cursor.close()
    conexion.close()

def borrar_entrenador(id_entrenador):
    conexion = conectar()
    cursor = conexion.cursor()
    consulta = "delete from entrenadores where id_entrenador = %s"
    cursor.execute(consulta, (id_entrenador,))
    conexion.commit()
    consulta = "delete from actividades where id_entrenador = %s"
    cursor.execute(consulta, (id_entrenador,))
    conexion.commit()
    cursor.close()
    conexion.close()

```

inscripciones.py

```

from bdd import conectar

def crear_inscripcion(id_cliente, id_actividad):
    conexion = conectar()
    cursor = conexion.cursor()
    consulta = "INSERT INTO inscripciones (id_cliente, id_actividad)
VALUES (%s, %s)"
    cursor.execute(consulta, (id_cliente, id_actividad))
    conexion.commit()
    cursor.close()
    conexion.close()

def leer_inscripciones():
    conexion = conectar()
    cursor = conexion.cursor()
    cursor.execute("select inscripciones.id_inscripcion, clientes.*,
actividades.* from (inscripciones inner join clientes) inner join

```

```

actividades where inscripciones.id_cliente = clientes.id_cliente and
inscripciones.id_actividad = actividades.id_actividad")
    clientes = cursor.fetchall()
    cursor.close()
    conexion.close()
    return clientes

def borrar_inscripcion(id_inscripcion):
    conexion = conectar()
    cursor = conexion.cursor()
    consulta = "delete from inscripciones where id_inscripcion = %s"
    cursor.execute(consulta, (id_inscripcion,))
    conexion.commit()
    cursor.close()
    conexion.close()

```

funciones.py

```

from clientes import crear_cliente, leer_clientes, actualizar_cliente,
borrar_cliente
from actividades import crear_actividad, leer_actividades,
actualizar_actividad, borrar_actividad
from entrenadores import crear_entrenador, leer_entrenadores,
actualizar_entrenador, borrar_entrenador
from inscripciones import crear_inscripcion, leer_inscripciones,
borrar_inscripcion

def mostrar_menu():
    print("\n=== Gestión del Centro Deportivo ===")
    print("1. Gestión de Clientes")
    print("2. Gestión de Actividades")
    print("3. Gestión de Entrenadores")
    print("4. Gestión de Inscripciones")
    print("5. Salir")

def menu_clientes():
    print("\n=== Gestión de Clientes ===")
    print("1. Registrar nuevo cliente")
    print("2. Leer clientes existentes")
    print("3. Actualizar un cliente")
    print("4. Eliminar un cliente")
    print("5. Volver")

def menu_actividades():
    print("\n=== Gestión de Actividades ===")
    print("1. Registrar nueva actividad")
    print("2. Leer actividades existentes")
    print("3. Actualizar una actividad")

```

```

print("4. Eliminar una actividad")
print("5. Volver")

def menu_entrenadores():
    print("\n=== Gestión de Entrenadores ===")
    print("1. Registrar nuevo entrenador")
    print("2. Leer entrenadores existentes")
    print("3. Actualizar un entrenador")
    print("4. Eliminar un entrenador")
    print("5. Volver")

def menu_inscripciones():
    print("\n=== Gestión de Inscripciones ===")
    print("1. Registrar nueva inscripción")
    print("2. Leer inscripciones existentes")
    print("3. Eliminar una inscripción")
    print("4. Volver")

def ejecutar_opcion_clientes(opcion):

    if opcion == 1:
        nombre = input("Ingrese el nombre del nuevo cliente: ")
        edad = int(input("Ingrese la edad del nuevo cliente: "))
        tipo_membresia = input("Ingrese el tipo de membresía del nuevo
cliente: ")
        crear_cliente(nombre, edad, tipo_membresia)
        print("Cliente registrado exitosamente.")

    elif opcion == 2:
        clientes = leer_clientes()
        print("\nListado de Clientes:")
        for id_cliente, nombre, edad, tipo_membresia in clientes:
            print(f"ID: {id_cliente}, Nombre: {nombre}, Edad: {edad}
Membresía: {tipo_membresia}")

    elif opcion == 3:
        id_cliente = int(input("Ingrese el ID del cliente a actualizar:
"))
        nuevo_nombre = input("Ingrese el nuevo nombre del cliente: ")
        nueva_edad = input("Ingrese la nueva edad del cliente: ")
        nuevo_tipo_membresia = input("Ingrese el nuevo tipo de membresía
del cliente: ")
        actualizar_cliente(id_cliente, nuevo_nombre, nueva_edad,
nuevo_tipo_membresia)
        print("Cliente actualizado exitosamente.")

    elif opcion == 4:

```

```

        id_cliente = int(input("Ingrese el ID del cliente a eliminar: "))
        borrar_cliente(id_cliente)
        print("Cliente eliminado exitosamente.")

    elif opcion == 5:
        return

    else:
        print("Opción no válida. Intente de nuevo.")

def ejecutar_opcion_actividades(opcion):
    if opcion == 1:
        nombre_actividad = input("Ingrese el nombre de la nueva actividad: ")
        horario = input("Ingrese el horario de la nueva actividad: ")
        duracion = int(input("Ingrese la duración de la nueva actividad: "))

        id_entrenador = int(input("Ingrese el ID del entrenador de la nueva actividad: "))
        crear_actividad(nombre_actividad, horario, duracion, id_entrenador)
        print("Actividad registrada exitosamente.")

    elif opcion == 2:
        actividades = leer_actividades()
        print("\nListado de Actividades:")
        for id_actividad, nombre_actividad, horario, duracion, id_entrenador in actividades:
            print(f"ID: {id_actividad}, Nombre: {nombre_actividad}, Horario: {horario} Duración: {duracion}, Entrenador: {id_entrenador}")

    elif opcion == 3:
        id_actividad = int(input("Ingrese el ID de la actividad a actualizar: "))
        nuevo_nombre = input("Ingrese el nuevo nombre de la actividad: ")
        nuevo_horario = input("Ingrese el nuevo horario de la actividad: ")
        nueva_duracion = input("Ingrese la nueva duración de la actividad: ")
        nuevo_entrenador = input("Ingrese el nuevo entrenador de la actividad: ")
        actualizar_actividad(id_actividad, nuevo_nombre, nuevo_horario, nueva_duracion, nuevo_entrenador)
        print("Actividad actualizada exitosamente.")

    elif opcion == 4:

```

```

        id_actividad = int(input("Ingrese el ID de la actividad a
eliminar: "))
        borrar_actividad(id_actividad)
        print("Actividad eliminada exitosamente.")

    elif opcion == 5:
        return

    else:
        print("Opción no válida. Intente de nuevo.")

def ejecutar_opcion_entrenadores(opcion):
    if opcion == 1:
        nombre = input("Ingrese el nombre del nuevo entrenador: ")
        especialidad = input("Ingrese la especialidad del nuevo
entrenador: ")
        crear_entrenador(nombre, especialidad)
        print("Entrenador registrado exitosamente.")

    elif opcion == 2:
        entrenadores = leer_entrenadores()
        print("\nListado de Entrenadores:")
        for id_entrenador, nombre, especialidad in entrenadores:
            print(f"ID: {id_entrenador}, Nombre: {nombre}, Especialidad:
{especialidad}")

    elif opcion == 3:
        id_entrenador = int(input("Ingrese el ID del entrenador a
actualizar: "))
        nuevo_nombre = input("Ingrese el nuevo nombre del entrenador: ")
        nueva_especialidad = input("Ingrese la nueva especialidad del
entrenador: ")
        actualizar_entrenador(id_entrenador, nuevo_nombre,
nueva_especialidad)
        print("Entrenador actualizado exitosamente.")

    elif opcion == 4:
        id_entrenador = int(input("Ingrese el ID del entrenador a
eliminar: "))
        borrar_entrenador(id_entrenador)
        print("Entrenador eliminado exitosamente.")

    elif opcion == 5:
        return

    else:
        print("Opción no válida. Intente de nuevo.")

```



```

def ejecutar_opcion_inscripciones(opcion):
    if opcion == 1:
        id_cliente = int(input("Ingrese el ID del cliente de la inscripción: "))
        id_actividad = int(input("Ingrese el ID de la actividad de la inscripción: "))
        crear_inscripcion(id_cliente, id_actividad)
        print("Inscripción registrada exitosamente.")

    elif opcion == 2:
        entrenadores = leer_inscripciones()
        print("\nListado de Inscripciones:")

        for id_inscripcion, id_cliente, nombre, edad, tipo_membresia, id_actividad, nombre_actividad, horario, duracion, id_entrenador in entrenadores:
            print(f"ID: {id_inscripcion}, ID cliente: {id_cliente}, Cliente: {nombre}, Edad: {edad}, Membresía: {tipo_membresia}, ID actividad: {id_actividad}, Actividad: {nombre_actividad}, Horario: {horario} Duración: {duracion}, Entrenador: {id_entrenador}")

    elif opcion == 3:
        id_inscripcion = int(input("Ingrese el ID de la inscripción a eliminar: "))
        borrar_inscripcion(id_inscripcion)
        print("Inscripción eliminada exitosamente.")

    elif opcion == 4:
        return

    else:
        print("Opción no válida. Intente de nuevo.")

```

main.py

```

import funciones

def main():
    continuar = True
    while continuar:
        funciones.mostrar_menu() # Mostrar el menú de opciones
        try:
            menu = int(input("Seleccione un menú: "))

```

```

    if menu == 1:
        mismo_menu = True
        while mismo_menu:
            funciones.menu_clientes()
            opcion = int(input("Seleccione una opción: "))
            funciones.ejecutar_opcion_clientes(opcion)
            if opcion == 5:
                mismo_menu = False

    if menu == 2:
        mismo_menu = True
        while mismo_menu:
            funciones.menu_actividades()
            opcion = int(input("Seleccione una opción: "))
            funciones.ejecutar_opcion_actividades(opcion)
            if opcion == 5:
                mismo_menu = False

    if menu == 3:
        mismo_menu = True
        while mismo_menu:
            funciones.menu_entrenadores()
            opcion = int(input("Seleccione una opción: "))
            funciones.ejecutar_opcion_entrenadores(opcion)
            if opcion == 5:
                mismo_menu = False

    if menu == 4:
        mismo_menu = True
        while mismo_menu:
            funciones.menu_inscripciones()
            opcion = int(input("Seleccione una opción: "))
            funciones.ejecutar_opcion_inscripciones(opcion)
            if opcion == 5:
                mismo_menu = False

    if menu == 5:
        break

except ValueError:
    print("Error: Debe ingresar un número válido.")

if __name__ == "__main__":
    main()

```

=== Gestión del Centro Deportivo ===

1. Gestión de Clientes
2. Gestión de Actividades
3. Gestión de Entrenadores
4. Gestión de Inscripciones
5. Salir

Seleccione un menú: 1

=== Gestión de Clientes ===

1. Registrar nuevo cliente
2. Leer clientes existentes
3. Actualizar un cliente
4. Eliminar un cliente
5. Volver

Seleccione una opción: 1

Ingrese el nombre del nuevo cliente: Maria Lopez

Ingrese la edad del nuevo cliente: 43

Ingrese el tipo de membresía del nuevo cliente: Anual

Cliente registrado exitosamente.

=== Gestión de Clientes ===

1. Registrar nuevo cliente
2. Leer clientes existentes
3. Actualizar un cliente
4. Eliminar un cliente
5. Volver

Seleccione una opción: 2

Listado de Clientes:

ID: 2, Nombre: Raul Perez, Edad: 23 Membresía: Mensual

ID: 3, Nombre: Maria Lopez, Edad: 43 Membresía: Anual

=== Gestión de Clientes ===

1. Registrar nuevo cliente
2. Leer clientes existentes
3. Actualizar un cliente
4. Eliminar un cliente
5. Volver

Seleccione una opción: 3

Ingrese el ID del cliente a actualizar: 3

Ingrese el nuevo nombre del cliente: María Antonia Lopez

Ingrese la nueva edad del cliente: 54

Ingrese el nuevo tipo de membresía del cliente: Mensual

Cliente actualizado exitosamente.

=== Gestión de Clientes ===

1. Registrar nuevo cliente
2. Leer clientes existentes
3. Actualizar un cliente
4. Eliminar un cliente
5. Volver

Seleccione una opción: 2

Listado de Clientes:

ID: 2, Nombre: Raul Perez, Edad: 23 Membresía: Mensual

ID: 3, Nombre: María Antonia Lopez, Edad: 54 Membresía: Mensual

=== Gestión de Clientes ===

1. Registrar nuevo cliente
2. Leer clientes existentes
3. Actualizar un cliente
4. Eliminar un cliente
5. Volver

Seleccione una opción: 4

Ingrese el ID del cliente a eliminar: 3

Cliente eliminado exitosamente.

=== Gestión de Clientes ===

1. Registrar nuevo cliente
2. Leer clientes existentes
3. Actualizar un cliente
4. Eliminar un cliente
5. Volver

Seleccione una opción: 2

Listado de Clientes:

ID: 2, Nombre: Raul Perez, Edad: 23 Membresía: Mensual

=== Gestión del Centro Deportivo ===

1. Gestión de Clientes
2. Gestión de Actividades
3. Gestión de Entrenadores
4. Gestión de Inscripciones
5. Salir

Seleccione un menú: 3

=== Gestión de Entrenadores ===

1. Registrar nuevo entrenador
2. Leer entrenadores existentes
3. Actualizar un entrenador
4. Eliminar un entrenador
5. Volver

Seleccione una opción: 1

Ingrese el nombre del nuevo entrenador: Juan Morales

Ingrese la especialidad del nuevo entrenador: pilates

Entrenador registrado exitosamente.

=== Gestión de Entrenadores ===

1. Registrar nuevo entrenador
2. Leer entrenadores existentes
3. Actualizar un entrenador
4. Eliminar un entrenador
5. Volver

Seleccione una opción: 2

Listado de Entrenadores:

ID: 1, Nombre: Juan Morales, Especialidad: pilates

=== Gestión de Entrenadores ===

1. Registrar nuevo entrenador
2. Leer entrenadores existentes
3. Actualizar un entrenador
4. Eliminar un entrenador
5. Volver

Seleccione una opción: 3

Ingrese el ID del entrenador a actualizar: 1

Ingrese el nuevo nombre del entrenador: Juan Ramirez

Ingrese la nueva especialidad del entrenador: pilates

Entrenador actualizado exitosamente.

=== Gestión de Entrenadores ===

1. Registrar nuevo entrenador
2. Leer entrenadores existentes
3. Actualizar un entrenador
4. Eliminar un entrenador
5. Volver

Seleccione una opción: 5

=== Gestión del Centro Deportivo ===

1. Gestión de Clientes
2. Gestión de Actividades
3. Gestión de Entrenadores
4. Gestión de Inscripciones
5. Salir

Seleccione un menú: 2

=== Gestión de Actividades ===

1. Registrar nueva actividad
2. Leer actividades existentes
3. Actualizar una actividad
4. Eliminar una actividad
5. Volver

Seleccione una opción: 1

Ingrese el nombre de la nueva actividad: pilates

Ingrese el horario de la nueva actividad: 15:00-16:00

Ingrese la duración en horas de la nueva actividad: 1

Ingrese el ID del entrenador de la nueva actividad: 1

Actividad registrada exitosamente.

=== Gestión de Actividades ===

1. Registrar nueva actividad
2. Leer actividades existentes
3. Actualizar una actividad
4. Eliminar una actividad
5. Volver

Seleccione una opción: 2

Listado de Actividades:

ID: 2, Nombre: pilates, Horario: 15:00-16:00 Duración: 1, Entrenador: 1

```
=== Gestión de Actividades ===
1. Registrar nueva actividad
2. Leer actividades existentes
3. Actualizar una actividad
4. Eliminar una actividad
5. Volver
Seleccione una opción: 4
Ingrese el ID de la actividad a eliminar: 2
Actividad eliminada exitosamente.
```

```
=== Gestión de Actividades ===
1. Registrar nueva actividad
2. Leer actividades existentes
3. Actualizar una actividad
4. Eliminar una actividad
5. Volver
Seleccione una opción: 5
```

```
=== Gestión del Centro Deportivo ===
1. Gestión de Clientes
2. Gestión de Actividades
3. Gestión de Entrenadores
4. Gestión de Inscripciones
5. Salir
Seleccione un menú: 4
```

```
=== Gestión del Centro Deportivo ===
1. Gestión de Clientes
2. Gestión de Actividades
3. Gestión de Entrenadores
4. Gestión de Inscripciones
5. Salir
Seleccione un menú: 4

=== Gestión de Inscripciones ===
1. Registrar nueva inscripción
2. Leer inscripciones existentes
3. Eliminar una inscripción
4. Volver
Seleccione una opción: 1
Ingrese el ID del cliente de la inscripción: 2
Ingrese el ID de la actividad de la inscripción: 4
Inscripción registrada exitosamente.
```

```
=== Gestión de Inscripciones ===
1. Registrar nueva inscripción
2. Leer inscripciones existentes
3. Eliminar una inscripción
4. Volver
Seleccione una opción: 2

Listado de Inscripciones:
ID: 2, ID cliente: 2, Cliente: Raul Perez, Edad: 23, Membresía: Mensual, ID actividad: 4, Actividad: zumba, Horario: 12:00-14:00 Duración: 2, Entrenador: 1

=== Gestión de Inscripciones ===
1. Registrar nueva inscripción
2. Leer inscripciones existentes
3. Eliminar una inscripción
4. Volver
Seleccione una opción: 3
Ingrese el ID de la inscripción a eliminar: 2
Inscripción eliminada exitosamente.
```

```
=== Gestión de Inscripciones ===
1. Registrar nueva inscripción
2. Leer inscripciones existentes
3. Eliminar una inscripción
4. Volver
Seleccione una opción: 4

=== Gestión del Centro Deportivo ===
1. Gestión de Clientes
2. Gestión de Actividades
3. Gestión de Entrenadores
4. Gestión de Inscripciones
5. Salir
Seleccione un menú: 5
PS C:\Users\CampusFP\Desktop\Programación\Python\Python Ej 19> 
```

<https://github.com/Rodrigo-Garcia-Ortiz/Programacion.git>