

Tema: Clase POO

Objetivo:

Conectar Python con una base de datos SQLite

Insertar, consultar, actualizar y eliminar registros usando un patrón de repositorio

Teniendo en cuenta la base de datos `simple_person.db`, realizar la siguiente implementación en código Python

```
class PersonRepository:

    def add_person(self, name: str, last_name: str, age: int, city: str) -> int:
        """Inserta una persona y retorna su id."""

    def get_person_by_id(self, person_id: int) -> dict:
        """Retorna los datos de una persona por su ID."""

    def get_all_people(self) -> list:
        """Retorna una lista con todas las personas."""

    def update_person(self, person_id: int, name: str, last_name: str, age: int, city: str) -> bool:
        """Actualiza una persona por su ID. Retorna True si fue exitosa."""

    def delete_person(self, person_id: int) -> bool:
        """Elimina una persona por su ID. Retorna True si fue exitosa."""
```

Implementación en Python

```
import sqlite3
```

```
class PersonRepository:
    def __init__(self, db_path='simple_person.db'):
        self.db_path = db_path

    def add_person(self, name: str, last_name: str, age: int, city: str) -> int:
        """Inserta una persona y retorna su id."""
        with sqlite3.connect(self.db_path) as conn:
            cursor = conn.cursor()
            cursor.execute(
                "INSERT INTO person (name, last_name, age, city) VALUES (?, ?, ?, ?)",
                (name, last_name, age, city)
            )
            conn.commit()
            return cursor.lastrowid

    def get_person_by_id(self, person_id: int) -> dict:
        """Retorna los datos de una persona por su ID."""
        with sqlite3.connect(self.db_path) as conn:
```

```

    cursor = conn.cursor()
    cursor.execute("SELECT * FROM person WHERE id = ?", (person_id,))
    row = cursor.fetchone()
    if row:
        return {"id": row[0], "name": row[1], "last_name": row[2], "age": row[3], "city": row[4]}
    return {}

```

def get_all_people(self) -> list:

```

    """Retorna una lista con todas las personas."""
    with sqlite3.connect(self.db_path) as conn:
        cursor = conn.cursor()
        cursor.execute("SELECT * FROM person")
        rows = cursor.fetchall()
        return [{"id": row[0], "name": row[1], "last_name": row[2], "age": row[3], "city": row[4]} for row in
rows]

```

def update_person(self, person_id: int, name: str, last_name: str, age: int, city: str) -> bool:

```

    """Actualiza una persona por su ID. Retorna True si fue exitosa."""
    with sqlite3.connect(self.db_path) as conn:
        cursor = conn.cursor()
        cursor.execute(
            "UPDATE person SET name = ?, last_name = ?, age = ?, city = ? WHERE id = ?",
            (name, last_name, age, city, person_id)
        )
        conn.commit()
        return cursor.rowcount > 0

```

def delete_person(self, person_id: int) -> bool:

```

    """Elimina una persona por su ID. Retorna True si fue exitosa."""
    with sqlite3.connect(self.db_path) as conn:
        cursor = conn.cursor()
        cursor.execute("DELETE FROM person WHERE id = ?", (person_id,))
        conn.commit()
        return cursor.rowcount > 0

```

este código da respuesta a la implementación.