TECNICATURA SUPERIOR EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

Evidencia de Aprendizaje N° 2

1. Modelo de clases:

Vamos a diseñar un **diagrama sencillo (UML)** de la clase

|  |
| --- |
| **Clase Contacto** |
| Id:  Nombre  Apellido  Telefono  Email |
| Agregar  Eliminar  Actualizar  Listar |

|  |
| --- |
| ConexionBase |
| Conexion  Cursor |
| Conectar  Ejecutar  cerrar |

Relación: La clase Contacto usará métodos de ConexionBase para guardar, modificar y eliminar registros.

1. Modelo de datos

Este es el modelo de la tabla que se creara en SQLite: los atributos de la tabla son los atributos de la clase.

|  |  |  |
| --- | --- | --- |
| Campo | Tipo de dato | Restricciones |
| id | INTEGER | PRIMARY KEY AUTOINCREMENT |
| nombre | TEXT | NOT NULL |
| apellido | TEXT | NOT NULL |
| telefono | TEXT | NOT NULL |
| email | TEXT | NOT NULL |

3. Investigación teórica:

○ Explicación de los conceptos de POO que aplicarán (clases, objetos, atributos,

métodos).

Los conceptos de POO que estamos utilizando son:

**Clase**: es la plantilla que define la estructura de un conjunto de objetos con características comunes. En este caso, la clase “Contacto” se utiliza para crear objetos que tienen las mismas características.

**Objeto**: es una instancia de una clase que representa una entidad concreta del mundo real (ej: un contacto con nombre, teléfono y email).

**Atributos**: son las características o propiedades que describen un objeto. Son las variables dentro de la clase (ej: nombre, apellido).

**Métodos**: define el comportamiento o la acción que un objeto de una clase puede realizar. Son las funciones dentro de la clase que definen comportamientos (ej: Agregar, Eliminar).

○ Definición de DDL, DML y DCL con ejemplos de las sentencias que usarán en

el proyecto (CREATE TABLE, INSERT, SELECT, UPDATE, DELETE, etc.).

* **DDL (Data Definition Language)** → define la estructura.

Es el conjunto de comandos que se utiliza para crear, modificar y eliminar objetos dentro de una base de datos.

Por ejemplo, para crear una tabla:

CREATE TABLE Contactos (

id INTEGER PRIMARY KEY AUTOINCREMENT,

nombre TEXT NOT NULL,

apellido TEXT NOT NULL,

telefono TEXT,

email TEXT

);

**DML (Data Manipulation Language)** → manipula los datos.

Son comandos que usan para consultar, insertar y actualizar y eliminar datos de las tablas de una base de datos.

INSERT INTO Contactos (nombre, apellido, telefono, email)

VALUES ('Juan', 'Pérez', '351111111', 'juan@mail.com');

#el id es autoincremental, por eso no se anota como valor en INTO

UPDATE Contactos SET telefono='351222222' WHERE id=1;

DELETE FROM Contactos WHERE id=1;

**DCL (Data Control Language)** → permisos y control

Es el conjunto de comando que se utiliza para gestionar los permisos y derechos de acceso de los usuarios dentro de una base de datos.

GRANT SELECT ON Contactos TO usuario;

○ Descripción del proceso de conexión a SQLite en Python

Paso 1: importar el modulo sqlite3

import sqlite3

Paso 2: conectarse a la base de datos utilizando sqlite3.connect() que recibe como argumento el nombre de la base de datos

conexion = sqlite3.connect(‘Libreta\_contactos.bd’)

Paso 3: crear un objeto cursor, que permitirá interactuar con la base de datos.

cursor = conexión.cursor()

Paso 4: Insertar un contacto de prueba

cursor.execute("""

INSERT INTO Contactos (nombre, apellido, telefono, email)

VALUES (?, ?, ?, ?)

""", ("Juan", "Pérez", "351111111", "juan@mail.com"))

#Guardar cambios

conn.commit()

Consultar todos los contactos

cursor.execute("SELECT \* FROM Contactos")

print(cursor.fetchall())

Cerrar conexión

conn.close()

#Método conectar: abre la conexión con la base (.connect()) y crea un **cursor**,

#que es como un “puntero” para ejecutar sentencias SQL (.cursor()).

#Método ejecutar:

#Recibe una sentencia SQL (sql) y opcionalmente parámetros (params).

#Usa cursor.execute(sql, params) para ejecutarla.

#Llama a commit() para **guardar los cambios** en la base (ejemplo: INSERT, UPDATE, DELETE).

#Método listar:

#Es un **método estático** (no necesita crear un objeto Contacto para usarlo → se llama directo: Contacto.listar()).

#Ejecuta un **SELECT** para obtener todos los contactos de la base.

#Devuelve una lista con tuplas.