

**Algoritmos Y Estructuras De
Datos I:
TinySQLDb**

Integrantes:

Morera Valverde Deiler, carné: 2023115868 100 %

Profesor: Leonardo Andres Araya Martinez

II Semestre 2024

Índice

| | |
|---|----------|
| 1. Introducción | 3 |
| 2. Descripción del problema | 3 |
| 3. Descripción de la solución | 3 |
| 3.1. Cliente PowerShell | 3 |
| 3.2. API Interface | 3 |
| 3.3. Stored Data Manager | 4 |
| 4. Requerimientos implementados | 4 |
| 5. Diseño general | 4 |
| 6. Casos de prueba | 5 |
| 6.1. Operación exitosa | 5 |
| 6.2. Error por índice duplicado | 6 |
| 7. Conclusiones | 6 |
| 8. Bibliografía | 7 |

1. Introducción

El objetivo de este proyecto es diseñar e implementar un motor de bases de datos relacional sencillo, TinySQLDb, que permita a los usuarios ejecutar sentencias SQL a través de una interfaz cliente-servidor. Este proyecto busca proporcionar una base sólida para entender cómo funcionan los motores de bases de datos relacionales.

2. Descripción del problema

El problema consiste en la creación de un sistema administrador de bases de datos que pueda recibir consultas en SQL, validarlas y ejecutarlas utilizando archivos para almacenar las tablas y los datos. El sistema está compuesto por tres componentes principales: el cliente, la API Interface, y el Stored Data Manager.

3. Descripción de la solución

3.1. Cliente PowerShell

El cliente se ha implementado como un módulo de PowerShell, permitiendo a los usuarios ejecutar consultas SQL desde la línea de comandos. El cliente envía las consultas al servidor mediante sockets, y recibe los resultados en formato tabla.

3.2. API Interface

La API Interface maneja la comunicación entre el cliente y el servidor utilizando sockets. Se encarga de recibir las consultas SQL del cliente, procesarlas y enviar los resultados de vuelta.

3.3. Stored Data Manager

El Stored Data Manager es responsable de acceder a los archivos del sistema donde se almacenan las tablas y datos, permitiendo la ejecución de las consultas sobre ellos.

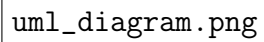
4. Requerimientos implementados

A continuación se describen algunos de los requerimientos implementados:

- **Cliente PowerShell:** El cliente permite ejecutar scripts SQL y ver los resultados en formato tabla.
- **Crear base de datos:** La sentencia `CREATE DATABASE` crea una nueva base de datos como una carpeta en el sistema de archivos.
- **Crear tablas:** Se utiliza la sentencia `CREATE TABLE` para definir nuevas tablas dentro de una base de datos.

5. Diseño general

El diseño de TinySQLDb sigue un enfoque orientado a objetos. A continuación, se muestra el diagrama de clases del sistema, que refleja la relación entre los componentes principales.



uml_diagram.png

Figura 1 Diagrama de clases del sistema TinySQLDb

6. Casos de prueba

6.1. Operación exitosa

```
CREATE DATABASE Universidad ;  
SET DATABASE Universidad ;  
CREATE TABLE Estudiantes (
```

```
ID INTEGER,
Nombre VARCHAR(30),
FechaNacimiento DATETIME
);
INSERT INTO Estudiantes (1, "Juan", "2000-01-01 01:02:00");
SELECT * FROM Estudiantes;
```

6.2. Error por índice duplicado

```
INSERT INTO Estudiantes (1, "Juan", "2000-01-01 01:02:00");
// Error: ID duplicado
```

7. Conclusiones

Este proyecto permite explorar el funcionamiento básico de un motor de bases de datos relacional, abordando aspectos como la ejecución de consultas SQL, la creación de índices y el almacenamiento de datos en archivos.

8. Bibliografía