

BonsaiDB :: Manual de Usuario

John Fajardo

Versión: 1.1

Fecha: 11 de julio de 2025

1. Introducción

¡Bienvenido a BonsaiDB!

BonsaiDB es un motor de base de datos ligero de alto rendimiento, escrito en C++17. Este proyecto ha sido diseñado como una demostración práctica de conceptos avanzados de backend, incluyendo estructuras de datos eficientes, gestión de almacenamiento en disco y programación concurrente.

Este manual te guiará a través del proceso de compilación, instalación y uso de BonsaiDB mediante su Interfaz de Línea de Comandos (CLI), además de mostrarte cómo ejecutar las pruebas y benchmarks incluidos.

2. Compilación e Instalación

Para compilar BonsaiDB desde la fuente, necesitarás un compilador de C++ compatible con el estándar C++17 y CMake.

2.1. Requisitos

- **CMake:** Versión 3.10 o superior.
- **Compilador C++17:** GCC 7+, Clang 5+ o MSVC 2017+.

2.2. Pasos de Compilación

1. Abre una terminal y clona el repositorio del proyecto desde GitHub.
2. Navega a la raíz del proyecto y crea un directorio de compilación (*out-of-source build*) para mantener el código fuente limpio:

```
mkdir build
cd build
```

3. Ejecuta CMake para generar los archivos de compilación y, a continuación, compila el proyecto:

```
cmake ..  
make -j$(nproc)
```

Nota: `$(nproc)` es un comando de Linux que devuelve el número de núcleos de procesamiento disponibles, permitiendo una compilación en paralelo más rápida. En otros sistemas puedes reemplazarlo por un número, por ejemplo, `make -j4`.

4. Una vez finalizado, encontrarás el ejecutable principal **bonsaidb**, junto con los ejecutables de las pruebas, dentro del directorio **build**.

3. Uso de la Interfaz de Línea de Comandos (CLI)

BonsaiDB se gestiona a través de una aplicación de línea de comandos interactiva.

3.1. Iniciar el Motor

Para iniciar la CLI, ejecuta el programa **bonsaidb** desde el directorio **build**, especificando el archivo de base de datos que deseas crear o utilizar:

```
./bonsaidb mi_base_de_datos.db
```

Una vez dentro, verás el indicador **bonsaidb>**.

3.2. Comandos Disponibles

Comando	Sintaxis	Descripción
insert	insert <id> <nombre> <edad> <saldo>	Inserta un nuevo registro en la base de datos. El nombre no debe contener espacios.
select	select <id>	Busca y muestra un registro específico por su id .
delete	delete <id>	Elimina un registro de la base de datos usando su id .
dump	dump	Muestra todos los registros almacenados en la base de datos en formato CSV.
help	help	Muestra este mismo mensaje de ayuda con la lista de comandos.
exit	exit	Cierra la sesión interactiva y finaliza el programa.

4. Pruebas y Benchmarks

El proyecto incluye un conjunto de pruebas para verificar la correctitud del motor y benchmarks para medir su rendimiento.

4.1. Ejecutar las Pruebas

Después de compilar el proyecto, los ejecutables de las pruebas estarán disponibles en el directorio `build/test`. Puedes ejecutarlos individualmente para verificar el funcionamiento de cada componente. Por ejemplo:

```
./test/test_insert  
./test/test_query  
./test/test_concurrency
```

4.2. Ejecutar los Benchmarks

El repositorio incluye un script para compilar y ejecutar las pruebas de estrés de forma automática. Desde la raíz del proyecto, ejecuta:

```
./benchmarks/ejecutar_benchmarks.sh
```

Este script compilará el proyecto en un directorio temporal (`build_bench`) y guardará los resultados del benchmark en `benchmarks/resultados/bench.log`.