

# MyDataBaseCE

\*Proyecto número 3 del curso Algoritmos y Estructuras de Datos II

1<sup>st</sup> Sebastián Hernández Bonilla

Área académica de Ingeniería en Computadores  
Instituto Tecnológico de Costa Rica  
Cartago, Costa Rica  
sebhernandez@estudiantec.cr

2<sup>nd</sup> Daniel Duarte Cordero

Área académica de Ingeniería en Computadores  
Instituto Tecnológico de Costa Rica  
Cartago, Costa Rica  
daduarte@estudiantec.cr

3<sup>rd</sup> Esteban Secaida Vargas

Área académica de Ingeniería en Computadores  
Instituto Tecnológico de Costa Rica  
Cartago, Costa Rica  
esecaida@estudiantec.cr

**Abstract**—This document is a detailed description of everything done in the third project of the Algorithm and Data Structures II course. In this the following points will be detailed: Brief description of the project, code class diagram, architecture diagram, description of the data structures developed and problems encountered during the project.

**Index Terms**—Java, algoritmos, Huffman, BackEnd, FrontEnd, Arduino, Compresión

## I. DESCRIPCIÓN DEL PROBLEMA

MyDataBaseCE corresponde a una aplicación web, la cual debe contar con un aplicación cliente, accesada a través de un navegador web, y un aplicación servidor que es la que tiene toda la lógica del motor de base de datos. La interfaz debe ser similar a una herramienta de bases de datos, como DBVisualizer, DataGrip, MySQL. Es un motor de bases de datos noSQL que deberá realizar todas las operaciones básicas de una base de datos SQL. Se debe implementar un controlador en Arduino para poder ingresar la contraseña de cada usuario e indicar ciertos mensajes por medio de una bocina y dos leds.

## II. DIAGRAMA DE ARQUITECTURA

### III. DIAGRAMA DE CLASES

### IV. DESCRIPCIÓN DE LAS ESTRUCTURAS DE DATOS DESARROLLADAS

Para este proyecto se implementaron varias estructuras de datos que permitieron el desarrollo del mismo. El código BackEnd está implementado en Java y fue programado orientado a objetos. Cuenta con distintas clases y estructuras básicas que cumplen diferentes funciones.

#### A. Huffman Compression

La clase HuffmanEncoder recibe un String que es el que va a procesar para las funciones compress() y decompress(). Primero con la función getCharacterFrequency() obtiene la cantidad de veces que se repiten los chars a lo largo de todo el String y crea un HashMap que luego utiliza para la función compress. Con ese HashMap luego crea el root

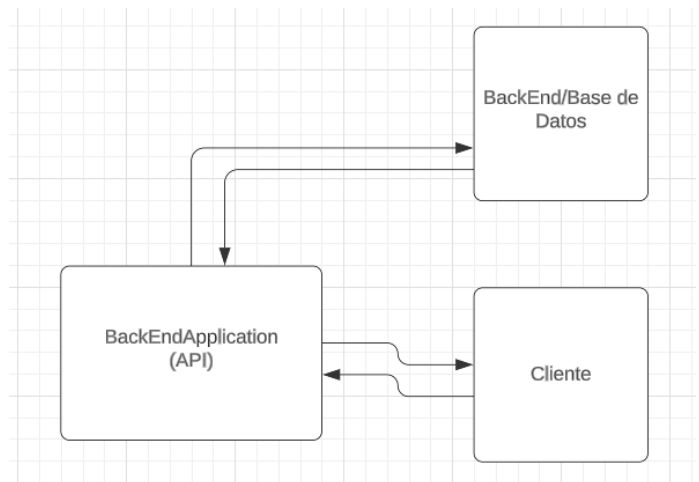


Fig. 1. Diagrama de Arquitectura

del árbol de Huffman para el String comprimido. Con los nodos creados pasa a la función buildHuffmanTree() que se encarga de crear el árbol y generateHuffmanCodesRecursive() que genera el String comprimido de bits. Por último, la función decompress() recibe igualmente un String pero este es el comprimido. Esta va haciendo comparaciones en el árbol hasta generar de nuevo la palabra inicialmente procesada.

## V. CONTROLADOR

El controlador está creado con un Arduino que se encarga de enviar la contraseña en código Morse por medio de dos botones. Por medio de un buzzer emite un sonido cuando la autenticación de un usuario sea exitosa enciende un led cuando la autenticación de un usuario no sea exitosa y enciende un led cuando al hacer un DELETE el mismo implique más de un objeto de un XML store.

## VI. PROBLEMAS ENCONTRADOS

Como problemas encontrados se tiene:

- 1) Dependiendo de como se estripen los botones en el Arduino hay veces que suele interpretar el mensaje diferente al que uno le está mandando.

#### VII. LINKS IMPORTANTES

<https://github.com/EstDan09/myDataBaseCE-frontend>

<https://drive.google.com/file/d/15dhnWeHPYFwIErMX4PiSFdop6s8n88NW/view>

#### REFERENCES

- [1] GeeksforGeeks. (s.f). Huffman Coding — Greedy Algo-3.  
<https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/>