

MANUAL TÉCNICO

Información del sistema:

- OS: Ubuntu 22.04.2 LTS
- OS Type: 64-bit
- CPU: Intel core i5 2.50 GHz
- GPU: Mesa Intel HD Graphics 4400
- Compilador de C++ compatible con C++11 o superior
- RAM: 8 GB
- IDE: IntelliJ idea
- Control de versiones: git 2.34.1
- Github: <https://github.com/Hatsune02/practice2-EDD.git>

Descripción del proyecto:

Consiste en la implementación de HashTables y arboles AVL para poder manejar y tener una gestión de contactos a los cuales se les puede cualquier campo de valor, desde solo un nombre y apellido, hasta mas valores que requiera necesario el usuario al momento de guardarlo. Todo esto usando estructuras propias en el lenguaje C++ y con un MakeFile para la ejecución de este.

Características Principales:

Uso de tablas Hash para almacenamiento de grupos de contactos y de valores de contactos.

Uso de árbol AVL para guardar todos los datos de los contactos.

Uso de graphviz para poder ver de manera grafica como está la estructura total del sistema.

Uso de un pequeño lenguaje para poder insertar, y buscar valores en el sistema.

Introducción

El Sistema de Gestión de Contactos desarrollado para Code&Bugs es una aplicación diseñada para almacenar, gestionar y manipular información de contactos de manera eficiente y segura. Este manual proporciona una descripción detallada de la arquitectura, componentes y funcionalidades del sistema, así como los requisitos necesarios para su correcto funcionamiento.

Arquitectura del Sistema

El sistema se basa en una arquitectura modular que consta de los siguientes componentes principales:

Línea de Comandos: Permite al usuario interactuar con el sistema ingresando comandos para ejecutar diversas acciones.

Gestor de Grupos: Administra la creación, modificación y eliminación de grupos de contactos, así como la definición de campos personalizados para cada grupo.

Gestor de Contactos: Encargado de la inserción, búsqueda y manipulación de contactos dentro de los grupos definidos.

Motor de Búsqueda: Implementa un algoritmo eficiente para la búsqueda de contactos en función de los criterios especificados.

Generador de Gráficos: Utiliza Graphviz para generar gráficos que representan visualmente la estructura de almacenamiento de los contactos.

Exportador de Datos: Permite exportar los datos de los grupos en archivos de texto según el formato especificado.

Gestor de Memoria: Controla el almacenamiento de datos en memoria, incluyendo la gestión de la tabla hash y los árboles AVL.

Almacenamiento de Datos

El sistema utiliza una estructura de datos compuesta por una tabla hash y árboles AVL para almacenar los contactos de manera eficiente y ordenada. Cada grupo de contactos tiene un nombre único y puede definir campos personalizados con tipos de datos como enteros, cadenas, caracteres y fechas.

Estructuras

Tabla Hash: Una Tabla Hash es una estructura de datos que permite almacenar pares clave-valor, donde cada clave es única y se utiliza para acceder al valor asociado de forma rápida. El funcionamiento básico de una Tabla Hash se basa en el uso de una función hash que transforma la clave en un índice en el arreglo subyacente donde se almacenará el valor.

Almacenamiento Inicial: Una Tabla Hash tiene una capacidad inicial que determina el tamaño del arreglo subyacente. Al principio, este arreglo está vacío.

Función Hash: Cuando se agrega un par clave-valor a la Tabla Hash, se aplica una función hash a la clave para determinar el índice donde se almacenará el valor en el arreglo.

Crecimiento Dinámico: A medida que se agregan elementos a la Tabla Hash, si la carga (número de elementos / capacidad total) supera un umbral predefinido, la Tabla Hash se redimensiona automáticamente para aumentar su capacidad y evitar colisiones excesivas.

Búsqueda: Para buscar un valor en una Tabla Hash, se aplica la función hash a la clave y se busca en el índice correspondiente.

Árbol AVL: Un árbol AVL es un tipo de árbol binario de búsqueda balanceado en el que la diferencia de altura entre los subárboles izquierdo y derecho de cada nodo (llamada factor de equilibrio) está limitada a $\{-1, 0, 1\}$. El funcionamiento básico de un árbol AVL se centra en mantener esta propiedad de equilibrio después de cada operación de inserción o eliminación.

Inserción: Al insertar un nuevo nodo en un árbol AVL, se realiza primero una inserción estándar de un árbol binario de búsqueda. Luego, se actualizan los factores de equilibrio de los nodos en el camino desde el nodo recién insertado hasta la raíz, y se realizan rotaciones simples o dobles según sea necesario para restaurar el equilibrio.

Balanceo Automático: El árbol AVL se mantiene automáticamente balanceado después de cada inserción o eliminación, lo que garantiza que la altura del árbol sea $O(\log n)$, donde n es el número de nodos en el árbol.

Eficiencia: La búsqueda, inserción y eliminación en un árbol AVL tienen un rendimiento de $O(\log n)$, ya que la altura del árbol está limitada por la propiedad de equilibrio.

En el caso del proyecto se utilizaron dos Tablas Hash, una en la cual tenga como valor otra tabla Hash, y esta segunda tabla Hash tiene como valor un árbol AVL. En la parte del árbol AVL cada nodo tiene una llave para que se pueda ir insertando el árbol y un campo donde se pueda ingresar el valor, ya sea un nombre, apellido o número de teléfono.

Lenguaje de Comunicación

El sistema cuenta con una línea de comandos que simula una terminal de comandos, donde el usuario puede ingresar comandos para realizar diversas acciones, como la creación de grupos, inserción de contactos, búsqueda y generación de gráficos.

Funcionalidades Principales

El Sistema de Gestión de Contactos ofrece las siguientes funcionalidades principales:

Creación y gestión de grupos de contactos con campos personalizados.

Inserción y búsqueda eficiente de contactos.

Generación de gráficos que representan la estructura de almacenamiento.

Exportación de datos de contactos en archivos de texto.

Gestión automática de re-hash cuando la tabla hash supera el Factor de Carga en un 60%.