

Universidad de San Carlos de Guatemala

Facultad de Ingenieria

Escuela de Ciencias y Sistemas

Estructuras de Datos Seccion N

Ingeniero Luis Fernando Espino Barrios

Auxiliar Susel Retana

Guatemala, 2 de Julio de 2020



Manual Tecnico

Nombre: Byron Antonio Orellana Alburez
Carnet: 201700733

Nombre Jackeline Alexandra Benitez Benitez
Carnet 201709166

Introducción

La empresa Llega Rapidito, durante varios años ha prestado su servicio de transporte con cobertura en todo el país.

Los directores de la compañía han decidido realizar un cambio de sistema para el control de vehículos, conductores, clientes y viajes, ya que el sistema actual cuenta con varios fallos y últimamente algunos procesos trabajan de forma lenta, esto debido a la gran cantidad de registros que maneja la empresa.

Por lo que se le ha contratado a usted para el desarrollo de una nueva solución, esta contará con una aplicación que estará implementada con estructuras de datos en memoria, esto con el propósito de agilizar los procesos utilizados para el control de la información.

Por lo tanto es importante conocer sobre diversas estructuras de datos, tales como Matriz de Dispersión o Tabla Hash, el árbol B, Todos los tipos de Listas, colas, pilas y grafos. Ya que nos permiten tener un mejor entendimiento sobre la información que se maneja y como almacenarla.

Objetivos y Alcances del Sistema:

Objetivos:

- Realizar un correcto registro de todas las entidades.
- Realizar una correcta visualización de todas las estructuras incluyendo la estructura general.
- Hacer un correcto proceso de Registro de viajes por medio de Blockchain.

Alcances de Sistema:

- Realizar un registro de Viajes a través de Blockchain.
- Determinar la cantidad mas corta de los viajes.
- Visualizar los tops de x entidad

Especificaciones Técnicas

Lenguaje de Programación: Java

Github:

https://github.com/ByLy23/EDD_Jun_2020_201700733_P2_201709166.git

Sistema Operativo: Sistema Operativo Windows 10.

IDE: Apache NetBeans IDE 11.2

Estructuras de Datos Utilizadas

Árbol B:

En las ciencias de la computación, los árboles-B o B-árboles son estructuras de datos de árbol que se encuentran comúnmente en las implementaciones de bases de datos y sistemas de archivos. Al igual que los árboles binarios de búsqueda, son árboles balanceados de búsqueda, pero cada nodo puede poseer más de dos hijos.¹ Los árboles B mantienen los datos ordenados y las inserciones y eliminaciones se realizan en tiempo logarítmico amortizado.

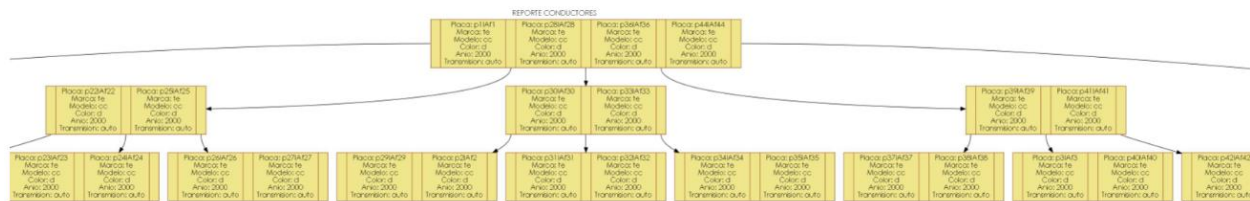
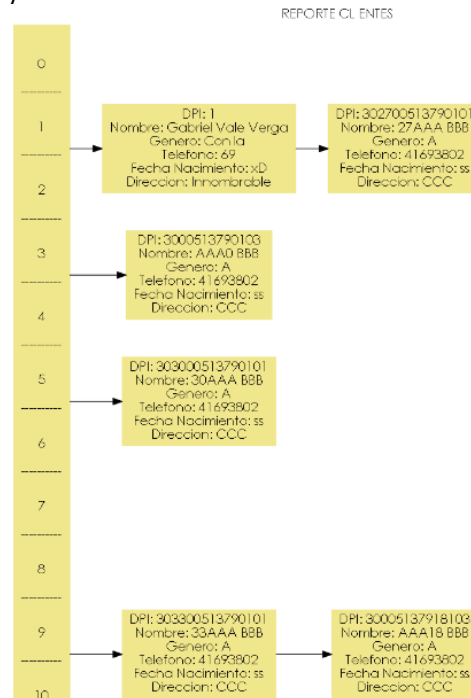


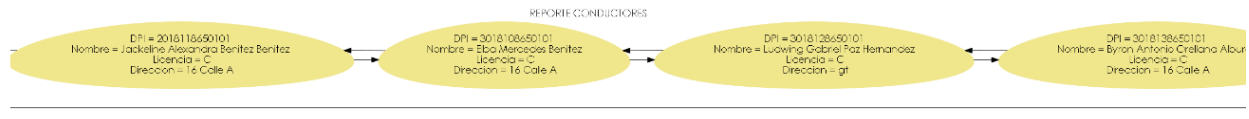
Tabla Hash:

Una tabla hash, matriz asociativa, hashing, mapa hash, tabla de dispersión o tabla fragmentada es una estructura de datos que asocia llaves o claves con valores. La operación principal que soporta de manera eficiente es la búsqueda: permite el acceso a los elementos (teléfono y dirección, por ejemplo) almacenados a partir de una clave generada (usando el nombre o número de cuenta, por ejemplo). Funciona transformando la clave con una función hash en un hash, un número que identifica la posición (casilla o cubeta) donde la tabla hash localiza el valor deseado.¹



Lista Doblemente Enlazada Circular

En una lista enlazada doblemente circular, cada nodo tiene dos enlaces, similares a los de la lista doblemente enlazada, excepto que el enlace anterior del primer nodo apunta al último y el enlace siguiente del último nodo, apunta al primero.



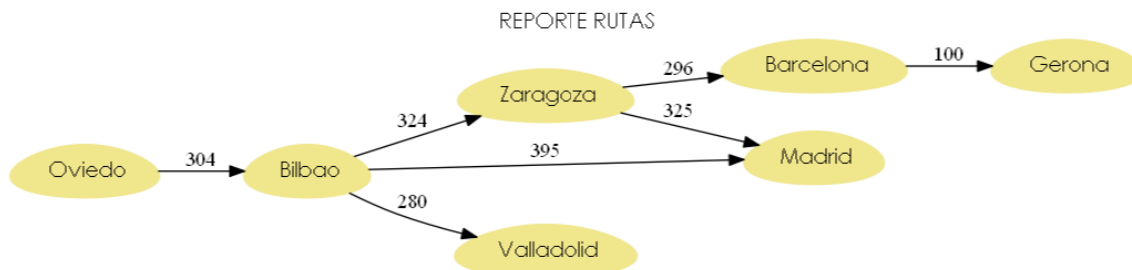
Lista Simple

es una estructura de datos, caracterizada por ser una secuencia de elementos en la que la operación de inserción push se realiza por un extremo y la operación de extracción pull por el otro.



Grafo Dirigido:

Un grafo dirigido (o digrafo) G consiste de un conjunto de vértices y un conjunto de arcos E . A los vértices se les llama también nodos o puntos y a los arcos aristas dirigidas o líneas dirigidas. Un arco es un par ordenado de vértices (v, w) donde v es la cola y w la cabeza del arco. Un arco (v, w) se expresa también como $v \rightarrow w$ y se dibuja como.



Metodos y Clases

- **GettersYSetters:** Para obtener y settear los datos que sean necesarios en cualquier clase
- **Grafo:** Para hacer la estructura del grafo.
- **Tabla Hash:** Para establecer la estructura de la Tabla Hash.
- **Insertar:** Para Insertar cualquier dato
- **Manejador:** clase que contiene la lista de Usuarios en el sistema y maneja el usuario que inicia sesión.
- **Arbol B:** clase que contiene los atributos del árbol b y toda su estructura.
- **List:** clase que permite el manejo de Listas para el almacenamiento de usuarios.
- **agregarUsuario(Usuario):** método que maneja el metodo nativo de la clase Usuario que agrega a la lista.
- **buscarUsuario(String):** método que devuelve un usuario en específico, dependiendo del parámetro recibido.
- **agregarConductor(Conductor):** método que maneja el metodo nativo de la clase Conductor que agrega a la lista.
- **buscarConductor(String):** método que devuelve un usuario en específico, dependiendo del parámetro recibido.
- **agregarVehiculo(Vehiculo):** método que maneja el metodo nativo de la clase Vehiculo que agrega a la lista.
- **buscarVehiculo (Vehiculo):** método que devuelve un usuario en específico, dependiendo del parámetro recibido.
- **ActionListener():** Permite tener llamar una accion
- **ActionPerformed():** Hace la accion
- **getContentPane():** Para añadirlo a Panel
- **SetBounds:** Cambiar tamaño y coordenadas
- **SetSize:** Cambiar Tamaño
- **SetVisible:** Para Ocultar o Aparecer

- **JOptionPane:** para mensajes pequeños
- **Button, Label, Combos, JtextField:** para implementar en el panel
- **Jframe:** Ventana que se usa
- **DefaultTableModel:** Para crear tablas
- **Jpanel:** para crear nuevas laminas o espacios en el frame
- **Defaultsetoperation:** para cerrarlo automaticamente
- **Blockchain:** Clase que maneja la mayoría del funcionamiento de todo
- **Graficar:** Grafica todo lo que contiene la clase
- **Array:** Para guardar los objetos creados
- **If y Switch:** para el proceso de ocultar y aparecer los labels en la carga
- **Jtable:** Para crear la tabla
- **Toolkit:** para obtener tamaño de la pantalla
- **AddItem:** para añadir items al combo
- **Get y set:** para obtener los campos de un jtext field
- **Hide:** para ocultar un form
- **Show:** método para mostrar un objeto.
- **FileReader:** para leer el archivo.
- **DataGridView:** objeto que sirve para mostrar

Dudas o Consultas

- **Byron Orellana**
Software Developer byly_32@gmail.com
Telefono 58986369
- **Jackeline Benitez**
Software Developer jackelinebenitez112@gmail.com
Telefono: 56919369