1. **Identificación del Problema**

El Problema: El desconocimiento general de los peajes dentro del territorio colombiano

Identificación de las Necesidades

- Se requiere manipular información en archivos de los peajes dentro del territorio Colombiano
- Se requiere marcar en el mapa la ubicación de los peajes
- Se requiere poder visualizar cierta información de los peajes.
- Se requiere almacenar la información manipulada en algún tipo de estructura de datos.

2. **Recopilación de Información**

Marco Teórico

Estructura de Datos

Una estructura de datos se puede definir como un conjunto de datos que se caracterizan por su organización y sus características similares. Los datos contenidos dentro de la estructura, son llamados datos estructurados, lo cual consiste en que estos se encuentran almacenados dentro de un único indicador. Las estructuras de datos se pueden clasificar en cuatro tipos:

- Contiguas: En este caso como lo indica el nombre son estructuras que almacenan los datos de manera contigua dentro de la memoria del computador. Esto permite encontrar otros datos mediante la posición relativa de uno a la de los demás.
- Enlazadas: Son estructuras de datos que no necesariamente requieren almacenar los datos en una misma ubicación, sino que se utilizan punteros para dirigir de un dato a otro de manera directa.

- Estáticas: Las estructuras de datos que tienen un tamaño fijo y determinado con anterioridad
- Dinámicas: Las estructuras de datos que no tienen restricciones de almacenamiento, sino que dependen de la capacidad de memoria del computador.

Tomado y resumido de ***Fundamentos de Informática y Programación, Capítulo 5***

Cola de Prioridad

La cola de Prioridad es una estructura de datos en la cual se priorizan cierto tipos de datos dependiendo del valor del identificador dado. En este caso existen dos tipos de colas de prioridad. La primera siendo máxima que solo permite extraer de la estructura el dato con la llave de mayor valor.Por otro lado, la segunda también conocida como mínima prioridad se enfoca en tomar los datos con la llave de menor valor.

Tomado, traducido y resumido de ***Introduction To Algorithms, Capítulo 6***

Arbol Binario

EL árbol binario es una estructura que consiste de una agrupación de nodos en el cual el primero se le conoce como la raíz y de este pueden salir múltiples nodos, en este caso serian dos ya que binario hace referencia a la cantidad de nodos que salen de otro nodo en un nivel superior. Cada nodo contiene un objeto en su interior y es diferenciado mediante una llave que contiene cada nodo. Esta llave sirve como indicador para ordenar, encontrar o eliminar un objeto de información si este se encuentra dentro del árbol.

Tomado, traducido y resumido de ***Introduction To Algorithms, Capitulo 12***

Diccionarios

Es una estructura de datos la cual agrupa objetos con similitudes bajo una misma clave. Esta clave puede ser una cadena de texto o un valor, siempre y cuando sea posible compararlo con otros valores y deben de estar ordenadas.

Tomado de Diccionarios.

Lista de C#

La lista de este lenguaje proviene de la biblioteca de estructuras genéricas del mismo lenguaje de programación. Esta lista es el reemplazo del ArrayList en otros lenguajes de programación y tiene incluido varios métodos que permiten manipular la información con mayor facilidad. Permite un manejo de más de dos millones de elementos y tiende a funcionar mejor que la clase ArrayList en la mayoría de los casos.

Tomado, traducido y resumido de List<T> Class .

Estado Del Arte

Toll Guru: Es una aplicación que permite calcular precios de viajes teniendo en cuenta los costos de gasolina y peajes. Solo está disponible para Canadá, Estados Unidos y México.

Google Maps: Es un servicio de mapas web, ofrece muchas funcionalidades, condiciones de tráfico en tiempo real, mapas de calles, planeador de rutas, etc. Sin embargo, sólo tiene información acerca de los lugares donde se encuentran las estaciones de peaje y nada más.

3. Búsqueda de Soluciones Creativas

Manipulación de Datos

Como el problema remarca se van a tener cantidades de datos considerables por lo que se deberán usar estructuras de datos para trabajar con ellos de manera eficiente.

Las siguientes son las posibles estructuras de datos:

- Árbol binario de búsqueda
- Colas de prioridad
- Diccionarios
- Listas

Ideas Creativas

1. Crear un árbol binario en el cual se almacenen toda la información del archivo de peajes.
2. Crear un árbol binario que solo cargue del departamento especificado
3. Crear una cola donde se almacenen listas de los peajes de cada departamento
4. Crear un conjunto de diccionarios por lo cuales se pueden agrupar los peajes por precio o departamento
5. Crear una lista con todos los peajes del país
6. Crear un árbol que solo cargue los peajes de un departamento específico
7. Crear una lista de listas que contengan los peajes por departamento
8. Crea una lista de colas de prioridad de los peajes por departamento

4. Transición de la formulación de ideas a los diseños preliminares

Debido a que la aplicación está dedicada a dar a conocer el valor de los peajes a las personas que van a viajar en Colombia y que la cantidad de peajes que hay es menor a 200 decidimos concentrarnos en las soluciones que representaban a toda la cantidad de peajes posible .
Por otra parte, las opciones que contienen las colas de prioridad como características quedarían fuera ya que no se está buscando en especifico un tipo de peaje asi que no se estaría utilizando su verdadero propósito que es ordenar objetos de mayor a menor y también se estaría ignorando la necesidad de buscar el valor mayor o menor dentro de la lista ya que se busca mostrar todos los peajes en el mapa.

- Opción 1: La creación del árbol binario es una opción bastante buena ya que tiene la capacidad de agregar los peajes sin tener algún tipo de dificultad frente a la cantidad de objetos que este puede almacenar. Además, permite recuperar cada uno de los objetos con gran facilidad. Incluso el árbol binario puede organizar los peajes en orden de precio o alfabético si se hace autobalanceable.
- Opción 2: Crear un conjunto de diccionarios en el cual las llaves utilizadas son los nombres de los departamentos para agrupar los peajes que pertenezcan al mismo. Esto permitirá agrupar y recopilar de mejor manera los peajes y tener un mejor orden en la información
- Opción 3: Crear una lista con todos los peajes permite un fácil acceso por parte del programa al cargar y mostrar la información de los peajes en el mapa. Además no requiere contraseña o llaves para buscar los peajes ya que estos se encuentran almacenados todos en una misma estructura.
- Opción 4: Al crear una lista de listas se crea una lista con llaves que representen los departamentos, y cada uno de los compartimentos es una lista que contiene los peajes de ese departamento. Esto permite que sea facil la busqueda de los peajes por departamento y crea una organización de los datos específica.

5. Evaluación y Selección

Criterios

- Memoria: Este criterio evalúa que tantos recursos consume el programa cuando se ejecuta.
    1. Alto consumo de memoria
    2. Consumo de memoria moderado

3. Bajo consumo de memoria.

- Velocidad de Carga: Que tan rápido carga la información del archivo csv al programa en su estructura de datos.
    1. Lento: Se demora en cargar la información almacenada.
    2. Medio: Carga la información a una velocidad moderada.
    3. Alto: Carga de manera veloz la información almacenada
- Representación : La base de datos cumple con la necesidad principal del problema
    1. No muestra la información requerida de manera efectiva
    2. Muestra parcialmente la información requerida de manera efectiva
    3. Muestra completamente la información requerida de manera efectiva.

	Memoria	Velocidad de Carga	Representación	Total
Opción 1	3	2	2	6
Opción 2	3	3	2	7
Opcion 3	3	3	3	9
Opción 4	2	1	2	5

Memoria: para todas la estructuras usadas excepto la lista de listas (la opción 4) tienen una complejidad espacial de O(n) y la opción 4 que tiene una complejidad espacial de O(n^2)

Velocidad de carga: Para este caso se evaluó la complejidad temporal del método add de cadas estructura porque ese es el que influye más en el tiempo de carga del programa.

Representación: En cuanto a representación la idea de la aplicación es mostrar todas las estaciones de peaje que hay en el país y dar la información de ellas.
Y la lista es la mejor para esto porque da toda la información necesario en una sola iteración, además de ser la más versátil para el manejo de la información para este caso.
Por otro lado las otras estructuras son más especializadas en buscar los peajes por llaves o comparadores, sin embargo no es convencional hallar estaciones de peaje por nombre o precio. Además de que en los datos dados por el gobierno no hay rigurosidad en los nombres de la estaciones.

Estructura de datos	Complejidad temporal
Árbol binario	O(log n)
Diccionario	O(1), si hay colisión O(n)
Lista	O(1)
Lista de listas	O(n)

6. Síntesis

El desarrollo del método de la ingeniería ha permitido desarrollar un proyecto a base de unos datos y lograr un objetivo planteado en base a lo que estos expresan y así no solo es desarrollar soluciones sino también lograr desarrollar un conflicto en base a estos.
Segundo,en el desarrollo de las ideas se utilizó como estrategia principal el *brainstormin*g, y se obtuvieron una gran cantidad de posibles soluciones de las cuales terminaron siendo bastante similares las unas con las otras. Al continuar, se tomó en cuenta que ciertas ideas

presentaban dificultades al cumplir con algunos de los requisitos fundamentales que se habían propuesto para el desarrollo de la solución; al final después de hacer una evaluación general de ideas solo quedaron cuatro opciones.Profundizando en la evaluación se tomaron tres criterios que permitieron evaluar las cuatro ideas finales y descubrir que a veces la opción más sencilla pueda llevar a la mejor solución.

Por otra parte, el método de la ingeniería permite que se hagan observaciones generales para crear ideas bastante flexibles no solo par aun trabajo sino para proyectos que puedan tener muchas similitudes o se relacionen.

Referencias

Quetglás, Martin. Toledo, Francisco. Cerverón,Vicente.(2002). *Fundamentos de Informática y Programacion*. Recuperado de: http://robotica.uv.es/Libro/Indice.html

(2012).*Diccionarios*.Fundamentos de Programacion y Robotica: Grupo MINA. Recopilado de https://www.fing.edu.uy/inco/cursos/fpr/wiki/index.php/Diccionarios .

List<T> Class. .NET: Microsoft. Recopilado de: https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?view=netframework-4.7.2

Agencia Nacional de Infraestructura(2019) *Tabla de Costos de Peajes.* Recopilado de https://www.datos.gov.co/Transporte/Tabla-de-Costos-de-Peajes/jhpq-24h2