



## LABORATORIO NO. 02

### “Árboles Binarios de Búsqueda vs. Árboles AVL”

#### COMPETENCIAS

- ✓ Desarrollo de habilidades colaborativas a través del trabajo en equipo. [3 integrantes por equipo]
- ✓ Desarrollo de pensamiento crítico al decidir la manera más adecuada de implementación.

#### OBJETIVO

- ✓ Aplicación de estructuras de datos para el manejo de información.
- ✓ Practicar y familiarizarse con el Sistema de control de versiones de forma colaborativa.

#### PRÁCTICA EN LABORATORIO

##### Descripción

La SAT (Superintendencia de Administración Tributaria) ha iniciado un proceso de licitación para la implementación de un sistema que les permita almacenar los datos de los vehículos que se encuentran actualmente en circulación, y poder verificar si ya realizaron el pago de la calcomanía electrónica.

La empresa en la que usted labora desea participar y propone una POC (Prueba de concepto) utilizando un sistema web (MVC) que permita la administración de los registros de los vehículos, utilizando como base una estructura de datos capaz de minimizar el tiempo de búsqueda de los datos.

##### Requerimientos funcionales

Los requerimientos funcionales son los siguientes:

- Carga de datos:
  - o El usuario seleccionará en qué tipo de estructura de datos desea guardar la información, teniendo como opciones:
    - Árbol AVL
    - Árbol ABB
    - Ambas

→ 3



- La aplicación debe permitir el ingreso manual de los datos.
- La aplicación debe permitir la carga de datos a través de un archivo separado por comas (csv). **[Se adjunta archivo de ejemplo]**
- **Ordenamiento**
  - La aplicación deberá ser capaz de ordenar los datos por:
    - Dpi (id)
    - Número de serie (serie)
    - Correo electrónico (email)
  - El ordenamiento se deberá realizar ÚNICAMENTE a través del recorrido de la estructura de árbol
- **Búsqueda**
  - La aplicación deberá ser capaz de ordenar los datos por:
    - Dpi (id)
    - Número de serie (serie)
    - Correo electrónico (email)
  - La búsqueda se realizará ÚNICAMENTE a través de los mecanismos propios de la estructura de datos.
- Estadísticas
- La aplicación mostrará las estadísticas para cada operación
  - Tiempo de carga del archivo.
  - Tiempo de ordenamiento de los datos.
  - Para búsquedas, deberá mostrar cuantas comparaciones fueron necesarias para llegar al objetivo.
  - Si la estructura es un árbol AVL mostrará cuantas rotaciones fueron necesarias para equilibrar el árbol.
  - Profundidad actual del árbol.

### Requerimientos no funcionales

Adicionalmente los requerimientos no funcionales:

- Aplicación web utilizando el patrón MVC.
- Estructura de datos
  - Los datos deben ser almacenados en una estructura de datos árbol AVL y Árbol Binario de Búsqueda.
- Ordenamiento de datos
  - El ordenamiento de los datos debe realizarse a través de un recorrido "In-order" del árbol y no se podrá utilizar funciones propias de .net, Bootstrap, etc.
- Almacenamiento de datos



- Debido a que no se utilizará un motor de base de datos, la información será almacenada en memoria.
- Deberá crear un proyecto solamente para las estructuras y otro para su implementación, deberá incluir este proyecto utilizando DLLs.
- Control de versiones
  - El proyecto debe ser subido a un repositorio en Github, privado y como colaboradores deben estar los integrantes del grupo y el catedrático.
  - TODOS los integrantes del equipo deberán subir sus cambios individualmente al repositorio para poder comprobar el trabajo realizado a través de los "commits" que cada usuario ha hecho.
  - Deberá utilizar ramas para organizar el modo de trabajo.