



FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA DE SISTEMAS

ESTRUCTURAS DE DATOS  
JORGE OSWALDO VALENZUELA BUITRAGO.

PROYECTO ENTREGA #1

SANTIAGO ANDRÉS LEAL OMAÑA

SANTIAGO ZUÑIGA MARTÍNEZ

NICOLÁS SÁNCHEZ LAITON

08/09/2022

## **Introducción:**

Llevar el control de las distintas formas de organización de un país siempre ha sido importante para poder y administrar los distintos recursos que circula por medio de estas, en caso de Colombia según la división político-administrativa, esto se comprende en un sistema de codificación de los departamentos, municipios y centros población que conforman de forma jerárquica el territorio de Colombia, puede ser importante llegar a manejar ciertos datos y operaciones con estas operaciones políticas, el propósito del proyecto es poder realizar un programa que permita hacer estas operaciones de manera sencilla usando el programa de C++ y los conceptos aprendidos durante las clases de Estructuras de Datos

## **1)Commands:**

En este hpp se incluye las librerías map, fstream, string, vector y iostream, además de incluir unos .hpp como structures.hpp y utils.hpp

Funciones:

-carga\_divipola

Aspectos sintácticos:

-Nombre: carga\_divipola.

-Parámetros de entrada:

-filename (nombre del archivo a cargar).

-dpto (mapa de llaves de departamento con identificador).

-Resultado devuelto: No retorna nada.

Semánticos: Carga en una o más estructuras de datos en memoria el contenido del archivo identificado por filename.

Complejidad:  $O(n)$

-listar\_departamentos

Aspectos sintácticos:

-Nombre: listar\_departamentos

-Parámetros de entrada:

-dpto (mapa de llaves de departamento con identificador).

-Resultado devuelto: No retorna nada.

Semánticos: Imprime en n líneas (una para cada departamento) la información básica del departamento que se cargó de la Divipola.

Complejidad:  $O(n)$ .

#### -listar\_municipios

Aspectos sintácticos:

-Nombre: listar\_municipios.

-Parámetros de entrada:

-code (codigo del departamento).

-dpto (mapa de llaves de departamento con identificador).

Resultado devuelto: No retorna nada.

Semánticos: Imprime en n líneas (una para cada municipio) la información básica del departamento que se cargó de la Divipola.

Complejidad:  $O(n \log(n))$

#### -listar\_poblaciones

Aspectos sintácticos:

-Nombre: listar\_poblaciones.

-Parámetros de entrada:

-code (codigo del departamento).

-dpto (mapa de llaves de departamento con identificador).

-Resultado devuelto: No retorna nada.

Semánticos: Imprime en n líneas (una para cada población) la información básica del departamento que se cargó de la Divipola.

Complejidad:  $O(n \log(n))$

#### -info\_sumaria

Aspectos sintácticos:

-Nombre: info\_sumaria.

-Parámetros de entrada:

-code (codigo del departamento).

-dpto (mapa de llaves de departamento con identificador).

-Resultado devuelto: No retorna nada.

Semánticos: Imprime en una línea la información básica del nombre del departamento que corresponde al código dado como parámetro junto con la cantidad de municipios y centros poblados que lo conforman.

Complejidad:  $O(n \log(n))$

#### -carga\_SC

Aspectos sintácticos:

-Nombre: carga\_SC.

-Parámetros de entrada:

-file\_name (nombre del archivo a cargar).

-dpto (mapa de llaves de departamento con identificador).

-Resultado devuelto: No retorna nada.

-Semánticos: Carga en una o más estructuras de datos en memoria el contenido del archivo identificado por file\_name con la información básica para representar el sistema de ciudades.

Complejidad:  $O(n \log(n))$

#### -esta\_en\_sistema

Aspectos sintácticos:

-Nombre: esta\_en\_sistema.

-Parámetros de entrada:

-divipola (codigo de un municipio).

-dpto (mapa de llaves de departamento con identificador).

-Resultado devuelto: No retorna nada

Semánticos: Determina si un municipio, con código dado por el usuario, existe dentro de los municipios definidos en el Sistema de Ciudades y cargados desde el archivo correspondiente.

Complejidad:  $O(n \log(n))$

## 2)Utils

En este hpp se incluye las librerías string, vector y algorithm.

Funciones:

-trim

Aspectos sintácticos:

-Nombre: trim

-Parámetros de entrada:

-input (cadena de texto la cual será editada).

-Resultado devuelto: Retorna el string con los espacios extra removidos.

Semánticos: Quita los espacios adicionales que puede haber en un string, ya sea por error del usuario al ingresar el string o por temas del mismo programa.

-tokenize

Aspectos sintácticos:

-Nombre: tokenize

-Parámetros de entrada:

-s (Cadena de texto que será tokenizada).

-del (cadena de texto constante).

-dlim (delimitador).

-Resultado devuelto: Regresa un vector en donde se encuentra el string tokenizado.

Semánticos: Esta función busca en el string el delimitador que se pasó por parámetro para ir dividiendo el strings en partes, y estas partes las va guardando en un vector llamado tokens el cual devolverá el string, pero ya separado. Para los casos en los que el vector esta entre comillas, la función revisa si hay comillas para saltar y luego dividir el string.

Complejidad:  $O(n \log(n))$

### **3)Help**

Este hpp incluye las librerías iostream y string.

Funciones:

-help

Aspectos sintácticos:

-Nombre: help

-Parámetros de entrada:

-command (cadena de texto).

Resultado devuelto: No retorna nada.

Semánticos: Con esta función lo que se busca es dar información sobre el comando que se puede usar, para eso el usuario ingresa en la terminal el comando help seguido del comando del cual quiere saber su uso, en caso de que el comando sea desconocido, se mostrara en pantalla un mensaje de error.

Complejidad:  $O(1)$

-help

Aspectos sintácticos:

-Nombre: help.

-Parámetros de entrada: No recibe parámetros.

-Resultado devuelto: No retorna nada.

Semánticos: Esta función le muestra al usuario los distintos tipos de comandos que hay, para acceder a esta función toca colocar el comando help sin ningún comando extra, para que se despliegue una lista de todos los comandos posibles para usar.

Complejidad:  $O(1)$

#### **4) Interpreter**

##### **-interpreter**

Aspectos sintácticos:

-Nombre: interpreter

-Parámetros de entrada: No recibe parametros.

-Resultado devuelto: No retorna nada

Semánticos: Es la función la cual toma los datos suministrados por el usuario en pantalla y luego los transforma en la información que se necesita.

Complejidad:  $O(1)$

##### **-executeCommand**

Aspectos sintácticos:

-Nombre: executeCommand

-Parámetros de entrada:

-tokens (vector constante).

-dpto (mapa de llaves de departamento con identificador).

-Resultado devuelto: No retorna nada.

Semánticos: Esta función va comparando la primera posición del vector para saber cual es el comando que se pide y así llevar al usuario a la función pertinente.

Complejidad:  $O(1)$

#### **Estructuras:**

##### **CP**

Aspectos sintácticos:

-nombre: CP.

-Modelo:

-name.

-latitude.

-longitude.

Para cada operación:

-Nombre: CP

-Parámetros de entrada:

-name (cadena de texto).

-latitude (número decimal).

-longitude (número decimal).

-Resultado devuelto: No retorna nada.

Semánticos: Es una estructura en la que se guarda la información de los centros poblados.

### ANM

Aspectos sintácticos:

-nombre: ANM

-Modelo:

-name.

-anm.

-cp.

-sc.

Para cada operación:

-Nombre: ANM

-Parámetros de entrada:

-n (cadena de texto constante).

-Resultado devuelto: No retorna nada.

Semánticos: Es una estructura en la que se guarda la información de las áreas no municipalizadas, que son clasificadas como municipios, pero no cumplen con los requisitos mínimos.

### CM

Aspectos sintácticos:

-nombre: CM

-Modelo: name, cp, latitude, longitude, population, sc.



Para cada operación:

-Nombre CM

-Parámetros de entrada:

-n (cadena de texto constante).

-L1 (número decimal constante).

-L2(número decimal constante).

-Resultado devuelto: No retorna nada.

Semánticos: Es una estructura en la que se guarda la información de los municipios.

### Department

Aspectos sintácticos:

-nombre: Department.

-Modelo: name, cm, anm.

Para cada operación:

-Nombre: Department.

-Parámetros de entrada:

-n (cadena de texto constante).

-Resultado devuelto: No retorna nada.

Semánticos: Es una estructura en la que se guarda la información de los departamentos.

