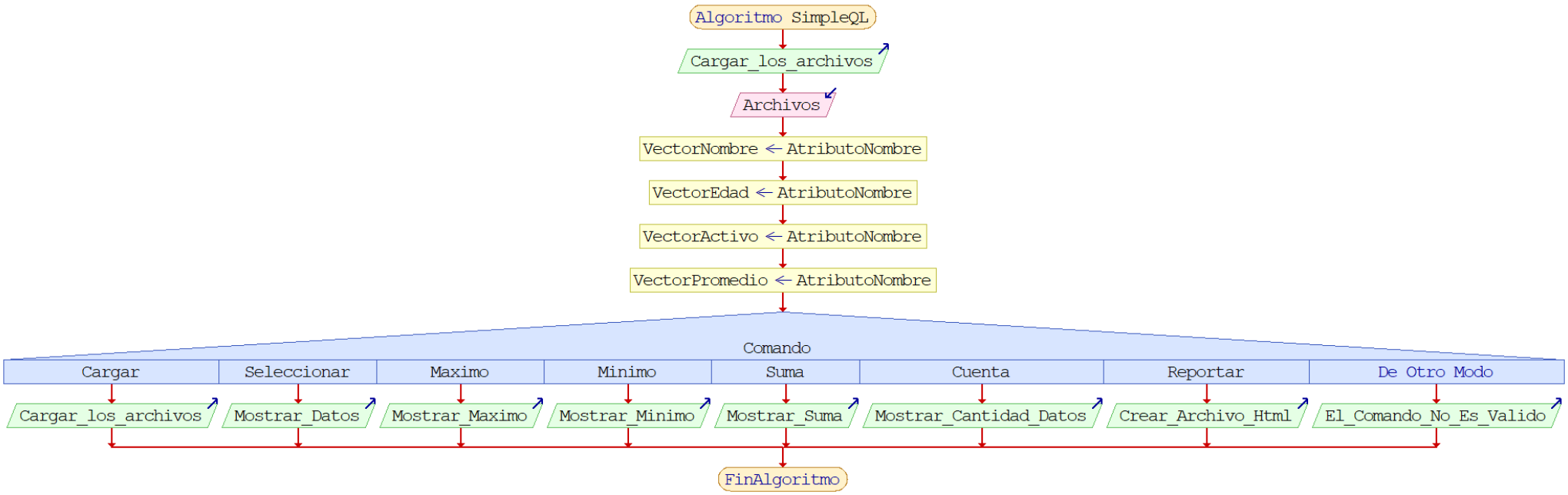
Nombre: Wuilmar Hamilton López García Carné: 201901069

Lenguajes Formales y de Programación A+

Manual Técnico SimpleQL.

Diagrama de flujo funcionamiento de la aplicación.



Métodos utilizados.

- cargar_datos(ruta): Recibe un parámetro llamado ruta el cual deberá contener la dirección del archivo que contiene los registros y se desea cargar a la memoria para poder trabajar en la aplicación, si el archivo no existe o no se encuentra en la dirección proporcionada le mostrara al usuario un mensaje de que ha ocurrido un error y le solicitara que cargue de nuevo los archivos. Cabe mencionar que la aplicación permite cargar únicamente archivos con formato json para trabajar.

- inicio(): El método pide una entrada de texto al usuario para guardarlo y descomponer la cadena en un vector para analizar el comando que desea ejecutar el vector

- desarrollo(): Analiza el vector creado y definido en el método inicio() y ejecuta las condiciones que debe tener la cadena de texto para identificar el comando que desea ejecutar el usuario. Este método es recursivo ya que se llama así mismo en todo momento para no detener el flujo del programa y pedir al usuario el comando que desea ejecutar.

- reporte(texto): Este método recibe un parámetro llamado texto el cual contendrá los datos que se desean imprimir en el reporte, seguido de esto generara un documento html el cual contiene un estilo definido en la carpeta del proyecto. Seguido de esto abrirá en el navegador predeterminado del computador del usuario el reporte con la cantidad de registros que el usuario haya solicitado mediante el comando “reportar” en el flujo del programa. Los registros que se mostraran en el reporte son los registros que estarán guardados en los archivos json que se cargaran al principio del flujo de la aplicación.

Para el análisis sintáctico se utilizó la función .Split() para separar las cadenas te texto que ingresaba el usuario mediante el teclado, guardando la información en vectores y analizando su contenido para determinar la acción que desea ejecutar.

Como la aplicación es case insensitive se utilizo la función .lower() para convertir todas las cadenas de texto ingresadas a minúscula y asi analizar su contenido.

Vectores Utilizados.

Para el desarrollo de la aplicación se utilizaron vectores tanto para analizar la entrada de texto como para guardar en memoria los datos de los registros provenientes de los archivos json. Para cada atributo del archivo json se utilizo un vector y para poder acceder a la información de ese registro se hace mediante la posición (index) de los datos en cada vector por individual.

- Vector_Nombre[ ]: Este Vector contendrá los nombres de todos los registros cargados a memoria de la aplicación.

- Vector_Edad[ ]: Este Vector contendrá las edades de todos los registros cargados a memoria de la aplicación.

- Vector_Activo[ ]: Este Vector contendrá el estado del todos los registros cargados a memoria de la aplicación.

- Vector_Promedio[ ]: Este Vector contendrá los promedios de todos los registros cargados a memoria de la aplicación.

- vector[ ]:Este Vector contendrá el comando y la entrada de texto que el usuario haya ingresado separándolo mediante un espacio y guardándolo de una forma:

  [“cargar”,”archivo.json, archivo2.json”]

- vector2[ ]: Dependiendo del comando que haya en el primer vector este segundo vector contendrá los archivos o atributos de los archivos json que el usuario desea ejecutar ejemplo:

  [“archivo.json”,” archivo2.json”]

Archivos utilizados.

Json: Este tipo de archivos son utilizados para guardar los registros con sus atributos, se utilizara este tipo de archivos para trabajar el aplicación cargando los registros contenidos en memoria para ejecutar los comandos. Si el archivo se desea cargar a la aplicación deberá tener la siguiente estructura:

```
[
    {
        "nombre": "registro 1",
        "edad": 45,
        "activo": true,
        "promedio": 56.456
    },
    {
        "nombre": "registro 2",
        "edad": 35,
        "activo": false,
        "promedio": 45.896
    }
]
```

Html: Este archivo se utilizara cuando se desee generar un reporte de registros. En el mismo se mostrara la cantidad de registros que el usuario desee teniendo siempre la misma estructura y mostrando en pantalla una tabla ordenada con los registros.

Css: Este archivo se utilizara para darle estilo y visualización al documento html mencionado anteriormente para hacer la interfaz más amigable al usuario y comprensible.