**Algoritmos y programación II**

**Tarea Integradora II - Especificación de requerimientos**

**Integrantes:**

* Ana Sofia Londoño A00380882
* Juan Jose Diaz A00381098

**Requerimientos funcionales:**

**R1: Desplegar el menú.** El sistema debe permitir desplegar un menú por consola que muestre y ejecute las siguientes acciones:

1. Insertar comando

2. Importar datos desde archivo .SQL

2. Salir

**R2: Almacenar países y ciudades.** El sistema debe permitir almacenar la información de países y ciudades. Los países deben de tener los siguientes atributos:

* ***Id***. De tipo String. Debe ser generado mediante UUID
* ***Name*.** De tipo String. Es el parámetro con el nombre de la ciudad
* ***Population.*** De tipo Double. Almacena la población actual de la ciudad en millones de personas
* ***CountryCode.*** De tipo String. Representa el indicativo del país

Por otro lado, las ciudades deben de tener los siguientes atributos:

* ***Id***. De tipo String. Debe ser generado mediante UUID
* ***Name*.** De tipo String. Es el parámetro con el nombre de la ciudad
* ***CountryID*.** De tipo String. Contendrá el identificador del país
* ***Population.*** De tipo Integer. Almacena la población actual de la ciudad

**R3: Agregar datos.** Mediante el comando INSERT INTO, el usuario podrá agregar un dato a la base de datos. La estructura del comando es la siguiente:

INSERT INTO <Nombre del tipo del objeto> (<Nombre del atributo>, <Nombre del atributo>, … <Nombre del atributo>) VALUES

(<Valor del atributo>, <Valor del atributo>, … <Valor del atributo> )

Los valores de los atributos de tipo String deben estar entre comillas simples (‘’) a excepción de que estemos tratando de un valor de tipo numérico (int o double). Si el usuario intenta agregar una ciudad cuyo countryID es inexistente en la base de datos, el programa debe mostrar una excepción en pantalla. Así mismo, si uno de los datos de entrada no está con el formato correcto, también debería lanzar una excepción.

**R4: Búsqueda y filtrado:** Mediante el comando SELECT, el usuario podrá buscar algo en la base de datos. Este comando tiene la siguiente estructura:

SELECT \* WHERE <Nombre del tipo de objeto> WHERE <Nombre de un atributo del objeto> <Signo de comparación> <Valor a comparar>

El nombre del tipo del objeto debe de ser o “countries” o “cities” (sin comillas). Si dentro de esta expresión no colocamos la palabra WHERE el sistema nos debe mostrar todos los objetos de ese tipo guardados en el sistema. El nombre del atributo de objeto debe de ir sin comillas y debe de pertenecer al objeto que estamos buscando. El signo de comparación puede ser >, < o = (sin comillas). Si el valor a comparar se trata de un String este debe de ir en comillas simples pero si este se trata de un valor double o entero este debe de ir sin comillas.

**R5: Ordenamiento:** A los comandos que inicien con el comando SELECT también se pueden combinar con ordenamientos y el sistema debe mostrar en pantalla los elementos con la condición con el orden especificado. Estos comandos deben tener la siguiente estructura:

<Comando iniciado en SELECT> ORDER BY <Nombre del atributo a ordenar>

El nombre del atributo a buscar debe de estar en el tipo de objeto a buscar. Si el resultado de la búsqueda sólo da un resultado, el ORDER BY será ignorado.

**R6: Eliminacion:** A traves del comando DELETE el programa debe permitir eliminar los registros dentro del sistema:

DELETE FROM <Nombre del tipo de objeto> WHERE <Nombre del atributo> <Signo de comparación> <Valor de selección>

El nombre del tipo del objeto debe de ser o “countries” o “cities” (sin comillas). El nombre del atributo de objeto debe de ir sin comillas y debe de pertenecer al objeto que estamos buscando. El signo de comparación puede ser >, < o = (sin comillas). Si el valor a comparar se trata de un String este debe de ir en comillas simples pero si este se trata de un valor double o entero este debe de ir sin comillas.

**R7: Almacenamiento e importación de datos:** El sistema debe poder almacenar toda la información de la base de datos en formato JSON de modo que al iniciar el programa pueda recuperar esta información y volverla a cargarla al programa.

Asimismo, el programa debe poder importar datos desde un cualquier archivo con extensión .sql que tenga el formato correcto, utilizando la opción 2 del menú. Dentro de este archivo deben de haber comandos como los que hay en el requerimiento R3 (Agregar datos). El programa debe ejecutarse de forma secuencial y si hay un error dentro de los comandos, el programa debe de mostrar una excepción.