Objetivos generales

- Comprender y aplicar los principios de las estructuras básicas de datos.
- Aplicar los conocimientos obtenidos en el laboratorio
- Conocer otro tipo de lenguajes en los que se pueda programar EDD.

Objectivos específicos

- Que el estudiante pueda aplicar arboles B +.
- Entender la utilidad de los árboles en aplicaciones reales.

Descripción de la actividad

Una de las herramientas más utilizadas por los informáticos son las bases de datos, estas contienen diferentes tipos de EDD internas con las cuales manejan los datos manteniendo una consistencia de los mismos y optimizando los tiempos de respuesta;

Por lo cual se le solicita crear un DBMS (DataBase Management System) relacional básico, estructura, captura de datos y las salidas serán manejadas por archivos de texto e interfaz gráfica. Para esto deberá utilizar un árbol B+;

Deberá crear una interfaz gráfica con el diseño que deseen bajo los siguientes criterios:

- Que cumplan la misma funcionalidad de los archivos de entrada
- Sea agradable e intuitiva con el usuario final

Entre las opciones extra de la UI del DBMS deberá poder mostrar los datos en tablas y mostrar un diagrama entidad relación del estado actual de las estructuras. (Recomendado Entidad relación de Peter Chen)

Carga Masiva

Múltiples archivos de entrada en formato XML que contendrán las instrucciones para crear las estructuras (Tablas) y realizar las operaciones de creación, eliminación, búsqueda y actualización de los datos dentro del árbol. Los archivos de entrada les puede cambiar el nombre y únicamente desde IU definir qué archivo va a entrar. El formato será el siguiente:

Estructura

- El nombre del archivo será: Estructura.xml
- Iniciará con la etiqueta `<estructura>` en ella podremos encontrar el formato completo;
- La etiqueta `<tabla>` definirá el nombre de nuestra tabla/entidad
- La etiqueta `<clave>` definirá la llave primaria de la tabla (El nombre deberá ser igual al de la etiqueta a la que hace referencia)
- El caso como `<nombreCampo> tipoDeDato </nombreCampo><TablaConRelacion>` Define que ese campo se relaciona con el campo nit de otra tabla. Ej. `<Nit> int </Nit><cliente>`
- El resto de etiquetas cumpliran el siguiente formato `<nombreCampo> tipoDeDato </nombreCampo>`

  Nota: Para facilitar la relación los campos con relación tendrán el mismo nombre; en el ejemplo se muestra Nit, por lo cual tanto en factura como en cliente los 2 se llaman Nit y tienen el mismo tipo de dato.

```
<estructura>
     <tabla> factura </tabla>
     <NoFactura> int </NoFactura>
     <Nit> int </Nit><cliente>
     <Fecha> char </Fecha>
     <Lugar> char </Lugar>
     <Valor> int </Valor>
     <clave> Nofactura </clave>
</estructura>
<estructura>
     <tabla>cliente</tabla>
     <Nit>int</Nit>
     <Nombre>char</Nombre>
     <Apellido>char</Apellido>
     <clave>Nit</clave>
</estructura>
```

Carga de datos

- El nombre del archivo será: entrada.dat
- La etiqueta inicial definirá la tabla/entidad a la que pertenece.
- El resto de etiquetas hacen referencia al campo al que pertenecen
- Deberá comprobar que los tipos de datos coincidan con el modelo (Ej si el tipo de dato del modelo era int, el dato a entrar debe ser int)

```
<producto>
     <Descripción>Boligrafo color negro</Descripción>
     <CodProducto>BL0001</CodProducto>
     <Valor>1.25</Valor>
</producto>
```

Eliminación

- El nombre del archivo será: elimina.dat
- La etiqueta inicial definirá la tabla/entidad a la que pertenece.
- El resto de etiquetas hacen referencia al campo al que pertenecen (por el cual deberá buscar el valor a eliminar)
- recuerde mantener la integridad de los datos, no debe permitir dejar datos huérfanos ni borrar en cascada.

```
<producto>
     <CodProducto>BL0001</CodProducto>
</producto>
```

Reporte

- El nombre del archivo será: reportes.rpt
- Iniciará con la etiqueta `<reporte>` en ella podremos encontrar el formato completo;
- La etiqueta `<tabla>` definirá el nombre de nuestra tabla/entidad
- La etiqueta `<clave>` definirá la llave primaria de la tabla (El nombre deberá ser igual al de la etiqueta a la que hace referencia)

```
<reporte>
     <producto>Valor</producto>
</reporte>
<reporte>
     <producto,cliente>factura</producto,cliente>
</reporte>
```

Notas:

Los archivos de entrada **NO TENDRÁ ERRORES LÉXICOS NI SINTÁCTICOS** sin embargo si puede contar con errores de tipos de datos diferentes, la etiqueta clave no exista y otros qué puedan hacer qué su árbol falle.

Todas las estructuras de datos utilizadas deberán ser programadas por uds, no se pueden usar librerías para estas.

Importante

- Usar lenguaje de programación Java
- Copias obtendrán nota de cero y se notificará a coordinación.
- Si se va a utilizar código de internet, entender la funcionalidad para que se tome como válido.

Entrega

La fecha de entrega es el día martes 09 de Mayo a las 14:00. Los componentes a entregar en repositorio de git son:

- Código fuente y ejecutable (java/.jar)
- Documento con el análisis para la aplicación y manual técnico.
- Manual de usuario.

Calificación

Pendiente de definir.