

TRABAJO CON BASEX

- Crear el contexto de la base de datos.

```
private Context contexto; //Representa la BD
private String directorioBD = "src/recursos/colecciones.x
private final OperacionesXML operadoracionesXML = new Ope
private final OperacionesIO operadoracionesIO = new Opera

//Método para crear base de datos.
private void crearBD() {
    if (contexto == null) {
        contexto = new Context();

        try {
            CreateDB baseDatos = new CreateDB( name:"Colecc
            baseDatos.execute( ctx:contexto);
```

contexto: se utilizará para interactuar con la base de datos.

baseDatos: Objeto que almacenará lo necesario para la creación de la base de datos (el nombre y la ruta)

baseDatos.execute(contexto): Se llama al método execute() del objeto **baseDatos** para ejecutar la creación de la base de datos. Se pasa el objeto **contexto** como argumento, lo que permite interactuar con la base de datos a través de este contexto.

- Para leer el script del archivo xquery y guardarlo en una cadena que después se procesa.

```
public class OperacionesIO {  
  
    private final String RUTA_SCRIPT = "src/script/";  
  
    //Método para leer el script y guardarlo en un String.  
    public String leerScript(String nombreFichero) {  
        try {  
            String fichero = RUTA_SCRIPT + nombreFichero;  
            File file = new File(pathname:fichero);  
            return new String(bytes:Files.readAllBytes(path:file.toPath(),  
        } catch (IOException ex) {  
            System.out.println(x:"Error al lerr el fichero");  
        }  
    }  
}
```

Files.readAllBytes, devuelve un array de bytes con todos los bytes del archivo. Este array de bytes se convierte luego a una cadena de caracteres (String) utilizando el constructor de la clase String El conjunto de caracteres (Charset) especifica cómo decodificar los bytes en caracteres. En este caso, se utiliza StandardCharsets.UTF_8 como conjunto de caracteres para ello.

- Para ejecutar el script del archivo xquery.

```
//Método para lanzar el Script, recibe cómo parámetro el nombre del Scrip
public String consultas(String nombreFichero) {

    if (contexto == null) {
        crearBD();
    }
    try {
        if (operadoracionesIO.leerScript(nombreFichero).length() != 0) {
            //Leemos la consulta llamando al método de leerScript y la ej
            String xq = new XQuery( query: operadoracionesIO.leerScript(nomk
            return xq;
        } else {
            return "Error al realizar la consulta a la base de datos.";
        }
    }
}
```

El **String xq** tiene los resultados que podremos mostrar por consola o bien en un control de un JForm.

En el SGBD BaseX, al escribir una consulta xquery, al trabajar con todo el documento xml hay que referenciar cada nodo según si está en mayúsculas o minúsculas. Por ejemplo, si tenemos el documento xml siguiente.

```
<!-- Fichero: productos.xml -->
<instituto>
  <aulas>
    <aula id="1">
      <nombre>Aula 1</nombre>
      <edificio>Edificio Principal</edificio>
      <puestos>30</puestos>
    </aula>
```

En la consulta xquery, instituto debe ser referenciado tal cual, en minúscula porque así aparece en el documento xml:

```
for $curso in /instituto/cursos/curso
where $curso/plazas = 20
order by $curso/nombre
return $curso/nombre
```

CONSULTAS XQUERY DE ACTUALIZACION EN BASEX

Consultas de modificación de valor:

```
for $productos in //tienda/productos/producto
where $productos/@id = "9"
return replace value of node $productos/pvp with "250"
```

Consultas de modificación de nodo:

Opción 1:

```
for $productos in //tienda/productos/producto
return replace node $productos/pvp with <precio>{$productos/pvp/text()}</precio>
```

Opción 2:

```
let $cursos := instituto/cursos/curso
return rename node $cursos[1]/plazas as "puestos"
```

Consultas de inserción:

```
let $nuevoVenta :=
  <venta numero="1003">
    <fecha>20/02/2024</fecha>
    <cliente>
      <nombre>Luis</nombre>
      <apellidos>García Pérez</apellidos>
      <direccion>Calle Mayor, 10</direccion>
      <telefono>444444444D</telefono>
    </cliente>
    <importe>350</importe>
    <iva>21</iva>
  </venta>
return
  insert node $nuevoVenta into /tienda/ventas
```

Consultas de eliminación:

```
for $aulas in /instituto/aulas/aula  
return delete node $aulas/puestos
```