

AD03. Manejo de conectores



Francisco Martínez Marín

2º DAM

Contenidos

0. Estructura del proyecto	3
1. Cargar información en las tablas	3
2. Mostrar toda la información de una tabla	3
3. Mostrar países filtrando por forma de gobierno	3
4. Insertar nueva ciudad	3
5. Borrar ciudad a partir de su id	4
6. Modificar el país al que pertenece una ciudad	4
7. Extra: Consulta SELECT en cualquier tabla y cualquier campo	4
8. Referencias	6

0. Estructura del proyecto

Tenemos los siguientes paquetes:

- **controladores:** Donde se produce la interacción con el usuario.
- **Entities:** Cada una de las clases de este paquete representa una tabla de nuestra base de datos.
- **DAO:** Las clases CiudadDAO, PaisDAO e IdiomaPaisDAO incluyen todo lo necesario para modificar y obtener consultas de la base de datos. También tenemos la clase Conexión que utilizamos para conectarnos a la base de datos y controlar los errores que puedan surgir en el proceso. La clase GenericoDAO permite obtener consultas de cualquier tabla y con cualquier campo.

1. Cargar información en las tablas

Tras la aparición del menú y elegir la opción “1” se crean cinco objetos de cada entidad (Pais, Ciudad y IdiomaPais) y se insertan en sus tablas correspondientes (country, city, countrylanguage).

2. Mostrar toda la información de una tabla

Elegimos la opción “2”, se nos muestra otro menú en el que podemos elegir la tabla que queremos que se muestre, obtenemos el resultado de la consulta con un DAO y la mostramos.

3. Mostrar países filtrando por forma de gobierno

Tras elegir la opción “3” mostramos otro menú para elegir la forma de gobierno por la que se quiere filtrar los países, evitando que el usuario introduzca directamente la forma de gobierno. Hacemos la consulta y mostramos el resultado.

4. Insertar nueva ciudad

Vamos solicitando cada uno de los campos necesarios para crear un registro en la tabla ciudad y los vamos validando. Una vez los tenemos todos creamos un objeto de tipo ciudad y llamamos al método CiudadDAO.crearCiudad(Ciudad) para realizar la inserción de este objeto de tipo Ciudad.

5. Borrar ciudad a partir de su id

Mostramos todas las ciudades para que el usuario pueda elegir qué ciudad va a borrar. Solicitamos al usuario el id de la ciudad que desea borrar, validamos que el id sea un entero entre 1 y 999999 y procedemos a llamar al método `CiudadDAO.borrarCiudad(int id)` que elimina el registro de la tabla `city` con ese id.

6. Modificar el país al que pertenece una ciudad

Para modificar el país al que pertenece una ciudad debemos cambiar el `CountryCode` del registro de la tabla `city`. Además el nuevo `CountryCode` debe existir. Por lo que lo primero que hacemos es mostrarle al usuario la relación de `CountryCodes` con el nombre de sus respectivos países. Después mostramos la tabla `city` para que pueda elegir a qué ciudad va a cambiarle el `CountryCode`. Le indicamos que nos introduzca el id de la ciudad que quiere cambiar de país y el nuevo código del país. Antes de llamar al método `CiudadDAO.cambiarPaisCiudad(int id, String codigoPais)` comprobamos que el nuevo `codigoPais` introducido por el usuario exista en la base de datos.

7. Extra: Consulta SELECT en cualquier tabla y cualquier campo

Ya lo tenía todo hecho y me he aventurado a mejorar la abstracción.

Como tenía que modificar las entidades he decidido crear unas nuevas (`CiudadGen`, `PaisGen`, `IdiomaPaisGen`).

También he creado un nuevo DAO (`GenericoDAO`) que permite hacer consultas simples a cualquier tabla y con los campos que quieras, aun así hay que crear una instancia de `GenericoDAO` por cada tabla/entidad.

El mayor desafío ha sido interactuar con objetos genéricos, que bien podrían ser de tipo `Pais`, `Ciudad` e `IdiomaPais`.

El objeto `Field` de `java.lang.reflect` permite interactuar con los atributos de objetos genéricos y modificarlos en tiempo de ejecución. Es decir que permiten interactuar con atributos sin saber de qué clase son (pueden ser de varias). He tenido que cambiar el nombre a los atributos de las nuevas entidades y los he dejado tal y como aparecen en la base de datos para evitar problemas.

Al elegir la opción “7” mostramos una lista con las tablas que están disponibles para mostrar.

El usuario elige una y nosotros comprobamos que la que ha elegido es una de las disponibles si no fuera así salimos al menú principal.

Si es una de las disponibles obtenemos el nombre de los campos de tabla y los mostramos para que el usuario pueda elegir que campos quiere que le mostremos de esa tabla, si no es una de las disponibles salimos al menú principal.

Le indicamos al usuario que tiene que introducir cada campo separado por un espacio. Comprobamos que todos los campos que ha introducido son válidos, hacemos la consulta mediante el objeto DAO correspondiente y mostramos el resultado. Para que no apareciera null cuando no se ha solicitado ese campo en el toString de cada entidad, cuando un atributo es null nuestro guiones.

8. Referencias

1. Apuntes y tutorías de la unidad 3.
2. Libro [Core Java, Volume II: Advanced Features, 13th Edition](#)
3. Imágen de portada y usuarios generada con [Stable Diffusion XL](#).