**-Bases de datos**

**Sistema gestor de tutoriales**

**Autores:**

* Castillo Angulo Adrian Camilo
* Escobar Valencia Isabella
* **Descripción del problema**

Construir un sistema gestor de tutoriales coda tutorial debe almacenar la información de una categoría que puede ser (lógica de programación, flutter, node.js) el acceso al tutorial se da a partir de una URL, de igual forma estableces una prioridad de lectura (1 - 10), estado (revisado, por revisar).

1. **Requerimientos funcionales**

|  |  |
| --- | --- |
| Nombre | R1- Ingresar un tutorial |
| Resumen | Ingresar datos en la base de datos del sistema gestión de tutoriales |
| Entradas | Nombre, prioridad, estado, url |
|  |  |
| Resultados | Mensaje de confirmación (“se ingresó un nuevo tutorial”) |
|  | |

|  |  |
| --- | --- |
| Nombre | R2- Editar un tutorial |
| Resumen | Permitirá a los usuarios editar información de los tutoriales |
| Entradas | Ninguna |
|  |  |
| Resultados | Se modifica la información de un tutorial de manera exitosa |

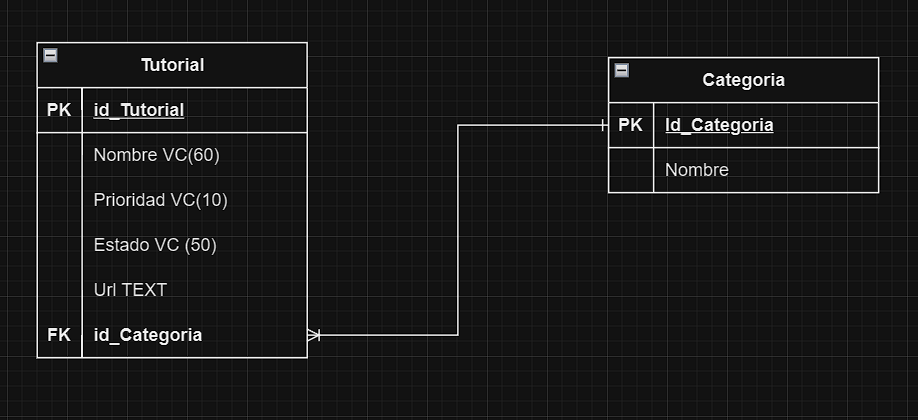
|  |  |
| --- | --- |
| Nombre | R3- Mostrar la información del tutorial |
| Resumen | Permitirá mostrar la información de los tutoriales |
| Entradas | Ninguna |
|  |  |
| Resultados | Aparecerá la información sobre el tutorial agregado |

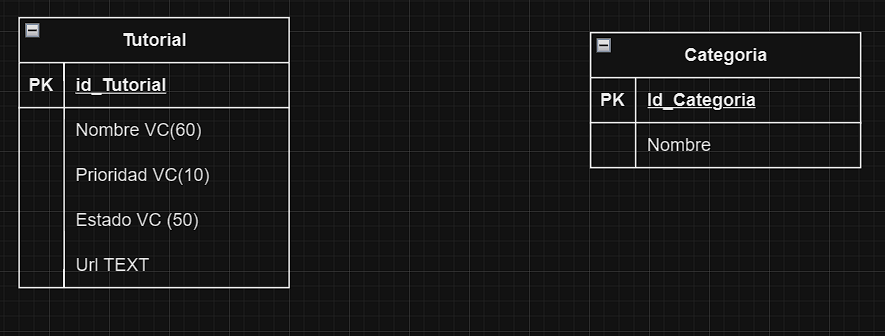
|  |  |
| --- | --- |
| Nombre | R4- Eliminar un tutorial |
| Resumen | Desplegar una lista para mostrar la información de todos los tutoriales que están registrados en la base de datos |
| Entradas | Ninguna |
|  |  |
| Resultados | El tutorial se eliminó con éxito |
|  | |

|  |  |
| --- | --- |
| Nombre | R5- Agregar una categoría |
| Resumen | Ingresar una nueva categoría en la base de datos de la gestión de tutoriales |
| Entradas | Ninguna |
|  |  |
| Resultados | Mensaje de confirmación, “se ingresó una nueva categoría” |

|  |  |
| --- | --- |
| Nombre | R6- Editar una categoría |
| Resumen | Se permitirá a los usuarios editar la información de una categoría |
| Entradas | Ninguna |
|  |  |
| Resultados | Se modifica la información de una categoría de manera exitosa |

|  |  |
| --- | --- |
| Nombre | R7- Eliminar una categoría |
| Resumen | Se permitirá a los usuarios eliminar la información de las categorías |
| Entradas | Ninguna |
|  |  |
| Resultados | Se modifica la información de un tutorial y categoría de manera exitosa |

1. **Requerimientos Técnicos**
   * Aplicar conceptos de SQL para la creación de la base de datos y administrarla
   * Utilizar MySQL como el motor de base de datos y Workbech para gestionarla.
   * Utilizar el framework Maven para gestionar las dependencias del proyecto.
   * Aplicar conceptos de programación orientada a objetos para modelar los datos y la conexión a base de datos
   * Creacion una interfaz web amigable y responsiva basada en boostrap 5 donde pueda ver la lista de los tutoriales
   * Realizar un adecuado uso de las excepciones.
2. **Diagrama relacional**
3. **Diagrama de clases**

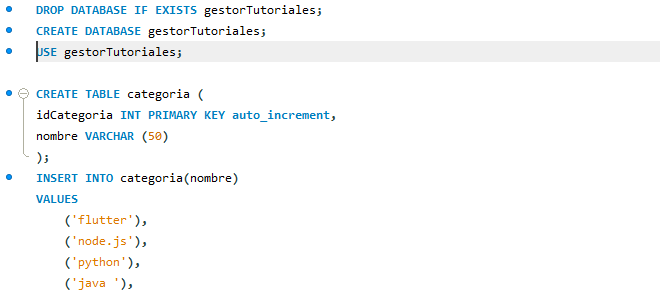


Link del repositorio:

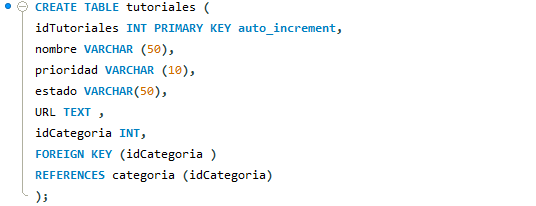
1. **Documentación**

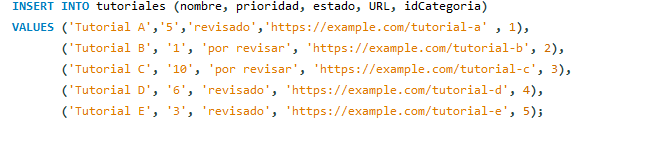
**Requerimiento funcional 1: Ingresar un tutorial y una categoría en la base de datos**

Se crean las tablas en la base de datos, las cuales son categoría y tutoriales de la siguiente manera:

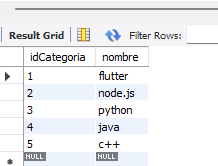
1. La tabla de categoría, con los nuevos registros que se van a ingresar, en este caso la categoría:

1.1 La tabla de tutoriales, con los registros que se ingresaran a la tabla como: el nombre del tutorial, la prioridad, el estado, la url:





1.2 Se crean las tablas, en este caso la tabla categoría, y se revisa si fueron creadas de una manera correcta. Como se puede ver, la tabla de categoría fue creada y guarda la información, con datos ingresados en la página, la cual se mostrará más adelante.

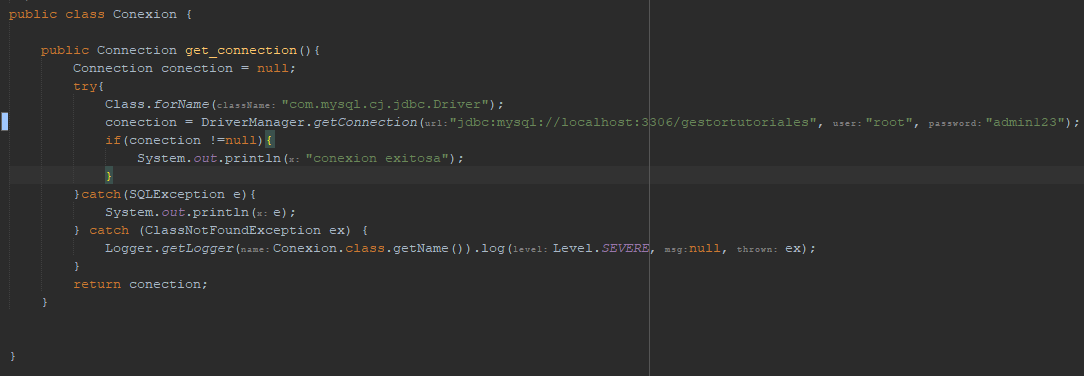


1.3 La tabla tutorial creada correctamente, con algunos datos ingresados.

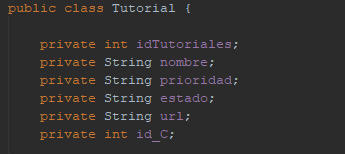
1. Se crea una página para agregar el tutorial y la tabla donde aparecerán todos los tutoriales agregados de la siguiente manera:



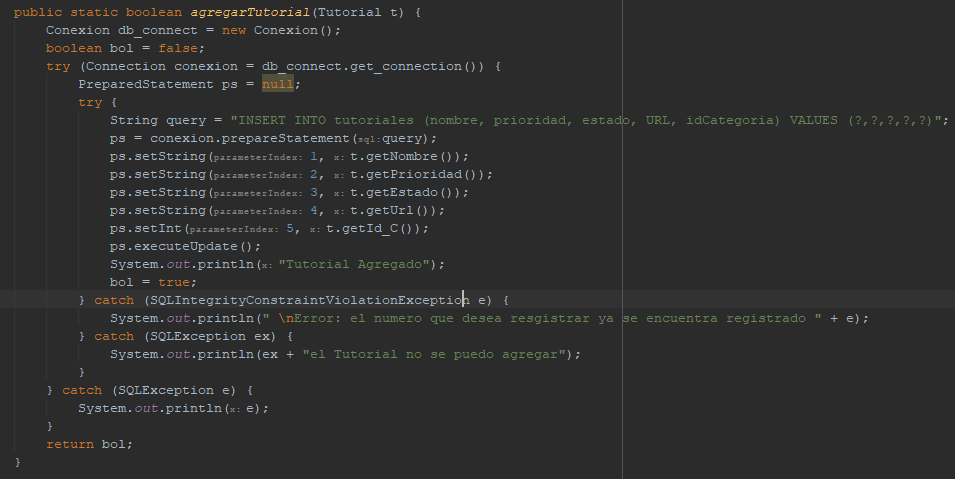
* 1. Para agregar el tutorial y este pueda ser guardado en la base de datos se crea una clase la cual se llama conexión, esta permite conectar la base de datos creada para guardar en las tablas mostradas anteriormente.



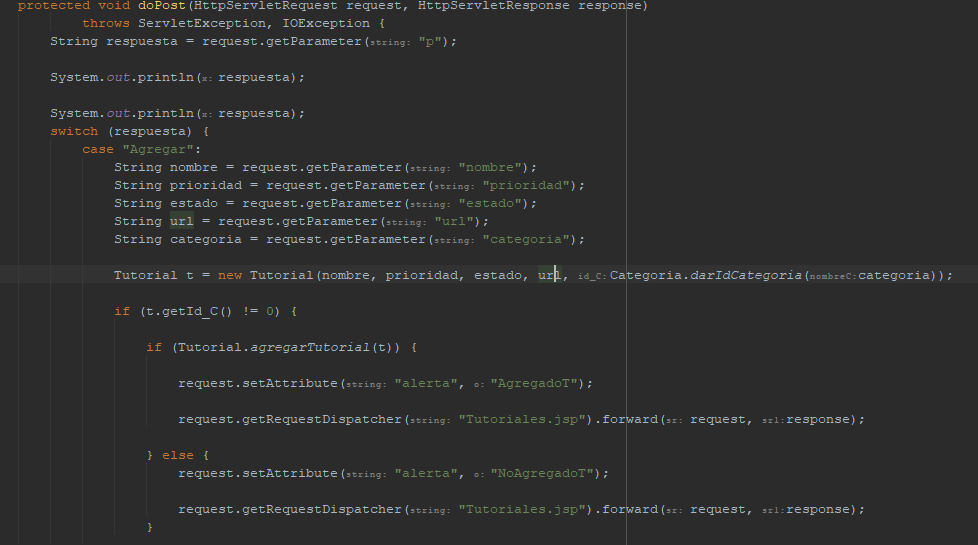
* 1. se crean una clase Tutorial para crear atributos



* 1. Se crean métodos en esta clase para agregar tutorial, editar, eliminar y listar, en este caso para agregar tutorial a la base de datos establece una conexión utilizando un objeto de la clase Conexión para preparar una consulta SQL para insertar un nuevo registro en la tabla de tutoriales, con los datos proporcionados por el objeto Tutorial pasado como parámetro, ejecuta la consulta preparada para realizar la inserción en la base de datos.



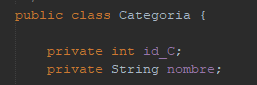
* 1. **Para agregar el tutorial**: se obtiene un parámetro, se obtienen los datos del nuevo tutorial en el formulario para asi crear un nuevo objeto Tutorial con los datos obtenidos y el ID de la categoría obtenido de la base de datos. Verifica si se ha obtenido correctamente un ID de categoría válido e intenta agregar el tutorial a la base de datos utilizando el método agregarTutorial(t) de la clase Tutorial, dependiendo del resultado de la operación de agregar, establece un mensaje de alerta adecuado en un atributo de solicitud.

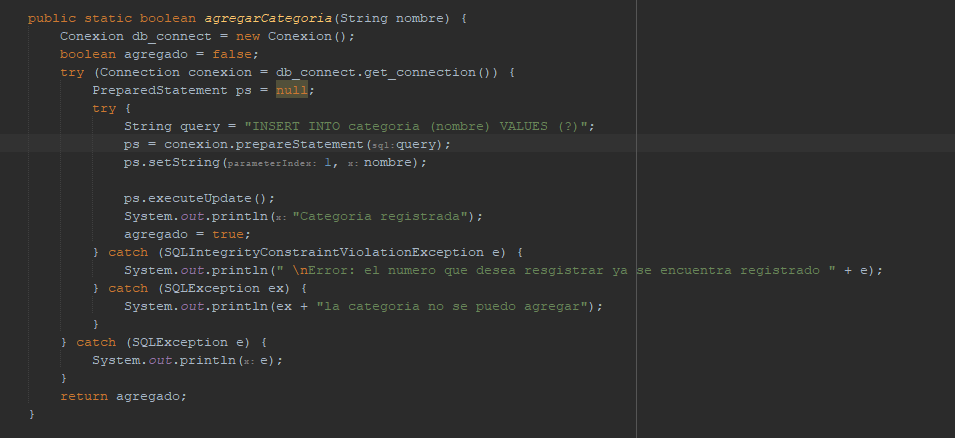
****

* 1. Se crea otra página a parte para agregar categoría.

****

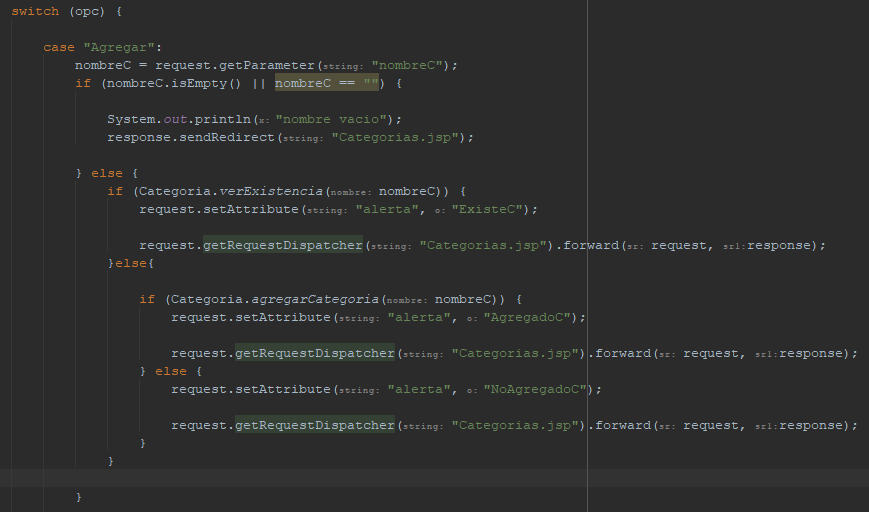
* 1. Se crea una clase llamada Categoría para agregar atributo

****

* 1. **Para agregar categoría:** se crea una clase llamada categoría en la cual se definen los atributos de nombre y el id; en esta misma clase se agrega una nueva categoría, se lista, edita y se elimina

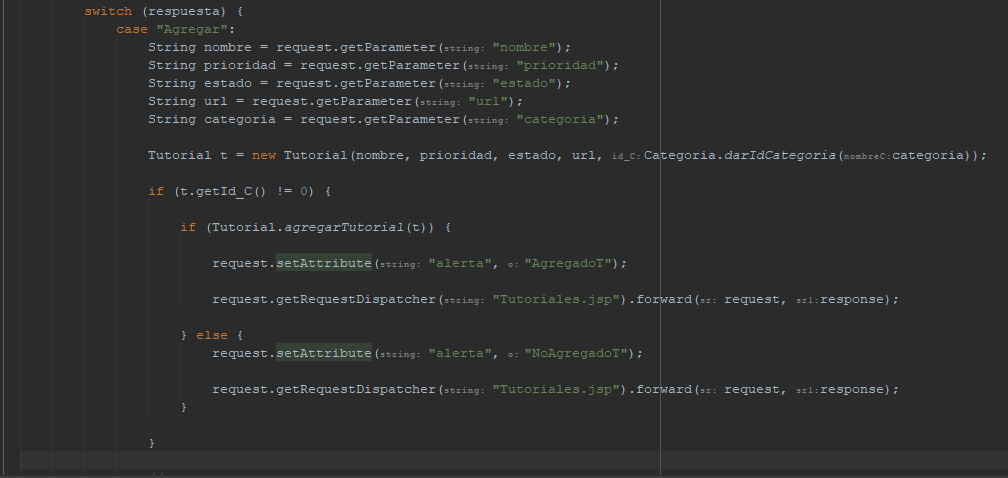
En esta clase se establece una conexión a la base de datos, la conexión se establece dentro de un bloque try-with-resources, pero para que los datos puedan guardarse en la base de datos, se define una cadena SQL que representa la consulta de inserción que se ejecutará, esta consulta insertará un nuevo registro en la tabla categoría, siendo asi ps.executeUpdate();: se ejecuta la consulta preparada para realizar la inserción en la base.

* 1. en el servlet se obtienen los parámetros del formulario

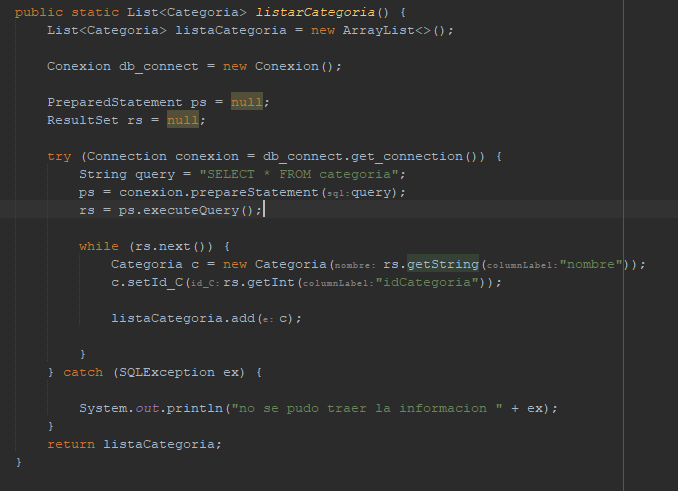


**Requerimiento funcional 2: Mostrar lista con la información de todos los tutoriales y las categorías.**

1. **Para listar el tutorial**: crea una lista vacía llamada listaTutoriales que contendrá objetos de tipo Tutorial. Establece una conexión a la base de datos utilizando un objeto de la clase Conexión, para preparar una consulta SQL y seleccionar todos los registros de la tabla tutoriales. Ejecuta la consulta preparada y almacena los resultados en un objeto ResultSet, itera sobre cada fila del resultado y crea un nuevo objeto Tutorial con los datos obtenidos de las columnas de la tabla tutoriales, establece el ID del tutorial utilizando el valor de la columna "idTutoriales" y agrega el objeto Tutorial a la lista listaTutoriales.
   1. Obtenemos los parámetros del formulario que representan los datos de un nuevo tutorial: nombre, prioridad, estado, URL y categoría ,creando un nuevo objeto Tutorial con estos datos, incluyendo el ID de la categoría obtenido a través del método darIdCategoria(categoria) de la clase Categoria, verifica si se ha obtenido correctamente un ID de categoría válido e intenta agregar el tutorial a la base de datos utilizando el método agregarTutorial(t) de la clase Tutorial, si la operación de agregado es exitosa, establece un mensaje de alerta como "AgregadoT" y redirige al usuario a la página "Tutoriales.jsp".



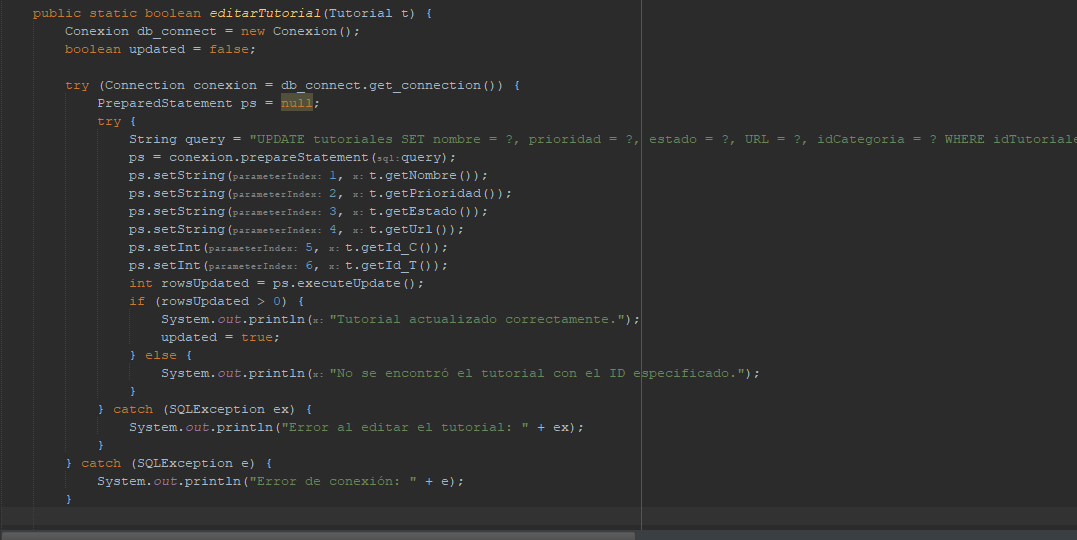
* 1. **Para listar categoría**

****

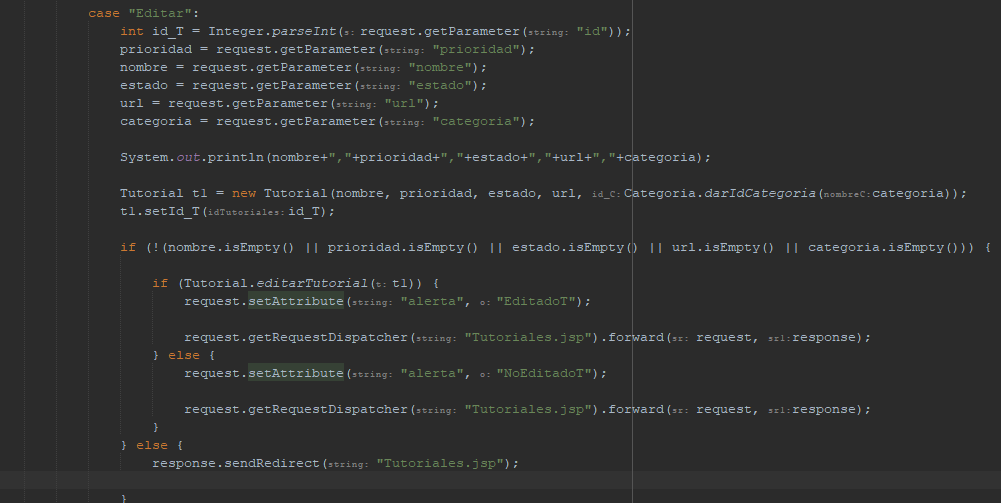
Este método se encarga de obtener y devolver una lista de todas las categorías almacenadas en una base de datos, crea una lista que almacenara los objetos de tipo categoría, estableciendo una conexión con la base de datos utilizando una instancia Ejecuta una consulta SQL para seleccionar todas las filas de la tabla categoría y agrega cada objeto Categoria a la lista listaCategoria

**Requerimiento funcional 3: Editar, eliminar y mostrar la información de todos los tutoriales y las categorías**

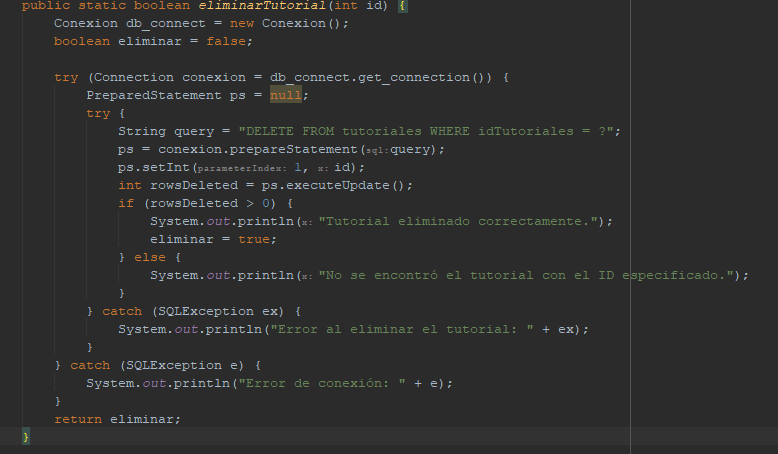
* 1. **Editar del formulario de tutorial**: actualiza la información de un tutorial en la base de datos, estableciendo una conexión a la base de datos utilizando un objeto de la clase Conexion. Prepara una consulta SQL para actualizar un registro en la tabla de tutoriales, estableciendo los valores de nombre, prioridad, estado, URL y ID de categoría según el ID del tutorial especificado, ejecutando la consulta preparada y obtiene el número de filas actualizadas.



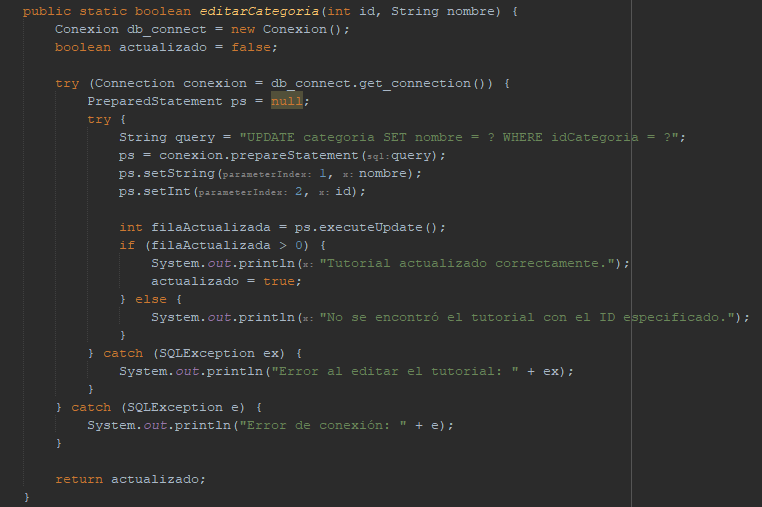
* 1. **Para el servlet** se maneja la solicitud de edición de un tutorial obteniendo los parámetros de la solicitud HTTP, como el ID del tutorial y sus detalles como prioridad, nombre, estado, URL y categoría. Luego, valida los datos y procede a editar el tutorial si todos los campos están completos. Dependiendo del resultado de la edición, se establece una alerta y se redirige o envía la solicitud de vuelta a la página JSP correspondiente para informar al usuario sobre el resultado.



**3.3 eliminar tutorial del formulario:** Se abre y se cierra la conexión a la base de datos de manera segura utilizando el bloque try-with-resources. Dentro de este bloque se realizarán las operaciones de eliminación del tutorial. Creación de la consulta SQL para eliminar el tutorial: para preparar una consulta SQL parametrizada para eliminar un tutorial de la tabla tutoriales donde el idTutoriales sea igual al valor proporcionado. Se asigna el valor del parámetro en la consulta preparada estableciendo el valor del parámetro id en la consulta preparada utilizando el método setInt.

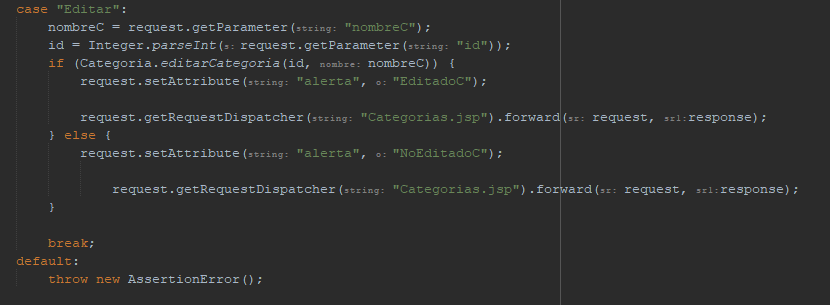
****

* 1. **Editar una categoría:** se actualiza el nombre de una categoría en una base de datos utilizando un identificador (id) y un nuevo nombre (nombre). Después de establecer una conexión a la base de datos, ejecuta una consulta SQL para actualizar el nombre de la categoría en la tabla correspondiente. Si la actualización tiene éxito, devuelve true; de lo contrario, devuelve false

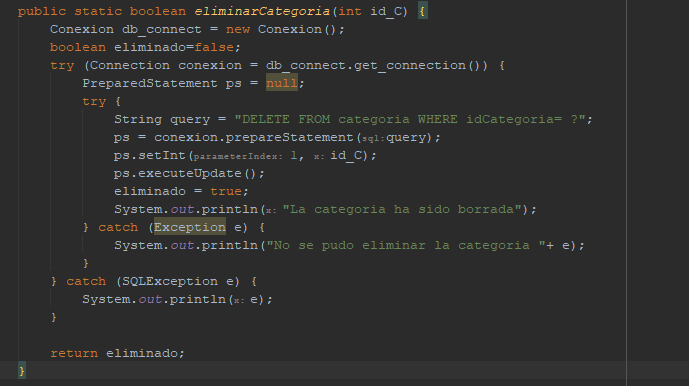
****

.

En el servlet se obtienen los parámetros de la solicitud HTTP, como el nombre de la categoría (nombreC) y su identificador (id) en la edición de la categoría se llama al método editarCategoria de la clase Categoria, pasando el ID de la categoría y el nuevo nombre, en este método probablemente actualiza el nombre de la categoría en la base de datos. Se establecen los atributos de la solicitud y redireccionamiento y dependiendo del resultado de la edición de la categoría, se establece un atributo de alerta (alerta) en la solicitud para notificar al usuario si la edición se realizó con éxito o no. Luego, se redirige la solicitud a la página Categorias.jsp para mostrar el resultado al usuario.



* 1. **eliminar categoría**

Se instancia un objeto Conexion, que contenga lógica para establecer una conexión a la base de datos inicializando una variable booleana se inicializa la variable eliminada como false. Esta variable se utilizará para indicar si se eliminó correctamente la categoría. Se Crea una consulta SQL para eliminar la categoría: Se prepara una consulta SQL parametrizada para eliminar una categoría de la tabla categoria donde el idCategoria sea igual al valor proporcionado y en la ejecución de la consulta se utiliza el método executeUpdate. Y por último la actualización de la variable eliminado: Se establece la variable eliminado en true, lo que indica que la categoría ha sido eliminada con éxito.