María José Gaytán Gil - A01706616

"Laboratorio 25: Transacciones"

Transacciones que hice en el proyecto:

```
// Método para renombrar una categoría basado en su nombre y marca
   static async renombrarCategoria(nombreMarca, nombreCategoriaActual,
nuevoNombreCategoria) {
       const connection = await db.getConnection();
       try {
           await connection.beginTransaction(); // Inicia una transacción

           // Actualiza el nombre de la categoría en la tabla de categorías
           await connection.execute(
               'UPDATE categorias SET categoria_nombre = ? WHERE nombre_marca = ?
AND categoria_nombre = ?',
               [nuevoNombreCategoria, nombreMarca, nombreCategoriaActual]
           );

           // Actualiza la categoría en la tabla de preguntas
           await connection.execute(
               'UPDATE preguntas SET Categoria = ? WHERE NombreMarca = ? AND
Categoria = ?',
               [nuevoNombreCategoria, nombreMarca, nombreCategoriaActual]
           );

           await connection.commit(); // Si todo va bien, confirma los cambios
       } catch (error) {
           await connection.rollback(); // Si hay un error, revierte los cambios
           console.log(error);
           throw new Error('Error al renombrar la categoría y actualizar la tabla
de preguntas');
       } finally {
           connection.release(); // Libera la conexión
       }
   }


   // Método para eliminar una categoría por su nombre y marca
   static async eliminarCategoriaPorNombre(nombreMarca, nombreCategoria) {
       const connection = await db.getConnection();
       try {
           await connection.beginTransaction(); // Inicia una transacción

           // Obtener IDs de preguntas asociadas a la categoría
```

```
            const [preguntas] = await connection.execute(
                'SELECT IDPreguntas FROM preguntas WHERE NombreMarca = ? AND
Categoria = ?',
                [nombreMarca, nombreCategoria]
            );

            // Eliminar opciones de las preguntas encontradas
            for (const pregunta of preguntas) {
                await connection.execute(
                    'DELETE FROM opciones_pregunta WHERE IDPreguntas = ?',
                    [pregunta.IDPreguntas]
                );
            }

            // Eliminar las preguntas asociadas a la categoría
            if (preguntas.length > 0) {
                await connection.execute(
                    'DELETE FROM preguntas WHERE NombreMarca = ? AND Categoria = ?',
                    [nombreMarca, nombreCategoria]
                );
            }

            // Finalmente, eliminar la categoría
            await connection.execute(
                'DELETE FROM categorias WHERE nombre_marca = ? AND categoria_nombre
= ?',
                [nombreMarca, nombreCategoria]
            );

            await connection.commit(); // Confirma los cambios si todo va bien
        } catch (error) {
            await connection.rollback(); // Revierte los cambios en caso de error
            console.log(error);
            throw new Error('Error al eliminar la categoría y sus preguntas y
opciones asociadas');
        } finally {
            connection.release(); // Libera la conexión
        }
    }
```