



Escuela:

Facultad de Informática

Carrera:

Licenciatura en informática

Hecho por:

Gastelum Landeros Antonio

Grupo: 2-1

Código Explicado:

Importación de módulos

```
const { DataTypes } = require('sequelize');
```

```
const sequelize = require('./conexion');
```

- **DataTypes:** lo usas para definir los tipos de datos de cada campo del modelo (como **STRING**, **INTEGER**, **DOUBLE**, etc.).
- **sequelize:** se importa la instancia de conexión a la base de datos desde el archivo **conexion.js**.

Definición del modelo Monedas

```
const Monedas = sequelize.define('monedas', {  
  id: { type: DataTypes.INTEGER, primaryKey: true },  
  origen: { type: DataTypes.STRING },  
  destino: { type: DataTypes.STRING },  
  valor: { type: DataTypes.DOUBLE }  
}, {  
  timestamps: false  
});
```

Aquí se define una tabla llamada **monedas** con las siguientes columnas:

Campo	Tipo	Descripción
id	INTEGER	Identificador único (clave primaria)
origen	STRING	Moneda de origen (ej: USD)
destino	STRING	Moneda destino (ej: EUR)
valor	DOUBLE	Tasa de cambio entre las dos

timestamps: false: indica que no se incluirán los campos createdAt y updatedAt automáticamente.

Función para agregar y editar monedas

```
async function gestionarMonedas() {
```

Esta función asincrónica se encarga de:

a) Asegurar conexión con la base de datos

```
await sequelize.sync();
```

- Verifica que la conexión esté lista y sincroniza los modelos con la base de datos (aunque no borra ni modifica tablas existentes).

b) Agregar nuevas monedas

```
await Monedas.bulkCreate([
```

```
{ id: 1, origen: 'USD', destino: 'EUR', valor: 0.93 },
```

```
{ id: 2, origen: 'USD', destino: 'JPY', valor: 154.65 },
```

```
{ id: 3, origen: 'EUR', destino: 'GBP', valor: 0.86 },
```

```
], { ignoreDuplicates: true });
```

- Inserta varios registros a la vez en la tabla.
- ignoreDuplicates: true evita errores si el id ya existe (no los vuelve a insertar si ya están).

c) Editar monedas existentes

```
await Monedas.update(
  { valor: 0.95 },
  { where: { origen: 'USD', destino: 'EUR' } }
);
```

```
await Monedas.update(
  { valor: 155.00 },
  { where: { origen: 'USD', destino: 'JPY' } }
);
```

- Actualiza el valor de la moneda específica según las condiciones dadas en where.

d) Confirmación

```
console.log('Monedas agregadas y editadas exitosamente.');
```

- Imprime un mensaje de éxito en consola.

e) Manejo de errores

```
} catch (error) {
  console.error('Error gestionando monedas:', error);
}
```

- Si ocurre algún error (de conexión, sintaxis SQL, etc.), lo muestra en consola.

Ejecución automática

```
gestionarMonedas();
```

- Se llama la función al final para que se ejecute al correr este archivo directamente.

Exportación del modelo

`module.exports = Monedas;`

- Permite que otros archivos (como `app.js`) puedan importar y usar el modelo `Monedas`.