



Escuela:

Facultad de Informática

Carrera:

Licenciatura en informática

Hecho por:

Gastelum Landeros Antonio

Grupo: 2-1

Código Explicado:

Importaciones necesarias

```
const express = require("express");
```

```
const router = express.Router();
```

```
const { Cliente, Proveedor, Articulo, Empleado } = require("../database");
```

- `express.Router()` crea un enrutador para agrupar rutas.
- Se importan los modelos desde el archivo `database.js`.

Función `crearCRUD(modelo, nombre)`

Esta función recibe:

- `modelo`: el modelo Sequelize (como `Cliente`).
- `nombre`: el nombre que se usará en la URL (como `clientes`).

Y dentro define 5 rutas para ese modelo:

Crear (POST)

```
router.post(`/${nombre}`, async (req, res) => {
```

```
  const nuevo = await modelo.create(req.body);
```

```
  res.json(nuevo);
```

```
});
```

- Crea un nuevo registro usando los datos enviados en el cuerpo de la petición.

Leer todos (GET)

```
router.get(`/${nombre}`, async (req, res) => {
```

```
  const lista = await modelo.findAll();
```

```
  res.json(lista);
```

```
});
```

- Devuelve una lista completa de todos los registros.

Leer por ID (GET)

```
router.get(`/${nombre}/:id`, async (req, res) => {  
  
  const item = await modelo.findByPk(req.params.id);  
  
  if (!item) return res.status(404).json({ mensaje: "No encontrado" });  
  
  res.json(item);  
  
});
```

- Busca un registro por su id. Si no lo encuentra, devuelve un error 404.

Actualizar (PUT)

```
router.put(`/${nombre}/:id`, async (req, res) => {  
  
  const item = await modelo.findByPk(req.params.id);  
  
  if (!item) return res.status(404).json({ mensaje: "No encontrado" });  
  
  await item.update(req.body);  
  
  res.json(item);  
  
});
```

- Busca un registro por ID, lo actualiza con los datos del req.body, y lo devuelve.

Eliminar (DELETE)

```
router.delete(`/${nombre}/:id`, async (req, res) => {  
  
  const item = await modelo.findByPk(req.params.id);  
  
  if (!item) return res.status(404).json({ mensaje: "No encontrado" });  
  
  await item.destroy();  
  
  res.json({ mensaje: `${nombre} eliminado` });  
  
});
```

```
});
```

- Busca el registro por ID, lo elimina de la base de datos y confirma la eliminación.

Generar CRUDs

```
crearCRUD(Cliente, "clientes");
```

```
crearCRUD(Proveedor, "proveedores");
```

```
crearCRUD(Articulo, "articulos");
```

```
crearCRUD(Empleado, "empleados");
```

- Llama a la función para cada modelo, generando automáticamente las 5 rutas para cada uno.

Exportar el router

```
module.exports = router;
```

- Esto permite que este router sea usado en app.js con app.use().