

Proyecto de Gestión de Sueldos (Node.js + SQLite)

Archivo: conexion.js

js

CopiarEditar

```
const sqlite3 = require('sqlite3').verbose();
```

```
const db = new sqlite3.Database('./sueldos.sqlite');
```

```
module.exports = db;
```

Explicación

- Se importa la librería sqlite3 con la opción verbose() para habilitar mensajes de depuración más detallados.
- Se crea una conexión a la base de datos sueldos.sqlite ubicada en el mismo directorio del proyecto.
- La conexión se exporta mediante module.exports para que pueda ser utilizada en otros archivos del proyecto.

Archivo: sueldos.js

js

CopiarEditar

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

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

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

```
router.get('/sueldos', (req, res) => {
```

```
  db.all("SELECT * FROM sueldos", (err, filas) => {
```

```
    if (err) {
```

```
      res.status(500).json({ error: err.message });
```

```

    } else {
        res.json(filas);
    }
});
});

```

module.exports = router;

Explicación

- Se configura un enrutador con Express para gestionar las rutas relacionadas con los sueldos.
- Se importa la conexión a la base de datos desde conexion.js.
- Se define una ruta GET en /sueldos que ejecuta una consulta SQL para obtener todos los registros de la tabla sueldos.
 - Si ocurre un error, se responde con un código 500 y el mensaje del error.
 - Si la consulta tiene éxito, se devuelven los datos en formato JSON.
- Se exporta el router para que pueda ser utilizado en el archivo principal del servidor.

Archivo: app.js

js

CopiarEditar

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

```
const cors = require('cors');
```

```
const app = express();
```

```
app.use(cors());
```

```
app.use(express.json());
```

```
app.use('/', require('./sueldos'));
```

```
app.listen(3000, () => {  
  console.log('Servidor corriendo en http://localhost:3000');  
});
```

Explicación

- Se configura el servidor principal utilizando Express.
- Se utiliza cors() para permitir solicitudes desde otros orígenes, útil cuando se trabaja con un frontend independiente.
- Se activa express.json() para que el servidor pueda interpretar cuerpos de peticiones en formato JSON.
- Se importa el módulo de rutas sueldos.js y se monta en la raíz (/).
- Se inicia el servidor en el puerto 3000 y se imprime un mensaje en la consola cuando está listo.

Funcionamiento General

1. El servidor se lanza en la dirección http://localhost:3000.
2. Cuando un cliente realiza una solicitud GET a la ruta /sueldos, el servidor consulta la base de datos y devuelve todos los registros existentes en la tabla sueldos.
3. El servidor está preparado para integrarse con interfaces web o aplicaciones frontend, gracias al soporte para JSON y CORS.