**Formulario de Boleta de Servicio**

Jennifer Rebeca García Moncada

Universidad Politécnica Internacional

Programación III

Profesor: Alexander Benjamin Curling

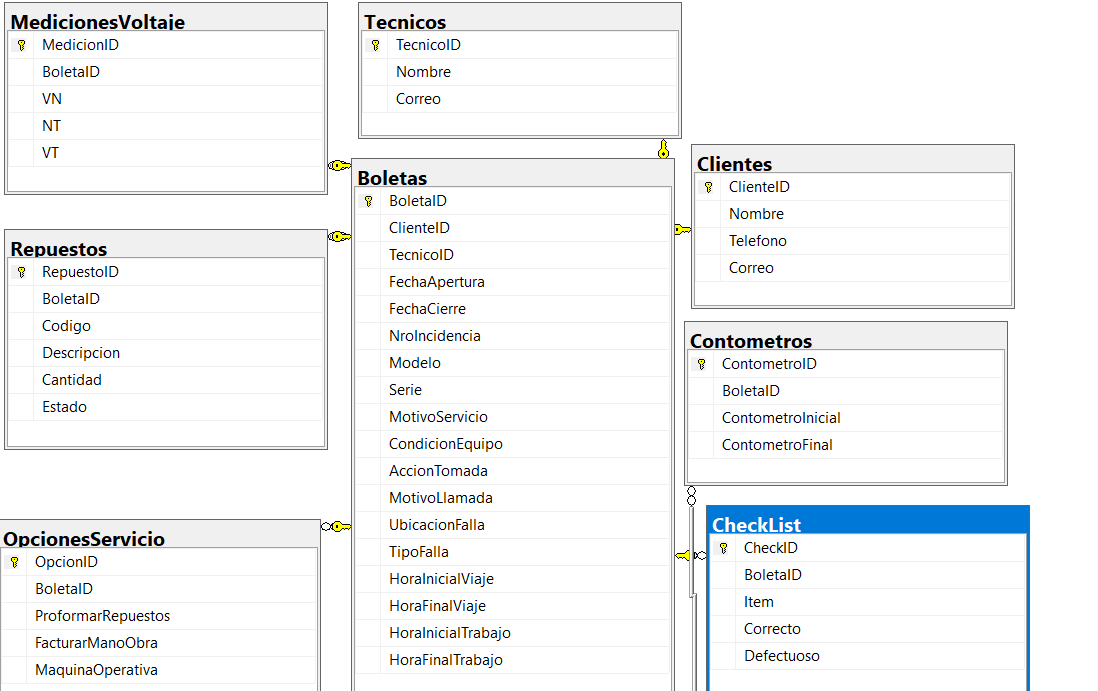
octubre, 2024

# Introducción

Este sistema de boletas de servicio se ha diseñado empleando Node.js en el backend, Express para la administración del servidor y SQL Server como base de datos principal. Su meta es facilitar el registro de datos exhaustivos acerca de las reparaciones y el mantenimiento de aparatos electrónicos. Los usuarios proporcionan información acerca del cliente, el técnico responsable, el equipo reparado, las horas laborales y otros datos, que posteriormente se guardan en una base de datos para su futura consulta.

# Estructura del Proyecto

## Tablas de SQL



## Aplicación Visual Code

El proyecto está organizado en las siguientes carpetas y archivos principales:

**app.js:** Archivo principal que gestiona las rutas y la lógica del servidor.

**connectionSQLServer.js:** Archivo que gestiona la conexión a la base de datos SQL Server.

**Paginas/Paginainicio.html:** Archivo HTML que contiene el inicio de la pagina.

**Paginas/Ficha.html:** Archivo HTML que contiene el formulario para llenar las boletas de servicio.

**Estilos/EstiloInicio.css:** Archivo CSS que contiene los estilos de la pagina de inicio.

**Estilos/EstiloFicha.css:** Archivo CSS que contiene los estilos del formulario.

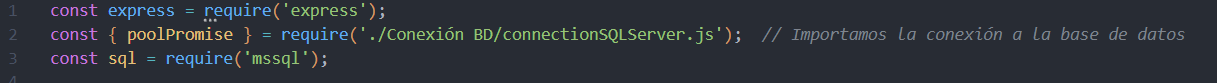
**node\_modules:** Dependencias del proyecto gestionadas con npm.

**package.json:** Archivo de configuración del proyecto Node.js y sus dependencias.

# Descripción de la funcionalidad

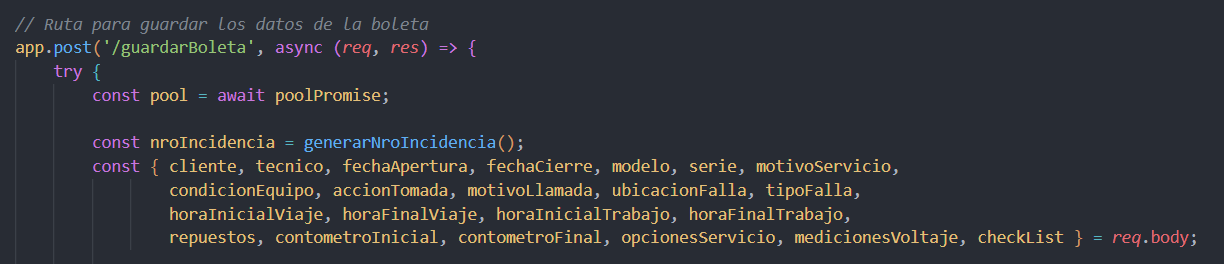
## App.js:

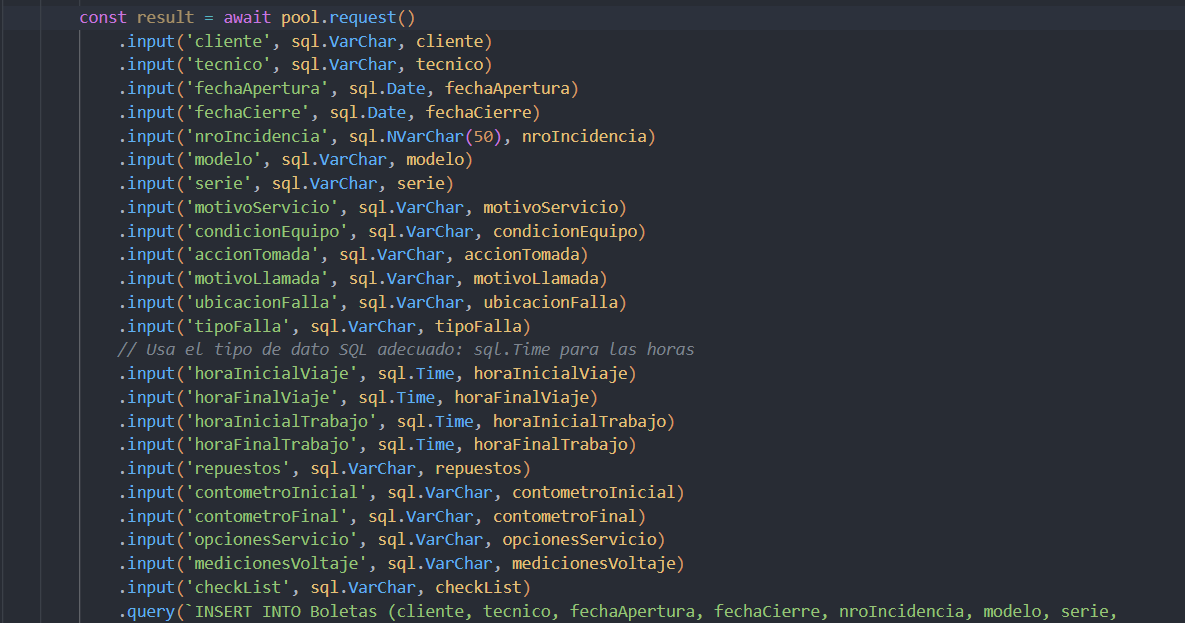
## Conexión a SQL Server:

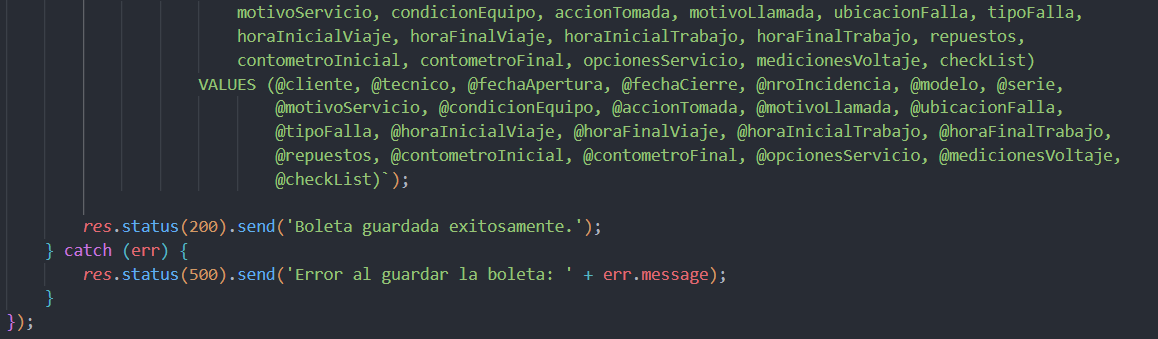


Este código configura la conexión a SQL Server a través de un pool de conexiones, lo que mejora el desempeño y la eficacia de las consultas.

## Ruta para guardar boletas:

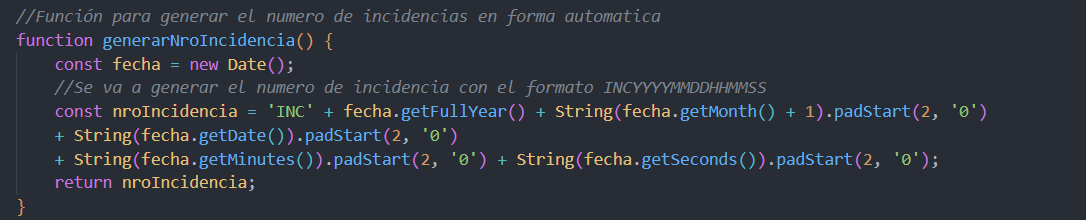






Esta ruta gestiona las peticiones POST recibidas desde el frontend, gestionando la lógica para almacenar el ticket en la base de datos SQL Server. Si se produce un fallo, se registra y se presenta un mensaje descriptivo.

## Generación de Números de Incidencia Automáticos:



Este procedimiento produce un número único de incidencia basado en la fecha y hora presentes, asegurando que cada boleta posea un identificador distintivo.

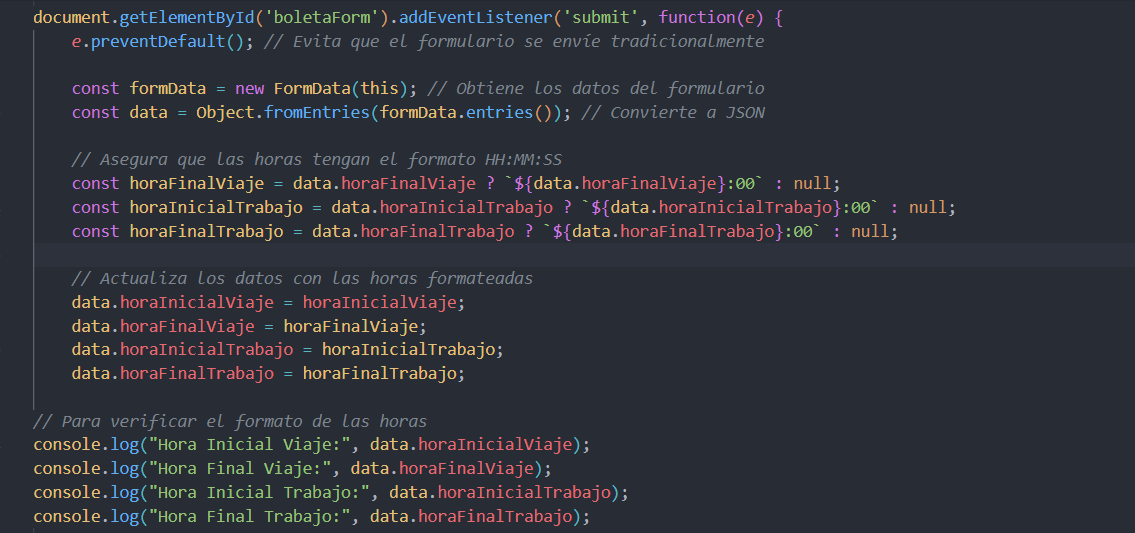
## Validaciones y Lógica en el Frontend (Ficha.html)

## Envío de Formulario con Fetch:



Este trozo de código en el frontend tiene la tarea de transmitir los datos introducidos en el formulario al backend mediante una petición AJAX a través de fetch. Si se produce un fallo, se registra y se presenta al usuario un mensaje.

## Ajuste de los Tiempos en el Frontend:



Este ajuste garantiza que las horas transmitidas al servidor se encuentren en el formato adecuado (HH:MM:SS) requerido para la base de datos SQL Server.

# Problemas Encontrados

## Error: Invalid Time en SQL Server

Al intentar guardar las horas de inicio y fin en SQL Server, el servidor lanza un error indicando que el formato del tiempo no era válido.

## Error 500: Internal Server Error

Durante las pruebas, el servidor arroja un error 500 al procesar las solicitudes POST.

# Consideraciones Finales

* Se recomienda implementar una autenticación de usuarios para que solo técnicos autorizados puedan acceder al sistema y registrar boletas.
* Se recomienda solucionar los errores presentados anteriormente.
* En futuras versiones, sería útil agregar más validaciones en el backend para garantizar la integridad de los datos, como la verificación de entradas numéricas y campos requeridos.
* Continuar investigando posibles soluciones relacionadas con el formato de tiempo en SQL Server, especialmente en el tipo de dato time y cómo se manejan en el middleware.
* Considerar la posibilidad de modificar el esquema de la base de datos para utilizar un formato de tiempo diferente (por ejemplo, datetime) o hacer ajustes en el manejo de tiempos a nivel de código.
* Revisar más a fondo los logs de error del servidor y realizar pruebas exhaustivas con distintos formatos de tiempo para identificar una solución definitiva.

# Conclusión

Se ha implementado parcialmente el sistema de boletas de servicio, consiguiendo con éxito ciertas funcionalidades, como el registro de datos específicos de las reparaciones y la vinculación con la base de datos SQL Server. No obstante, aún existen ciertos inconvenientes técnicos, particularmente en lo que respecta a la gestión de los campos de tiempo, que todavía provocan fallos de validación en el backend. Pese a los desafíos por resolver, el sistema ofrece una sólida base de desarrollo para la administración de boletas de servicio y se anticipa que, en iteraciones futuras, estos inconvenientes puedan ser solucionados para lograr una funcionalidad total.