Escuela:

Facultad de Informática

Carrera:

Licenciatura en informática

Hecho por:

Gastelum Landeros Antonio

Grupo: 2-1

Código Explicado:

Importaciones y configuración

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

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

const app = express();

const PORT = 3000;
```

- express: importa el framework web Express.
- sqlite3: permite interactuar con la base de datos SQLite.
- app: instancia de la aplicación Express.
- PORT: define el puerto donde el servidor va a escuchar (http://localhost:3000).

Middleware para leer JSON

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

Este middleware permite que el servidor pueda entender datos JSON enviados en las solicitudes POST.

Conexión con SQLite

```
const db = new sqlite3.Database('./boletos.sqlite', (err) => {
  if (err) {
    console.error('Error al conectar con la base de datos:', err.message);
  } else {
    console.log('Conectado a la base de datos boletos.sqlite');
  }
});
```

- Se conecta al archivo boletos.sqlite.

- Muestra en consola si la conexión fue exitosa o hubo error.

Ruta principal POST /boleto

app.post('/boleto', (req, res) => {

const { localidad, fecha, esEstudiante } = req.body;

- Crea una ruta que responde a peticiones POST en /boleto.
- Extrae del cuerpo JSON tres datos:
  - localidad: A, B o C
  - fecha: Viernes o Sábado
  - esEstudiante: true o false

Validación de campos

```
if (!localidad || !fecha || esEstudiante === undefined) {

  return res.status(400).json({ error: 'Faltan datos: localidad, fecha o esEstudiante.' });

}
```

- Si falta algún dato necesario, responde con error 400 (Bad Request).

Consulta a la base de datos

```
const sql = 'SELECT precio, descuento FROM precios WHERE localidad = ? AND fecha = ?';

db.get(sql, [localidad, fecha], (err, row) => {
```

- Consulta el precio y descuento de la tabla precios según la combinación localidad + fecha.
- Usa db.get() para obtener un solo resultado.

Manejo de errores o falta de datos

```
  if (err) {

    return res.status(500).json({ error: 'Error al consultar la base de datos.' });

  }
```

```
  if (!row) {
    return res.status(404).json({ error: 'No se encontró combinación de localidad y fecha.' });
  }
```

- Si ocurre un error en la consulta, devuelve 500.
- Si no se encuentra esa combinación en la base, devuelve 404.

Cálculo del total a pagar

```
  let total = row.precio;
  if (esEstudiante) {
    total = total - (total * row.descuento);
  }
```

- Asigna el precio base.
- Si esEstudiante es true, aplica el descuento multiplicando el porcentaje y restándolo del precio.

Envío de respuesta al cliente

```
  res.json({
    localidad,
    fecha,
    esEstudiante,
    precioBase: row.precio,
    descuento: esEstudiante ? (row.descuento * 100) + '%' : 'No aplica',
    totalPagar: total.toFixed(2)
  });
 });
});
```

- Responde con los datos:

- Precio base
- Descuento aplicado o "No aplica"
- Total a pagar ya calculado

Iniciar servidor

```
app.listen(PORT, () => {

  console.log(`Servidor escuchando en http://localhost:${PORT}`);

});
```

- Inicia el servidor y muestra en consola que está listo.