

# CONCIERTO

conexion.js

```
const { Sequelize } = require('sequelize')

//Conexion con la base de datos
const sequelize = new Sequelize({
  dialect: 'sqlite',
  storage: './Concierto.sqlite3'
})

module.exports = sequelize
```

Utiliza Sequelize, que es un ORM (Object-Relational Mapping) para Node.js

Configura una conexión a una base de datos SQLite

La base de datos se almacena en un archivo llamado 'Concierto.sqlite3'

Exporta la conexión para que pueda ser utilizada en otros archivos

modelo.js

```
const { DataTypes } = require('sequelize')
const sequelize = require('../conexion')

//Definicion del modelo de la bd
const Boletos = sequelize.define('Boletos', {
  localidad: {
    type: DataTypes.STRING,
    allowNull: false
  },
  fecha: {
    type: DataTypes.STRING,
    allowNull: false
  },
  precio: {
    type: DataTypes.FLOAT,
```

```

        allowNull: false
      },
      descuento: {
        type: DataTypes.FLOAT,
        allowNull: false
      }
    }, {
      tableName: 'Boletos',
      timestamps: false
    })

module.exports = Boletos

```

Define una tabla llamada 'Boletos' con los siguientes campos:

- localidad: String (texto) obligatorio que indica el lugar del concierto
- fecha: String (texto) obligatorio para la fecha del evento
- precio: Número decimal (float) obligatorio para el precio del boleto
- descuento: Número decimal (float) obligatorio para el porcentaje de descuento
- No incluye timestamps (created\_at, updated\_at)

app.js

```

const express = require('express')
const bodyParser = require('body-parser')
const Boletos = require('./models/modelo') //Importar el modelo Boletos
const app = express()
const puerto = 3000

app.use(bodyParser.json())

app.listen(puerto, () => {
  console.log('Servicio iniciado')
})

app.post('/calculartotal', async (req, res) => {
  const { localidad, fecha, esEstudiante } = req.body;

  //Buscar el boleto segun la localidad y fecha
  const boletos = await Boletos.findOne({
    where: { localidad, fecha }
  })

```

```

    })

    if (!boletos) {
        return res.status(404).json({ error: 'Boleto no encontrado' })
    }

    //Calcular el precio final
    const { precio, descuento } = boletos;
    const descuentoAplicado = esEstudiante ? precio * (descuento / 100)
: 0;

    const total = precio - descuentoAplicado;

    res.json({ localidad, fecha, precio, descuentoAplicado, total })
})

```

- Utiliza Express.js como framework web
- Configura el servidor para escuchar en el puerto 3000
- Define una ruta POST '/calculartotal' que:
- Recibe datos del boleto (localidad, fecha y si es estudiante)
- Busca el boleto en la base de datos según la localidad y fecha
- Calcula el precio final aplicando descuento si el comprador es estudiante
- Devuelve un JSON con toda la información del boleto y el cálculo
- La estructura del proyecto es la siguiente:
- El frontend envía una solicitud con los datos del boleto
- El servidor busca el boleto en la base de datos
- Si encuentra el boleto, calcula el precio final considerando si es estudiante
- Devuelve toda la información calculada al cliente
- Este es un sistema de venta de boletos que:
- Maneja diferentes localidades
- Tiene precios variables según la localidad
- Aplica descuentos para estudiantes
- Utiliza una base de datos SQLite para almacenar la información
- Proporciona una API REST para calcular los precios totales
-