BOLETO

Estructura del Proyecto:

Boleto/

├── public/

│ └── index.html ------- Interfaz de usuario

├── index.js --------Servidor backend

└── package.json -------- Configuración del proyecto

Archivo Package.json:

```
{
    "name": "boleto",
    "version": "1.0.0",
    "description": "Calculadora de boletos",
    "main": "index.js",
    "scripts": {
        "start": "node index.js",
        "dev": "nodemon index.js"
    },
    "author": "",
    "license": "ISC",
    "dependencies": {
        "express": "^4.18.2"
    },
    "devDependencies": {
        "nodemon": "^3.0.0"
    }
}
```

name: define el nombre del proyecto para el uso del npm
versión: sistema de versionado (1.0.0 = primera versión estable)

SCRIPTS:
Start: comando para iniciar en producción (node normal)
Dev: comando para el desarrollo (usa nodemon que reinicia automáticamente)

Dependencies:
Express: framework que necesitaremos para crear el servidor web

devDependencies:

Nodemon: Herramienta que usaremos solo en desarrollo para no reinicia manualmente

INDEX.JS

Configuracion inicial

```
const express = require('express');
const app = express();
const PORT = 3000;

app.use(express.json());
app.use(express.static('public'));
```

require('express'): Importa el framework Express

app = express(): Crea una nueva aplicación Express

PORT = 3000: Define el puerto donde correrá el servidor

express.json(): Permite recibir datos en formato JSON

express.static('public'): Sirve los archivos HTML, CSS y JS desde la carpeta public

Precios y configuración

```
const precios = {
    A: 300,
    B: 490,
    C: 670,
    D: 899
```

```
};
```

Define los precios base de cada sección

Estructura de objeto para acceso rápido

Facilita cambiar precios sin modificar la lógica

Funcion de Calculo

```
function calcularPrecio(seccion, cantidad, dia) {
    const precioUnitario = precios[seccion.toUpperCase()];

    if (!precioUnitario) {
        return { error: 'Sección inválida.' };
    }

    const diasValidos = ['viernes', 'sabado', 'domingo'];
    if (!diasValidos.includes(dia.toLowerCase())) {
        return { error: 'Día inválido.' };
    }

    let total = precioUnitario * cantidad;

    if (dia.toLowerCase() === 'domingo') {
        total *= 0.84; // 16% descuento
    }

    if (cantidad > 1) {
        total *= 0.95; // 5% descuento
    }

    return {
        seccion,
        cantidad,
        dia,
        total: total.toFixed(2)
    };
}
```

- Recibe sección, cantidad y día
- Obtiene el precio base de la sección
- Valida que la sección exista
- Valida que el día sea válido
- Calcula el precio base (precio × cantidad)
- Aplica descuento del domingo si aplica (16%)
- Aplica descuento por cantidad si aplica (5%)
- Retorna objeto con todos los detalles

Endpoint Del Servidor

```
app.post('/precio-boleto', (req, res) => {
    const { seccion, cantidad, dia } = req.body;

    if (!seccion || !cantidad || !dia) {
        return res.status(400).json({
            error: 'Faltan parámetros: sección, cantidad o día.'
        });
    }

    const resultado = calcularPrecio(seccion, cantidad, dia);

    if (resultado.error) {
        return res.status(400).json({ error: resultado.error });
    }

    res.json(resultado);
});
```

app.post: Crea una ruta que recibe peticiones POST

req.body: Extrae los datos enviados por el cliente

Validación de datos obligatorios

Llama a la función de cálculo

Maneja errores si los hay

Envía respuesta al cliente

INDEX.HTML

HTML Base

```html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Calculadora de Boletos</title>
```

Define el documento como HTML5

Establece el español como idioma

Configura codificación UTF-8 para caracteres especiales

Estilo Css

```css
body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    padding: 40px;
    background-color: #0a0a0a;
    color: #f0f0f0;
}

form {
    background: #1a1a1a;
    padding: 20px;
    border-radius: 12px;
    box-shadow: 0 0 15px rgba(200, 0, 0, 0.3);
    max-width: 400px;
    margin: auto;
}
```

Body:

Define la fuente principal

Establece el fondo oscuro

Configura el color del texto

Form:

Crea un contenedor centrado

Agrega sombra roja

Establece bordes redondeados

Limita el ancho máximo

Formulario Html

```
<form id="boletoForm">
    <h1>Calculadora de Boletos</h1>

    <label for="seccion">Sección:</label>
    <select id="seccion" required>
        <option value="">Selecciona una sección</option>
        <option value="A">A</option>
        <option value="B">B</option>
        <option value="C">C</option>
        <option value="D">D</option>
    </select>
```

Form:

Contenedor principal para los datos

Id=”boletoForm”: Identificador Para JavaScript

Required: Hace obligatorio el Campo

Option: cada opción Disponible

JavaScript Del Fronted

```
const form = document.getElementById('boletoForm');

form.addEventListener('submit', async (e) => {
    e.preventDefault();
```

```
const seccion = document.getElementById('seccion').value;
const cantidad = parseInt(document.getElementById('cantidad').value);
const dia = document.getElementById('dia').value;

const respuesta = await fetch('/precio-boleto', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ seccion, cantidad, dia })
});
```

- Obtiene Referencia al Formulario
- Escucha el Evento de Envió
- Previene el envió normal del formulario
- Recolecta los valores d los campos
- Convierte la cantidad a Numero
- Envía Datos al servidor Usando Fetch
- Espera la Respuesta
- Muestra el Resultado o Error

Manejo De Respuesta

```
const data = await respuesta.json();
 const resultadoDiv = document.getElementById('resultado');

 if (respuesta.ok) {
```

```
    resultadoDiv.textContent = `Total a pagar: $${data.total}`;
    resultadoDiv.style.color = "#00ff00";
} else {
    resultadoDiv.textContent = `Error: ${data.error}`;
    resultadoDiv.style.color = "#ff0000";
}
```

- Convierte La Respuesta a JSON
- Obtiene el Elemento Para Mostrar El Resultado
- En caso de estar todo bien, Muestra el total en verde
- Si Surge un error, muestra el Error en Rojo