

Este código está escrito en Node.js utilizando el framework Express para crear una API que calcula el precio total de boletos según la sección, el día del evento y la cantidad de boletos solicitados, aplicando descuentos según las reglas especificadas.

1. Importación y Configuración Inicial

```
const express = require('express');  
const app = express();  
app.use(express.json());
```

`require('express')`: Se importa el módulo Express, que facilita la creación de servidores web y APIs en Node.js.

`const app = express()`: Se crea una instancia de la aplicación Express.

`app.use(express.json())`: Se configura la aplicación para que pueda interpretar el cuerpo de las solicitudes HTTP que vienen en formato JSON. Esto es necesario para poder leer los datos enviados por el cliente.

2. Definición del Puerto

```
const PORT = 3000;
```

Se define el puerto donde correrá la API, en este caso el puerto 3000.

3. Definición de Constantes: Precios y Descuentos

```
const SECTION_PRICES = {  
  A: 300,  
  B: 490,  
  C: 670,  
  D: 899  
};
```

```
const MULTIPLE_TICKET_DISCOUNT = 0.05; // 5%
```

```
const SUNDAY_DISCOUNT = 0.16; // 16%
```

SECTION_PRICES: Objeto que mapea las secciones del evento a sus precios correspondientes.

MULTIPLE_TICKET_DISCOUNT: Porcentaje de descuento (5%) aplicado si se compran más de un boleto.

SUNDAY_DISCOUNT: Porcentaje de descuento (16%) aplicado por boleto comprado para el día domingo.

4. Definición de la Ruta POST /api/obtener_precio

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

Se crea un endpoint de tipo POST con la ruta /api/obtener_precio.

Se obtienen del cuerpo de la solicitud (req.body) los datos enviados por el usuario: seccion, cantidad y dia.

5. Validación de Datos Recibidos

```
if (!seccion || !cantidad || !dia) {  
  return res.status(400).json({ error: 'Faltan campos obligatorios.' });  
}
```

Se verifica que los tres campos obligatorios estén presentes.

Si falta alguno, la API responde con un error 400 (Bad Request) y un mensaje indicando que faltan campos.

6. Normalización y Validación de Valores

```
const seccionMayus = seccion.toUpperCase();  
const diaLower = dia.toLowerCase();  
  
if (!SECTION_PRICES[seccionMayus]) {  
  return res.status(400).json({ error: 'Sección inválida.' });  
}
```

```
if (!['viernes', 'sábado', 'domingo'].includes(diaLower)) {  
  return res.status(400).json({ error: 'Día inválido. Debe ser viernes, sábado o domingo.' });  
}
```

Se convierte la sección a mayúsculas para que la comparación sea consistente.

Se convierte el día a minúsculas para evitar problemas con mayúsculas o tildes.

Se valida que la sección exista en el objeto SECTION_PRICES.

Se valida que el día sea uno de los permitidos: viernes, sábado o domingo.

Si alguna validación falla, se responde con un error 400 y mensaje descriptivo.

7. Cálculo del Precio Total

```
let precioBase = SECTION_PRICES[seccionMayus];
```

```
let precioTotal = precioBase * cantidad;
```

```
if (diaLower === 'domingo') {  
    precioTotal *= (1 - SUNDAY_DISCOUNT);  
}
```

```
if (cantidad > 1) {  
    precioTotal *= (1 - MULTIPLE_TICKET_DISCOUNT);  
}
```

Se obtiene el precio base de un boleto según la sección.

Se calcula el precio total multiplicando el precio base por la cantidad de boletos.

Si el día es domingo, se aplica un descuento del 16% al total.

Si se compran más de un boleto, se aplica un descuento adicional del 5% al total.

Los descuentos se aplican en el orden indicado: primero el descuento del domingo, luego el descuento por cantidad.

8. Respuesta de la API

```
res.json({  
  seccion: seccionMayus,  
  dia: diaLower,  
  cantidad,  
  precio_total: Number(precioTotal.toFixed(2))  
});  
});
```

Finalmente, la API responde con un JSON que contiene:

La sección en mayúsculas.

El día en minúsculas.

La cantidad de boletos.

El precio total calculado, redondeado a dos decimales.

9. Inicio del Servidor

```
app.listen(PORT, () => {  
  console.log(`API corriendo en http://localhost:${PORT}`);  
});
```

El servidor se pone a escuchar en el puerto definido (3000).

Cuando está listo, imprime un mensaje en la consola indicando que la API está corriendo.

Resumen

Esta API permite calcular el precio total de boletos para un evento que ocurre viernes, sábado y domingo, con diferentes precios según la sección, y aplica descuentos en caso de compra múltiple o para boletos del domingo. La estructura del código es sencilla y modular, validando correctamente la entrada del usuario para evitar errores y asegurando respuestas claras.