



## Universidad

Universidad Autónoma de Sinaloa

## Carrera

Lic. en Informática

## Materia

Desarrollo web del lado del servidor

## Actividad

Crear API con método POST

## Grupo

2-3

## Fecha

08/06/2025 Culiacán, Sinaloa

## Maestro

José Manuel Cazarez Alderete

## Alumna

Núñez Sarabia Jessica Anahí



# ÍNDICE

**03**

JUSTIFICACIÓN

**05**

EVIDENCIA

# JUSTIFICACIÓN

El propósito de esta API es calcular el costo de boletos según la sección seleccionada y aplicar descuentos cuando corresponda. Se implementan reglas específicas:

- El precio cambia dependiendo de la sección.
- Si el boleto es para domingo, tiene un 16% de descuento.
- Si se compran más de un boleto, se aplica un 5% de descuento al total.

Se importa Express.js, que permite manejar solicitudes HTTP.

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

Se crea la aplicación Express y se define el puerto 3000.

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

Se habilita la lectura de JSON en las solicitudes POST.

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

Se crea un objeto con los precios de cada sección.

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

La función calcularPrecio() recibe la sección, cantidad de boletos y el día, aplicando los descuentos según las condiciones establecidas.

```
const calcularPrecio = (seccion, cantidad, dia) => {  
  if (!precios[seccion] || cantidad <= 0 || !dia) {  
    throw new Error('Sección inválida, cantidad no válida o día no especificado');  
  }  
};
```

Cálculo del precio base

```
let precioBase = precios[seccion] * cantidad;
```

---

### Definición de descuentos

```
const descuentos = {  
  cantidad: cantidad > 1 ? 0.95 : 1,  
  dia: dia.toLowerCase() === 'domingo' ? 0.84 : 1  
};
```

---

### Aplicación de descuentos

```
precioBase *= descuentos.cantidad * descuentos.dia;
```

---

### Retorno del resultado estructurado

```
return {  
  seccion,  
  cantidad,  
  dia,  
  precio_total: `$$${precioBase.toFixed(2)}`  
};  
};
```

---

Se agrega un middleware para capturar errores y enviar respuestas claras.

```
app.use((err, req, res, next) => {  
  res.status(400).json({ error: err.message });  
});
```

---

Se define la ruta POST /precio, que recibe los datos y usa la función calcularPrecio().

```
app.post('/precio', (req, res, next) => {  
  try {  
    const { seccion, cantidad, dia } = req.body;  
  
    if (!seccion || !cantidad || !dia) {  
      throw new Error('Faltan datos necesarios (sección, cantidad, día)');  
    }  
  
    const resultado = calcularPrecio(seccion, cantidad, dia);  
    res.json(resultado);  
  } catch (error) {  
    next(error);  
  }  
});
```

# EVIDENCIA

