## DIGITAL WARE

Consumo del api del back desde postman

Todos los metodos POST, PUT y DELETE dan un mensaje de respuesta exitoso o no como estos:

{"success":true, "message": "Datos insertados correctamente"}

{"success":false, "error": "Ha ocurrido un error al insertar el registro"}

## CATEGORIA

**GET**

### https://localhost:{puerto}/api/categoria/

No necesita ningún parámetro, retorna las categorías registradas en la base de datos

**POST**

### https://localhost:{puerto}/api/categoria/

Enviar como body raw con formato “JSON(application/json)” el siguiente objeto para realizar la inserción en la base de datos

```
{
        "CategoryName": "Lacteos",
        "Description": "Productos a base de leche"
}
```

## FACTURA

**GET**

### https://localhost:{puerto}/api/ Factura/

No necesita ningún parámetro, retorna las facturas registradas en la base de datos

**DELETE**

### https://localhost:{puerto}/api/ Factura/{id}

No necesita ningún parámetro de cuerpo, solo el id de la factura que se eliminara en la URL, este método también elimina el detalle de la factura donde se almacenan los productos asociados a esa factura, retorna las facturas registradas en la base de datos

**POST**

## https://localhost:{puerto}/api/ Factura/

Enviar como body raw con formato “JSON(application/json)” el siguiente objeto para hacer la inserción en la base de datos

```
{
	"idCliente": "3",
	"idTipoPago": "1",
	"detalleFacturaList": [
		{
			"cantidad":"6",
			"precio":"5000",
			"IdProducto":"1"
		},
		{
			"cantidad":"3",
			"precio":"5000",
			"IdProducto":"1"
		}
	]
}
```

## PRODUCTO

**GET**

## https://localhost:{puerto}/api/ Producto/

No necesita ningún parámetro, retorna los productos registrados en la base de datos

**PUT**

## https://localhost:{puerto}/api/ Producto/{id}

Enviar como body raw con formato “JSON(application/json)” el siguiente objeto para hacer la inserción en la base de datos

```
{
	"Nombre": "Pepsi 200ml",
	"Precio": 2000,
	"Stock": "59",
	"IdCategoria": 2
}
```

**DELETE**

## https://localhost:{puerto}/api/ Producto/{id}

No necesita ningún parámetro de cuerpo, solo el id del producto que se eliminara en la URL

**POST**

## https://localhost:{puerto}/api/ Producto/

Enviar como body raw con formato “JSON(application/json)” el siguiente objeto para hacer la inserción en la base de datos

```
{
        "Nombre": "Pepsi 200ml",
        "Precio": 2000,
        "Stock": "59",
        "IdCategoria": 2,
}
```

**Prueba técnica punto 3**

¿Cómo se podría asegurar una alta disponibilidad de los recursos expuestos en el API REST?

Se podría manejar métodos asincrónicos mejorando la latencia de respuesta en colas de peticiones, por otro lado, se podría realizar el desarrollo del api para que fuera escalable tanto vertical como horizontalmente, así si en un ponto el api no puede soportar todas las peticiones bastara con subir otro api al público gestionando que api está más libre en tráfico para enviar la petición saliente a dicha api poco concurrido.

¿Qué capas incluiría para soportar operaciones asíncronas?

Aplicaría las capas del patrón de diseño Command, que este patrón de diseño es provee toda la arquitectura para ejecutare peticiones asincrónicas mediante comandos en código.

¿Qué otras tecnologías manejarían o cambiaría dentro de su implementación?

Manejaria Swagger para la documentación del api, pues ahorra mucho tiempo en documentación, también JWT para realizar la autenticación por tokens y garantizar la seguridad de las peticiones realizadas al api