

Microservicio SOAP para Gestión de Infografías



Nombre: Alejandro Miguel Pardo Romano

Docente: Miguel Ángel Pacheco

Materia: Tecnologías Web II

Gestión: 1-2024

Introducción

Este proyecto tiene como objetivo la creación de un **microservicio SOAP** que se integra con el **backend de Infografías**, utilizando la API de backend de Infografías para obtener información sobre las infografías y procesar estadísticas a través de un servicio web SOAP. Se utiliza **Express** como servidor, **SOAP** para la comunicación y **MongoDB** como base de datos para almacenar las infografías.

Tecnologías Utilizadas

- **Node.js**: Plataforma de JavaScript del lado del servidor.
- **Express**: Framework para crear aplicaciones web y APIs en Node.js.
- **SOAP**: Protocolo de comunicación para la transmisión de mensajes estructurados.
- **MongoDB**: Base de datos NoSQL utilizada para almacenar las infografías.
- **Postman**: Herramienta para realizar y probar las solicitudes HTTP.
- **Axios**: Librería para realizar solicitudes HTTP en Node.js.

Detalles sobre la Base de Datos

El backend de **Infografías** utiliza **MongoDB** para almacenar los datos de las infografías. MongoDB es una base de datos NoSQL, lo que significa que no requiere un esquema fijo, lo que la hace ideal para manejar datos semiestructurados como los de las infografías.

Diseño de la Base de Datos

- **Colección **infografias****: Esta colección almacena documentos que contienen información sobre cada infografía, como el título, el texto y las imágenes asociadas.
- **Campos típicos**:
 - **titulo**: El título de la infografía.
 - **descripcion**: Descripción de la infografía.
 - **contenido**: Contenido textual de la infografía.
 - **imagenUrl**: URL de la imagen asociada a la infografía.

MongoDB permite una gestión flexible de los datos, lo que facilita la expansión del modelo a medida que las necesidades del proyecto crecen.

Objetivo del Microservicio

El microservicio tiene como objetivo consumir la API REST del backend de Infografías para obtener estadísticas sobre las infografías, como el número total de infografías, la longitud promedio de las infografías, entre otras métricas. A través del servicio web SOAP, el microservicio expondrá estas estadísticas a los clientes.

Beneficios de Usar SOAP

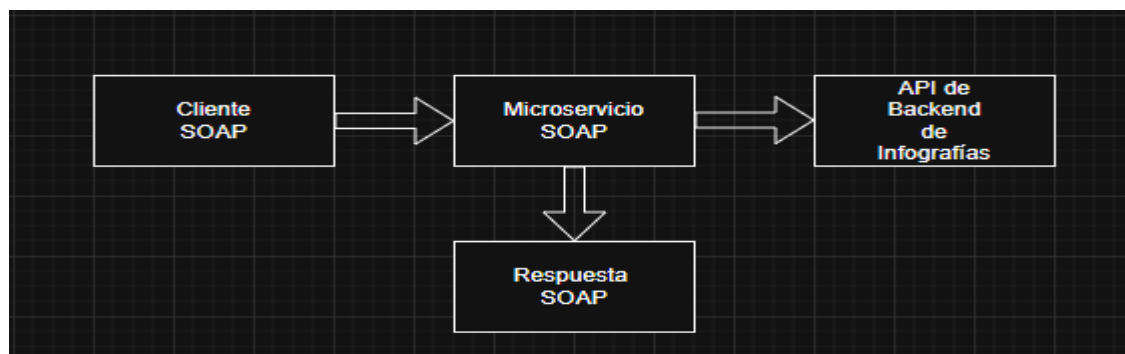
SOAP (Simple Object Access Protocol) es un protocolo de comunicación basado en XML utilizado para intercambiar datos entre aplicaciones. En este proyecto, elegimos SOAP por varias razones clave:

1. **Seguridad:** SOAP ofrece capacidades de seguridad adicionales, como WS-Security, que proporcionan integridad de los mensajes y autenticación. Esto es ideal para aplicaciones que manejan datos sensibles y requieren transacciones seguras.
2. **Estandarización:** SOAP es un protocolo altamente estandarizado, lo que lo hace ideal para integraciones complejas, especialmente cuando se necesita asegurar que las aplicaciones sean interoperables en diferentes plataformas.
3. **Robustez:** A diferencia de REST, SOAP proporciona una mayor capacidad de control sobre el proceso de transmisión de mensajes, lo que lo hace adecuado para sistemas empresariales que requieren transacciones estrictas.

En resumen, elegimos SOAP porque proporciona un enfoque más robusto y seguro para la comunicación entre servicios que otras opciones como REST o GraphQL, lo cual es importante en aplicaciones críticas.

Diagrama de Integración

A continuación, se muestra el diagrama de flujo del sistema que describe cómo interactúan los diferentes componentes del microservicio.



Descripción del Flujo

1. Cliente SOAP:

- El cliente realiza una solicitud POST a un servicio SOAP en el microservicio.

2. Microservicio SOAP:

- El microservicio recibe la solicitud SOAP y, usando **Axios**, realiza una solicitud GET a la API del backend de Infografías para obtener los datos de las infografías.

3. API de Backend de Infografías:

- El backend de Infografías responde con los datos de las infografías almacenadas en la base de datos MongoDB.

4. Respuesta SOAP:

- El microservicio SOAP devuelve los datos procesados al cliente en formato SOAP.

Implementación

Paso 1: Crear la API Express

- **Dependencias:** Instalamos las dependencias necesarias para Express y la conexión con la base de datos MongoDB.

```
bash
```

Copiar

```
npm install express axios cors mongoose
```

1. **Conexión con MongoDB:** Configuramos la conexión a MongoDB para obtener los datos de las infografías.

```
server.js > app.get('/infografias') callback
1  const express = require('express');
2  const mongoose = require('mongoose');
3  const cors = require('cors');
4  const app = express();
5  app.use(cors()); // Para habilitar solicitudes desde otros dominios
6
7  const port = 3001;
8
9  // Conexión a MongoDB
10 mongoose.connect('mongodb://localhost:27017/infografias', {
11 })
12 .then(() => console.log('Conectado a MongoDB - API de Infografías'))
13 .catch((err) => console.error('Error conectando a MongoDB:', err));
14
15 // Definir un esquema para las infografías
16 const infografiaSchema = new mongoose.Schema({
17   titulo: String,
18   descripcion: String,
19   imagen: String,
20   texto: String,
21   fechaCreacion: { type: Date, default: Date.now }
22 });
23
24 const Infografia = mongoose.model('Infografia', infografiaSchema);
25
26 // Endpoint para obtener todas las infografías
27 app.get('/infografias', async (req, res) => {
28   try {
29     const infografias = await Infografia.find();
30     res.json(infografias);
31   } catch (err) {
32     console.error('Error al obtener infografías:', err);
33     res.status(500).send('Error en el servidor');
34   }
35 });
36
37 app.listen(port, () => {
38   console.log(`API ejecutándose en http://localhost:${port}`);
39 });
40
```

2. **Endpoint para obtener infografías:** Creamos un endpoint `/infografias` que devuelve las infografías almacenadas en la base de datos MongoDB.

```
26 // Endpoint para obtener todas las infografías
27 app.get('/infografias', async (req, res) => {
28   try {
29     const infografias = await Infografia.find();
30     res.json(infografias);
31   } catch (err) {
32     console.error('Error al obtener infografías:', err);
33     res.status(500).send('Error en el servidor');
34   }
35 });
36
37 app.listen(port, () => {
38   console.log(`API ejecutándose en http://localhost:${port}`);
39 });
40
```

Paso 2: Configuración del Servicio WSDL en Express

- **Instalación de SOAP:** Instalamos la librería `soap` para configurar el servicio web SOAP.

Copiar

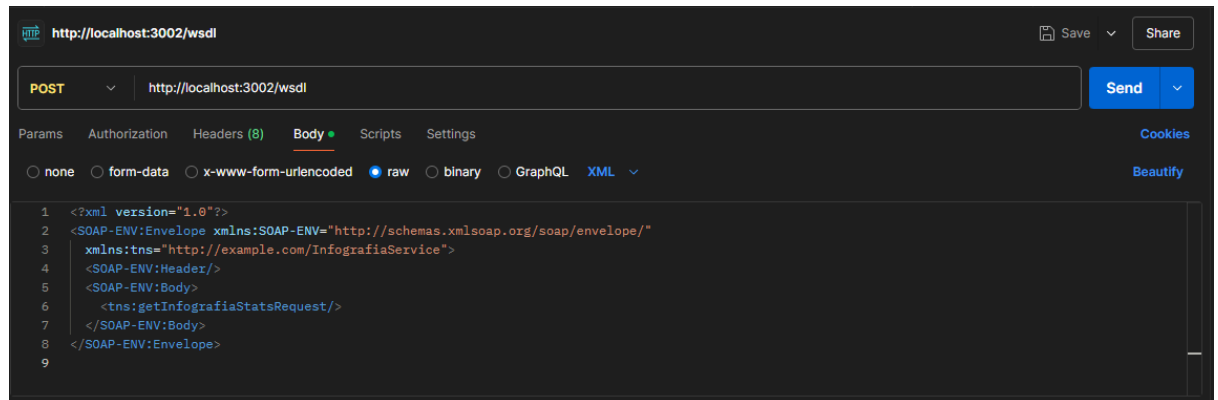
```
npm install soap
```

1. **Definición del Servicio SOAP:** Configuramos el servicio que manejará las solicitudes SOAP y devolverá las estadísticas de las infografías.

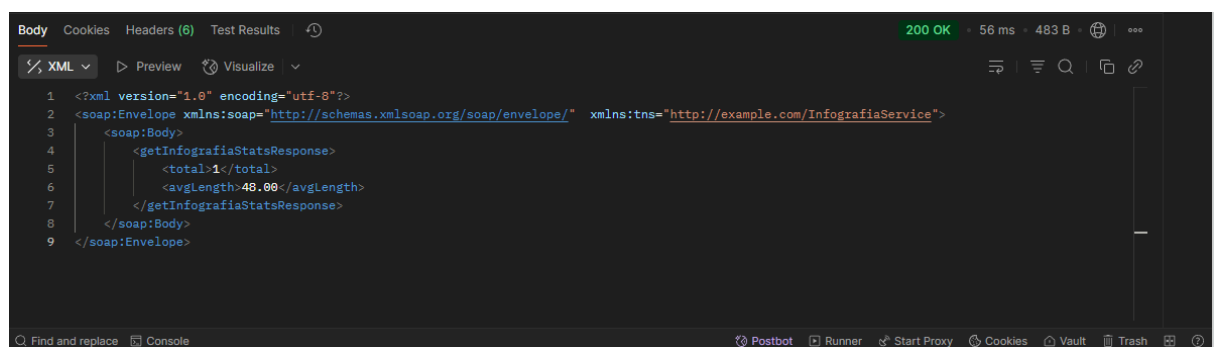
```
7 // Definir el servicio SOAP
8 const service = {
9   InfografiaService: {
10     InfografiaPort: {
11       getInfografiaStats: async function (args, callback) {
12         try {
13           // Consumir el endpoint de infografías
14           const response = await axios.get('http://localhost:3001/infografias');
15           const infografias = response.data;
16
17           if (infografias.length === 0) {
18             return callback(null, {
19               total: 0,
20               avgLength: null,
21             });
22           }
23
24           const total = infografias.length;
25           const avgLength = infografias.reduce((sum, infografia) => sum + infografia.texto.length, 0) / total;
26
27           return callback(null, {
28             total,
29             avgLength: avgLength.toFixed(2),
30           });
31         } catch (error) {
32           console.error('Error obteniendo datos:', error);
33           return callback(error);
34         }
35       },
36     },
37   },
38 };
39
```

Paso 3: Testing en Postman

1. **Configuración de Postman:** Mostramos cómo configurar una solicitud SOAP en Postman para consultar las estadísticas de las infografías.



2. **Pruebas de Respuesta:** Aseguramos que la respuesta devuelta por el servicio SOAP es correcta, mostrando las estadísticas de las infografías, como el total y la longitud promedio.



Conclusión

Este proyecto demostró cómo construir un microservicio SOAP que se conecta a un backend de Infografías y expone estadísticas sobre las infografías a través de un servicio SOAP. A través del uso de **Express**, **SOAP** y **MongoDB**, creamos una solución escalable y eficiente para gestionar las infografías y sus datos asociados. Además, el uso de **Postman** para probar el servicio nos permitió garantizar que todo funciona correctamente.

Retos Encontrados:

- Uno de los principales desafíos fue integrar correctamente la API de backend con el servicio SOAP, ya que ambos sistemas utilizan protocolos de comunicación diferentes (REST y SOAP). Sin embargo, esto se resolvió al usar **Axios** para realizar solicitudes HTTP desde el servicio SOAP.

Futuras Mejoras:

- **Ampliación de Endpoints:** Podemos agregar más funcionalidades al servicio SOAP para incluir más métricas relacionadas con las infografías.
- **Mejoras de Seguridad:** Implementar seguridad adicional para proteger las solicitudes SOAP.