

API Rest en .NET + Swagger + Microservicios

Integración de servicios



Microservicios

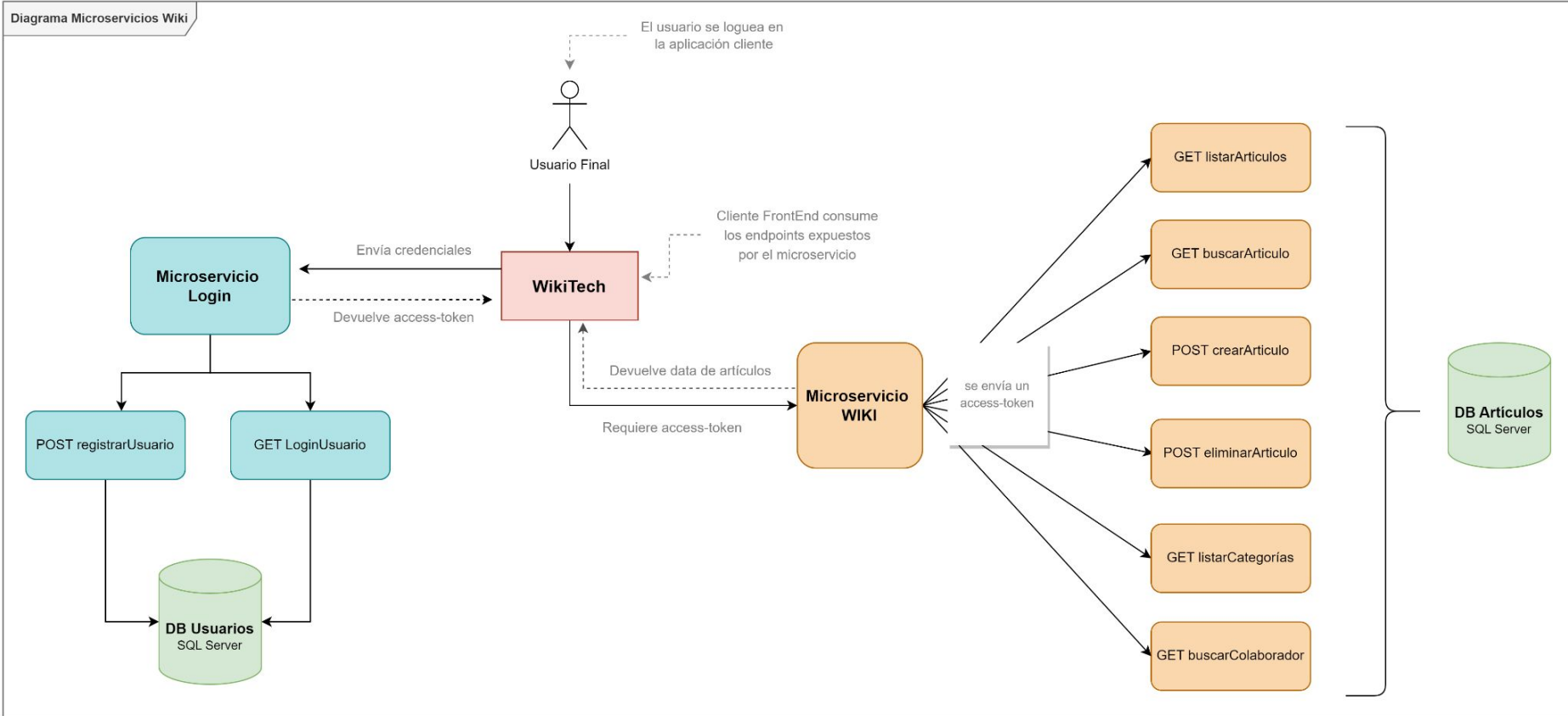
- Patrón de arquitectura
- Modularización del sistema
- Independencia y autonomía
- Database independiente
- Comunicación entre sí
- Gestión de equipos de desarrollo

API Rest

Application Programming Interface

- Contrato o interfaz definida
 - Desacoplamiento *cliente-servidor*
 - Protocolo HTTP → CRUD
 - URIs sencillas y predecibles
 - Representación de data → JSON
 - Headers HTTP y body
 - Códigos de estado
 - Intermediario entre dos sistemas
 - Facilita la gestión de los microservicios
-

Ejemplo de Integración - Diagrama de Arquitectura



Autenticación y Autorización

```
var jwt = _configuration.GetSection("Jwt").Get<Jwt>();
var claims = new[]
{
    new Claim(JwtRegisteredClaimNames.Sub, jwt.Subject),
    new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString()),
    new Claim(JwtRegisteredClaimNames.Iat, DateTime.UtcNow.ToString()),
    new Claim("id", userObtenido.IdUser.ToString()),
    new Claim("user", userObtenido.User.ToString()),
    new Claim("fullname", userObtenido.Fullname.ToString()),
};

var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(jwt.Key)); //c
var singIn = new SigningCredentials(key, SecurityAlgorithms.HmacSha256);
```

```
[Authorize]
[HttpDelete("{id}")]
[ProducesResponseType(StatusCodes.Status200OK, Type = typeof(bool))]
[ProducesResponseType(StatusCodes.Status404NotFound, Type = typeof(bool))]
0 referencias
public async Task<IActionResult> EliminarArticulo(int id)
{
    Articulo articulo = await _IArticuloServicio.BuscarArticulo(id);

    bool eliminado = false;
```

Documentación de APIs

 README.md

Aplicación Web WikiTech

WikiTech es una aplicación web basada en una wiki de artículos relacionados al ámbito tecnológico, donde los usuarios pueden ver los artículos, generar nuevos o eliminarlos.

La App consume el [Microservicio-Wiki](#) para obtener la data de los artículos y realizan las modificaciones necesarias. Además, para el registro y autenticación de usuarios, consume el [Microservicio-Login](#).


La Wiki tiene un formato colaborativo.

Tecnologías

- ASP.NET 6
- Entity Framework
- SQL Server

Diagrama de Arquitectura

```
graph TD
    User[Usuario Final] -- "El usuario se loguea en la aplicación desde" --> WikiTech[WikiTech]
    WikiTech -- "Envía credenciales" --> Login[Microservicio Login]
    Login -- "Devuelve access-token" --> WikiTech
    WikiTech -- "Cliente FrontEnd consume los endpoints expuestos por el microservicio" --> FE[Frontend]
```

 Swagger
Supported by SMARTBEAR

Microservicio.Wiki ^{1.0}

<https://localhost:7164/swagger/v1/swagger.json>

Artículo

GET	/api/articulo
POST	/api/articulo
GET	/api/articulo/{id}
DELETE	/api/articulo/{id}
PUT	/api/articulo/{id}

Métodos disponibles

1. POST crearArticulo

Tipo	POST
Endpoint	/crearArticulo
Objetivo	Generar una nueva entrada o artículo
Headers	Authorization: Bearer {{access-token}}

Parámetros de entrada

Nombre	Tipo	Requerido	Ejemplo
Json con metadata del objeto a crear	string	Si	<pre>{ "titulo": "Bootcamp Desarrollador Fullstack", "contenido": "Se lanzará nueva Bootcamp de programación", "fecha": "2019-01-06T17:16:40", "cantidad": 10 }</pre>

Swagger

OpenAPI

- Interfaz para documentar una API
 - Compatible con API REST
 - Información de códigos de estado
 - Pruebas de endpoints
 - Descripción de servicios
 - Swagger Editor/Swagger UI/Swagger Codegen
-

Documentación con Swagger

Api-login 1.0 OAS3

<https://localhost:7294/swagger/v1/swagger.json>

Login

GET

/login/mensaje

POST

/login/registerUser

POST

/login/loginSesion

Parameters

No parameters

Request body

```
{  "user": "sca@mail.com",  "password": "123"}
```

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7294/login/loginSesion' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "user": "sca@mail.com",
  "password": "123"
}'
```

Request URL

<https://localhost:7294/login/loginSesion>

Server response

Code	Details
200	<div><h4>Response body</h4><pre>{ "success": true, "message": "exito", "result": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJncnVwbzUiLCJqdGkiOiJQ61hahWUY29tIiwiaWZnVsbG5hbWUiOiJlY2YyZyBhbmV0IENvbnRlcyIsImV4cCI6MTY2Nzg5MDcyOSwiaXNjbG9ja3BhYiA"}</pre></div> <div><h4>Response headers</h4><pre>content-type: application/json; charset=utf-8 date: Tue, 08 Nov 2022 06:54:49 GMT server: Kestrel</pre></div>

Cliente Postman



Sca-Work New Import DEL eliminarAr GET listarCate Microserv POST crear GET listarArtic GET buscarAr

Collections + ...

- Microservicio-prueba
- Microservicio-Wiki
 - GET listarArticulos
 - POST crearArticulo
 - GET buscarArticulo
 - DEL eliminarArticulo
 - GET listarCategorias
 - GET buscarColaborador

Environments

Mock Servers

Monitors

Flows

History

Microservicio-Wiki / crearArticulo

POST https://localhost:7164/api/articulo

Params Authorization Headers (11) Body Pre-request Script Tests Settings

<input checked="" type="checkbox"/>	Accept-Encoding	gzip, deflate, br	
<input checked="" type="checkbox"/>	Connection	keep-alive	
<input checked="" type="checkbox"/>	Content-Type	application/json	
<input checked="" type="checkbox"/>	Authorization	Bearer {{access-token}}	
	Key	Value	Description

Body Cookies Headers (5) Test Results Status: 201 Created Time: 90 ms

Pretty Raw Preview Visualize JSON

```
1 {
2   "idArticulo": 37,
3   "titulo": "Bootcamp Desarrollador Fullstack",
4   "contenido": "Se lanzará nueva Bootcamp abierta para todo el que quiera incursionar en el mundo de la programación",
5   "fecha": "2019-01-06T17:16:40",
6   "puntaje": 10,
7   "idColaborador": 2,
```

Equipo y Tema

Grupo 5

API Rest en Net + Swagger +
Microservicios

Integrantes

Aragón Yamila

Cortes Valdes Scarlet

Galeano Carla

Machicado Cuba José

Repositorio GitHub

<https://github.com/Sacorv/WikiTech>

¡Muchas Gracias!

¿Preguntas?