

MICROSERVICIOS, SOA Y APIs – DOCUMENTO COMPLETO CON EJEMPLOS DE CÓDIGO (.md)

☀️ Microservicios, SOA y APIs – Resumen Completo con Ejemplos

Este documento contiene:

- Microservicios
- SOA
- ESB/Middleware
- APIs y Contratos
- Seguridad (tokens, API Key, JWT)
- Redes (intranet/extranet)
- Métodos HTTP
- Ejemplos reales en código

◇ 1. Conceptos Básicos

✓ Servicio

Resuelve un objetivo completo (ej: crear pedido).

✓ Microservicio

- Hace **una sola cosa**.
- Es **autónomo**, escalable y pequeño.
- Se comunica vía API.

🎯 Ejemplo real (Microservicio de Usuarios – Node.js)

```
```js
// users-service/index.js
const express = require("express");
const app = express();
app.use(express.json());

// Microservicio solo maneja usuarios
app.get("/usuarios", (req, res) => {
 res.json([{ id: 1, nombre: "Vicente" }]);
});

app.listen(3001, () => console.log("Users Microservice ON 3001"));
```

```

E 2. Arquitectura de Microservicios

- Muchos servicios pequeños.
- Cada uno tiene su **propia BD**, su propio servidor.
- Se accede mediante un **API Gateway**.

⌚ Ejemplo: API Gateway que enruta microservicios

```
// api-gateway/index.js
const express = require("express");
const proxy = require("express-http-proxy");
const app = express();

// Rutas hacia microservicios
app.use("/api/usuarios", proxy("http://localhost:3001"));
app.use("/api/pedidos", proxy("http://localhost:3002"));

app.listen(3000, () => console.log("API Gateway ON 3000"));
```

Esto es EXACTAMENTE lo que usan:

- Netflix
 - Amazon
 - Mercado Libre
-

🏛️ 3. SOA (Arquitectura Orientada a Servicios)

SOA permite que varios sistemas se comuniquen entre sí mediante **servicios más grandes**, usando un **Middleware/ESB** para transformar y enrutar datos.

✓ SOA usa:

- ESB (Enterprise Service Bus)
- Orquestación de servicios
- Contratos de servicio

⌚ Ejemplo de Orquestación (pseudo-código)

```
// Orquestador SOA para procesar un pedido
async function procesarPedido(pedido) {
    await servicioPagos.validarPago(pedido);
    await servicioStock.reservarStock(pedido);
    await servicioDespacho.generarDespacho(pedido);

    return { mensaje: "Pedido procesado correctamente" };
}
```

Aquí SOA **coordina** el flujo de varios servicios.

4. Middleware / ESB (Enterprise Service Bus)

Funciona como intermediario.

Funciones:

- Transformar datos (XML ↔ JSON)
- Enrutar solicitudes
- Seguridad
- Integración entre sistemas distintos

Ejemplo ESB transformando XML a JSON

```
// Ejemplo simple de transformación
const xml2js = require("xml2js");

function transformarXMLaJSON(xml) {
    return xml2js.parseStringPromise(xml);
}

const xml = `<cliente><nombre>Vicente</nombre></cliente>`;

transformarXMLaJSON(xml).then(console.log);
```

Salida:

```
{ "cliente": { "nombre": "Vicente" } }
```

5. APIs y Contratos

API

Interfaz que permite consumir recursos de un servicio.

Contrato de API

El contrato define:

- parámetros
- seguridad
- formato de datos
- limitaciones
- SLA

- rutas disponibles
-

🌐 6. Ejemplo de Contrato de API REAL (JSON)

```
{  
    "name": "Servicio de Pagos",  
    "version": "1.0.0",  
    "endpoints": [  
        {  
            "method": "POST",  
            "path": "/pago/validar",  
            "params": {  
                "monto": "number",  
                "tarjeta": "string"  
            },  
            "responses": {  
                "200": "Pago aprobado",  
                "400": "Tarjeta inválida",  
                "401": "Token inválido"  
            },  
            "auth": "API Key"  
        }  
    ]  
}
```

🔑 7. Seguridad en APIs

Métodos que mencionó el profe:

- API Key
- Tokens
- JWT
- OpenID
- MFA (ej: Steam Guard)

🌐 Ejemplo: API protegida con API Key (Node.js)

```
app.get("/recurso-seguro", (req, res) => {  
    const key = req.headers["x-api-key"];  
    if (key !== "12345XYZ") {  
        return res.status(401).json({ error: "API Key inválida" });  
    }  
    res.json({ mensaje: "Acceso autorizado" });  
});
```

Ejemplo JWT (JSON Web Token)

```
// Generar token
const jwt = require("jsonwebtoken");
const token = jwt.sign({ userId: 1 }, "clave_secreta", { expiresIn: "1h" });

console.log(token);
```

8. Redes: Intranet y Extranet

Intranet

Red privada interna.

Ejemplo:

```
http://192.168.1.10:8080
```

Extranet

Permite acceso controlado desde internet.

Ejemplo:

```
https://empresa.com/proveedores
```

9. Métodos HTTP con Ejemplos

GET

Visible en URL.

```
GET /api/productos?categoria=celulares
```

Ejemplo PHP:

```
$nombre = $_GET["nombre"];
echo "Buscando: ".$nombre;
```

✓ POST

Datos ocultos.

```
POST /api/pedidos/crear
```

Ejemplo Node.js:

```
app.post("/api/login", (req, res) => {
  const { email, pass } = req.body;
  res.json({ mensaje: "Login OK" });
});
```

🔧 10. Ejemplos Completos: Caso Mercado Libre

Mercado Libre usa:

- microservicio de productos
- microservicio de pagos
- microservicio de envíos
- microservicio de notificaciones

Ejemplo de Microservicio de Envíos:

```
app.post("/envios/generar", (req, res) => {
  const { pedidoId, direccion } = req.body;
  res.json({
    mensaje: "Envío generado",
    tracking: "CHL-1234567"
  });
});
```

💻 11. Arquitectura Cloud

- Servicios se ejecutan en servidores o nube.
 - Importa:
 - RAM
 - CPU
 - Latencia
-

⌚ RESUMEN FINAL (para prueba)

Microservicio:

- Una sola función
- Autónomo
- Escalable
- Se accede por API

SOA:

- Servicios más grandes
- Orquestación
- Middleware/ESB

API:

- Contrato
- Endpoints
- Token / API Key

Seguridad:

- JWT
- API Key
- OpenID

Redes:

- Intranet
- Extranet

Datos:

- JSON / XML / YAML
-