

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA
CENTRO UNIVERSITARIO QUETZALTENANGO
FACULTAD DE INGENIERÍA EN SISTEMAS

CURSO: Programación 2.
ING. Samuel Gomez



Nombre: Antonio Daniel Xocol Cuc Carné: 1490-22-14563

Tema: Servicios Web

Servicio Web:

Los servicios web son sistemas software diseñados para soportar interacciones máquina a máquina sobre una red. Permiten la comunicación entre aplicaciones distintas para intercambiar datos y funcionalidades a través de internet.

Los servicios web permiten la comunicación entre sistemas y aplicaciones, facilitando la integración y el desarrollo de arquitecturas escalables. Son ampliamente utilizados para construir APIs y integrar funcionalidades remotas.

Tipos de Servicios Web:

1. SOAP (Simple Object Access Protocol): Es un protocolo estándar que utiliza XML para intercambiar datos y facilitar la integración entre aplicaciones. Utiliza WSDL para describir los servicios.
2. REST (Representational State Transfer): Utiliza HTTP y sus métodos (GET, POST, PUT, DELETE) para exponer recursos y operaciones. No impone restricciones en los datos, ideal para servicios ligeros.
3. RPC (Remote Procedure Call): Permite invocar procedimientos remotos como si fueran locales. El cliente realiza una llamada al procedimiento y espera la respuesta. Utiliza un IDL para describir la interfaz.
4. Websockets: Permite una comunicación bidireccional y en tiempo real entre clientes y servidores. Mantiene una sesión abierta para intercambiar mensajes. Útil en aplicaciones que requieren notificaciones en tiempo real.
5. GraphQL: Permite a los clientes pedir específicamente los datos que necesitan. Los clientes pueden solicitar diferentes partes de los datos disponibles.

Ejemplos

1. // Servicio SOAP

```
import javax.jws.WebService;
```

```
import javax.jws.WebMethod;
```

```
@WebService
```

```
public class HelloWorld{
```

```
@WebMethod
```

```
public String helloWorld(){
```

```
    return "Hello World";
```

```
}  
}
```

2. // Cliente SOAP

```
import javax.xml.namespace.QName;  
import javax.xml.ws.Service;
```

```
public class Client {
```

```
    public static void main(String[ ] args) {
```

```
        Service service = Service.create(new QName("http://helloservice.org/wsdl", "HelloWorld"));
```

```
        HelloWorld hello = service.getPort(HelloWorld.class);
```

```
        System.out.println(hello.helloWorld());
```

```
    }
```

```
}
```

```
// Recurso REST
```

```
import javax.ws.rs.GET;
```

```
import javax.ws.rs.Path;
```

```
import javax.ws.rs.Produces;
```

```
import javax.ws.rs.core.MediaType;
```

```
@Path("/hello")
```

```
public class HelloResource {
```

```
    @GET
```

```
    @Produces(MediaType.TEXT_PLAIN)
```

```
    public String sayHello() {
```

```
        return "Hello World!";
```

```
}  
}
```

// Cliente REST

```
Client client = ClientBuilder.newClient();
```

```
String response = client.target("http://localhost:8080/hello").request().get(String.class);
```

```
System.out.println(response);
```

Los diagramas de usos pueden ser:

3. APIs públicas como servicios meteorológicos, mapas, etc. Permite acceder a funcionalidades de forma estandarizada.
4. Integración entre sistemas corporativos, por ejemplo un servicio de autenticación centralizado.
5. Aplicaciones móviles que consumen APIs backend para funciones como login con redes sociales.
6. Plataformas como Salesforce que exponen funcionalidad vía APIs REST.
7. Microservicios que se comunican entre sí para construir aplicaciones complejas.