

Desarrollo de Software Multiplataforma

App Web  
Orientadas a Servicios  
Saber

Gonzalez Lozano Fatima Estefania  
Martinez Montelongo Lizeth Guadalupe

MAYL. David Belmares

Unidad # 1

Introducción al Desarrollo Web  
Orientado a Servicios

Fecha de Entrega:

jueves, 22 de enero del 2026

## **Contenido de la Primera Unidad**

La opción “**Introducción al Desarrollo Web Orientado a Servicios**” deberá mostrar información clara y organizada sobre los siguientes temas:

### *1. Paradigma del desarrollo de aplicaciones orientadas a servicios*

- Distinguir los **servicios que se ofrecen en la nube**.
- Identificar las **características de las aplicaciones orientadas a servicios**.
- Identificar el **concepto y las características de las aplicaciones web híbridas (Mashup)**.

### *2. Arquitectura Orientada a Servicios (SOA)*

- Definir qué es la **Arquitectura Orientada a Servicios (SOA)**.
- Identificar los **principios de diseño** aplicados a cada servicio modelado.
- Identificar los **estándares relacionados con los servicios**, tales como:
  - ✓ XML
  - ✓ SOAP
  - ✓ WSDL
  - ✓ UDDI
  - ✓ REST

## **Evidencias**

En esta sección deberá de incluir capturas de pantallas de su portafolio , anexar entre 6 y 10 imágenes que muestren la evidencia de su App Web.

## *1. Paradigma del desarrollo de aplicaciones orientadas a servicios*

### **Servicios que se ofrecen en la nube**

Los principales modelos de servicios en la nube son:

- **IaaS (Infraestructura como Servicio):** alquiler de servidores, almacenamiento y redes. Ejemplo: Amazon EC2.
- **PaaS (Plataforma como Servicio):** entorno de desarrollo con herramientas y frameworks listos para usar. Ejemplo: Google App Engine.
- **SaaS (Software como Servicio):** aplicaciones completas accesibles vía navegador. Ejemplo: Microsoft 365.
- **FaaS (Function as a Service):** ejecución de funciones bajo demanda, sin gestionar servidores. Ejemplo: AWS Lambda.
- **XaaS (Anything as a Service):** concepto que engloba cualquier servicio entregado vía nube.

### **Características de las aplicaciones orientadas a servicios (SOA)**

La **arquitectura orientada a servicios (SOA)** es un paradigma que organiza el software en servicios independientes y reutilizables. Sus características clave:

- **Interoperabilidad:** los servicios se comunican mediante estándares abiertos (SOAP, REST, XML, JSON).
- **Reutilización:** cada servicio puede integrarse en múltiples aplicaciones.
- **Modularidad:** los sistemas se construyen como bloques independientes.
- **Escalabilidad y flexibilidad:** fácil integración de nuevos servicios sin rediseñar todo el sistema.
- **Autonomía:** cada servicio ejecuta una función empresarial completa (ej. procesar pagos, validar usuarios).
- **Integración simplificada:** evita duplicar código y facilita la conexión entre aplicaciones heterogéneas.

## **Concepto y características de aplicaciones web híbridas (Mashup)**

Un **mashup** es una aplicación web híbrida que combina datos, servicios o funcionalidades de diferentes fuentes en una sola interfaz.

- **Concepto:** integración de APIs abiertas o servicios web para crear un nuevo servicio.
- **Características principales:**
  - **Combinación:** mezcla de datos y funciones de múltiples proveedores.
  - **Agregación:** reúne información en un solo lugar.
  - **Visualización:** transforma datos en interfaces más útiles y atractivas.
  - **Rapidez de desarrollo:** se construyen en poco tiempo gracias a la reutilización de servicios existentes.
  - **Ejemplos comunes:** integración de Google Maps con datos de hoteles, o un mashup que combine YouTube y Twitter para mostrar tendencias

## *2. Arquitectura Orientada a Servicios (SOA)*

### **¿Qué es la Arquitectura Orientada a Servicios (SOA)?**

La Arquitectura Orientada a Servicios (SOA, por sus siglas en inglés) es un enfoque de diseño de software que organiza las aplicaciones como un conjunto de servicios independientes, reutilizables y bien definidos.

Cada servicio realiza una función específica del negocio y se comunica con otros servicios a través de protocolos estándar, generalmente sobre una red.

El objetivo principal de SOA es facilitar la integración, flexibilidad, mantenimiento y reutilización de los sistemas, permitiendo que aplicaciones desarrolladas en diferentes lenguajes o plataformas puedan interactuar entre sí.

### **Principios de diseño aplicados a cada servicio en SOA**

Los servicios dentro de una arquitectura SOA se diseñan siguiendo principios fundamentales que garantizan su correcto funcionamiento y reutilización:

#### **1. Acoplamiento débil**

Los servicios están poco dependientes entre sí, lo que permite modificar o reemplazar un servicio sin afectar a los demás.

#### **2. Reusabilidad**

Un servicio puede ser utilizado por múltiples aplicaciones o procesos de negocio sin necesidad de duplicar código.

#### **3. Autonomía**

Cada servicio controla su propia lógica y datos, funcionando de manera independiente.

#### **4. Abstracción**

El usuario del servicio solo conoce la interfaz, no la implementación interna del servicio.

## **5. Interoperabilidad**

Los servicios pueden comunicarse entre sí sin importar el lenguaje de programación o sistema operativo utilizado.

## **6. Descubrimiento**

Los servicios pueden ser publicados y localizados fácilmente para ser utilizados por otros sistemas.

## **7. Stateless (sin estado)**

Los servicios no mantienen información de estado entre solicitudes, mejorando el rendimiento y la escalabilidad.

## **Estándares relacionados con los servicios en SOA**

SOA se apoya en diversos estándares y tecnologías que permiten la comunicación y descripción de los servicios:

**XML (Extensible Markup Language)**

Es un lenguaje de marcado que permite estructurar y transportar datos de forma legible y estándar.

Se utiliza ampliamente en SOA para intercambiar información entre servicios.

## **SOAP (Simple Object Access Protocol)**

Es un protocolo basado en XML que permite la comunicación entre servicios web.

Utiliza generalmente HTTP o HTTPS para enviar mensajes estructurados y seguros.

Características principales:

- Comunicación formal y estandarizada
- Soporte para seguridad y transacciones
- Uso común en sistemas empresariales

## **WSDL (Web Services Description Language)**

Es un lenguaje basado en XML que describe cómo funciona un servicio web, indicando:

- Qué operaciones ofrece
- Qué mensajes intercambia
- Cómo se debe acceder al servicio

Sirve como un contrato entre el proveedor y el consumidor del servicio.

## **UDDI (Universal Description, Discovery and Integration)**

Es un estándar que permite publicar, buscar y descubrir servicios web. Funciona como un directorio donde las organizaciones registran sus servicios para que otros puedan encontrarlos.

## **REST (Representational State Transfer)**

Es un estilo arquitectónico que utiliza los métodos HTTP estándar como:

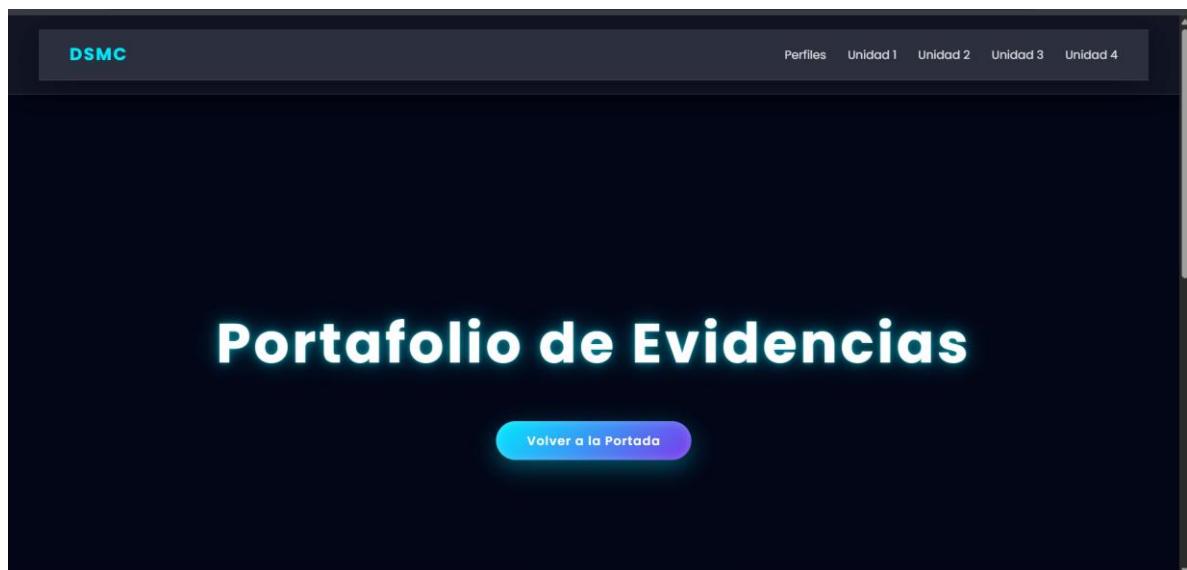
- GET
- POST
- PUT
- DELETE

Los servicios REST son más ligeros que SOAP y suelen usar formatos como JSON o XML.

### **Ventajas de REST:**

- Simplicidad
- Alto rendimiento
- Amplio uso en aplicaciones web y móviles

## Evidencias



This screenshot displays three cards on a dark background, each containing text about the portfolio's purpose, user learning outcomes, and project objectives.

- Propósito del Portafolio**  
Presentar de manera organizada y visual las evidencias académicas desarrolladas durante la materia de Desarrollo Web Orientado a Servicios.
- ¿Qué aprenderá el usuario?**  
Conceptos de arquitectura SOA, implementación y desarrollo de APIs, integración de servicios web y buenas prácticas en aplicaciones web.
- Objetivo del Proyecto**  
Diseñar y desarrollar una aplicación web funcional que concentre los conocimientos adquiridos en el curso.

# Panel de Usuarios

Información general de los integrantes



Lizeth Guadalupe Martínez Montelongo

Carrera: Desarrollo de Software

Edad: 19 años

Ocupación: Estudiante

Grupo: C

Correo: martinezlizethgpe@email.com



Fatima Estefania Gonzalez Lozano

Carrera: Desarrollo de Software

Edad: 19 años

Ocupación: Estudiante

Grupo: C

Correo: fg6734364@email.com

← Volver a Actividades

UNIDAD 1

Perfiles Unidad 1 Unidad 2 Unidad 3 Unidad 4

## Unidad 1

Desarrollo Web Orientado a Servicios

Saber

Saber Hacer 1

Saber Hacer 2

UNIDAD

Perfiles Unidad 1 Unidad 2 Unidad 3 Unidad 4

## Página en mantenimiento

Esta sección se encuentra actualmente en desarrollo.  
Próximamente estará disponible.

