

TALLER: WEB SERVICES EN EL DESARROLLO DE SOFTWARE

DATOS DEL GRUPO

Integrantes: Marsoley Carrero Bayona

Fecha: _____25/09/2025_____ **Duración estimada:** 1:30 Hrs

OBJETIVOS DE APRENDIZAJE

Al finalizar este taller, los estudiantes serán capaces de:

1. **Identificar** las ventajas y utilidades de los Web Services en el desarrollo de aplicaciones modernas.
2. **Reconocer** la estructura y elementos básicos que componen un Web Service.
3. **Analizar** casos prácticos donde los Web Services son la solución más apropiada.
4. **Diseñar** propuestas básicas de implementación de Web Services para problemas específicos.

ACTIVIDADES PRÁCTICAS

ACTIVIDAD 1: Identificando Web Services en la vida cotidiana

Instrucciones: En grupo, identifiquen 5 situaciones de la vida cotidiana donde interactúan con Web Services (pueden ser apps del celular, páginas web, etc.).

Ejemplo: Consultar el clima en una app móvil - la app consulta un Web Service meteorológico

Situación: Hacer un pedido en una app de comida a domicilio (ej: Rappi)

Web Service involucrado: Servicio de geolocalización para entrega y servicio de procesamiento de pagos.

Situación: Consultar el saldo de una tarjeta de crédito en la app del banco

Web Service involucrado: Servicio de consulta de saldos y transacciones del banco.

Situación: Recibir notificaciones de ofertas por email después de visitar un sitio web

Web Service involucrado: Servicio de marketing por email (ej: Mailchimp, SendGrid).

Situación: Rastrear un paquete en tiempo real desde la página de una transportadora

Web Service involucrado: Servicio de seguimiento de envíos (ej: API de DHL, FedEx).

Situación: Sincronizar el inventario de un restaurante con las plataformas de delivery

Web Service involucrado: Servicio de gestión de inventario en tiempo real (ej: API de integración con Rappi, PedidosYa).

Reflexión grupal: ¿Qué pasaría si estos Web Services no existieran?

Si estos Web Services no existieran, las aplicaciones tendrían que desarrollar todas estas funcionalidades de manera interna, lo que aumentaría significativamente los costos, tiempos de desarrollo y complejidad técnica. Además, se perdería la especialización, la escalabilidad, la confiabilidad y la capacidad de integración rápida con otros sistemas, resultando en una experiencia de usuario más limitada y propensa a errores.

ACTIVIDAD 2: Análisis de caso - Sistema de E-commerce

Contexto: Una tienda en línea necesita integrar los siguientes servicios externos:

1. Procesamiento de pagos
2. Cálculo de envíos
3. Validación de inventario en bodega
4. Notificaciones por email
5. Geolocalización para entrega

Instrucciones: Para cada servicio, completen la siguiente información:

Servicio de Procesamiento de pagos

1. ¿Qué datos debe recibir este Web Service?:
Número de tarjeta, fecha de expiración, CVV, monto, moneda, referencia del pedido.
2. ¿Qué información debe devolver?: Confirmación de pago, código de autorización, estado de la transacción (aprobado/rechazado), ID de la transacción.
3. ¿Qué ventajas tiene usar un Web Service externo vs. desarrollar internamente?:

Mayor seguridad (cumplimiento PCI-DSS), menor responsabilidad legal, actualizaciones automáticas de seguridad y funcionalidad, reducción de costos de desarrollo y mantenimiento, y acceso a métodos de pago globales.

Servicio de Cálculo de Envíos

¿Qué datos debe recibir este Web Service?: Dirección de origen (restaurante), dirección de destino del cliente, peso y dimensiones del pedido, tipo de servicio (estándar/express).

¿Qué información debe devolver?: Costo de envío, tiempo estimado de entrega, opciones de envío disponibles.

¿Qué problemas podría ocasionar si este servicio falla?: No se podrían calcular costos de entrega, generando retrasos en la confirmación de pedidos, confusiones con los clientes sobre el precio final, y posible cancelación de pedidos, lo que lleva a insatisfacción del cliente y pérdida de ventas.

Paso 3: Identifiquen los componentes en base a su WEB SERVICE ya socializado.

¿Quiénes serían los consumidores de este servicio?

- Apps de delivery (Rappi, Uber Eats, PedidosYa).
- Punto de Venta (POS) interno del restaurante.
- Aplicación móvil de gestión para el personal del restaurante.

¿Qué otros sistemas necesitarían integrarse?

- Sistema de cocina (para alertar sobre pedidos listos).
- Sistema de base de datos de proveedores (para reposición automática).
- Sistema de reportes y analytics.

Paso 4: Ventajas de su propuesta

¿Qué problemas resolvería su Web Service?:

- Sincronización en tiempo real del inventario entre el restaurante y múltiples plataformas de delivery.
- Reducción de errores y conflictos por venta de productos agotados (sobreventa).
- Automatización de alertas para reposición de insumos.
- Optimización del proceso de compra y gestión de stock.

¿Qué ventajas tendría para las sus clientes?

- Menos cancelaciones de pedidos debido a falta de producto.
- Experiencia de compra más confiable y transparente.
- Mayor satisfacción al garantizar la disponibilidad de los platos ordenados.

PREGUNTAS DE REFLEXIÓN

1. ¿Cuál considera que es la principal ventaja de usar Web Services en el desarrollo de software?

La principal ventaja es la capacidad de facilitar la integración de sistemas heterogéneos, promoviendo la reutilización de funcionalidades especializadas, permitiendo la escalabilidad

independiente de servicios y acelerando el tiempo de desarrollo al no tener que construir todo desde cero.

2. ¿Qué desafíos o dificultades podrían surgir al implementar Web Services?

Podrían surgir desafíos relacionados con la seguridad (autenticación, autorización, cifrado de datos), la dependencia de servicios de terceros (disponibilidad, latencia), la gestión de versiones de las APIs, el manejo robusto de errores y fallos, y la complejidad inherente en el diagnóstico de problemas en un sistema distribuido.

3. ¿En qué tipo de proyectos de software sería indispensable el uso de Web Services?

Sería indispensable en proyectos que requieran interoperabilidad entre sistemas distintos, como aplicaciones de e-commerce, plataformas de logística y delivery, sistemas de pagos en línea, aplicaciones móviles que consumen datos externos, sistemas de reservas (vuelos, hoteles, restaurantes) y cualquier arquitectura basada en microservicios.

Compromisos del Grupo

¿Qué se comprometen a investigar más después de este taller?

Nos comprometemos a investigar más a fondo sobre la implementación de seguridad en APIs, las mejores prácticas para el diseño de Web Services RESTful (versionado, documentación con OpenAPI) y estrategias para manejar la tolerancia a fallos (patrones como Circuit Breaker).

¿Qué aplicación práctica le darán a este conocimiento?

Aplicaremos este conocimiento en el diseño e implementación práctica de un prototipo de Web Service para la gestión de inventario de un restaurante, integrando al menos una plataforma de delivery simulada (como Rappi o Uber Eats) para demostrar la sincronización de stock en tiempo real.