

**Evaluación Continua MF0492\_3 – UF1846**

**Erick Montoya**

**Pregunta 1: ¿Cuál de las siguientes es una característica general de las arquitecturas de servicios distribuidos? (1 punto)**

A.

**Pregunta 2: ¿Cuál es una característica de los certificados digitales? (1 punto)**

C.

**Pregunta 3: ¿Qué estándar de seguridad se utiliza en servicios web para garantizar la autenticación y la integridad de los mensajes? (1 punto)**

A.

**Pregunta 4: ¿Qué protocolo es comúnmente utilizado para la implementación de servicios web basados en mensajes? (1 punto)**

B.

**Pregunta 5: ¿Cuál de las siguientes afirmaciones describe mejor el modelo RBAC (control de acceso basado en roles)? (1 punto)**

B.

**Pregunta 6: ¿Qué herramienta se puede utilizar para la definición y prueba de servicios web? (1 punto)**

D.

**Pregunta 7: ¿Qué tipo de criptografía utiliza una clave pública y una clave privada para cifrar y descifrar datos? (1 punto)**

B.

**Pregunta 8: ¿Cuál es una ventaja clave de utilizar UDDI en servicios web? (1 punto)**

A.

**Pregunta 9: ¿Cuál es la diferencia principal entre servicios web basados en mensajes y servicios web basados en recursos? (1 punto)**

C.

**Pregunta 10: ¿Qué característica distingue a los servicios web SOAP de los servicios web REST? (1 punto)**

C.

**Pregunta 11: ¿Cuál es la función principal de un proveedor de servicios en una arquitectura orientada a servicios (SOA)? (1 punto)**

b)

**Pregunta 12: ¿A qué hace referencia el “ProductoPromocionPedidoId” en el siguiente fragmento de código? (1 punto)**

```
@Repository
public interface ProductoPromocionPedidoRepository extends JpaRepository<ProductoPromocionPedido,
ProductoPromocionPedidoId> {
    @Query("SELECT SUM(p.cantidad) FROM ProductoPromocionPedido p WHERE p.idPedido = :idPedido")
    Integer calcularCantidadPedido(@Param("idPedido") int idPedido);
}
```

C)

### **Problema práctico (12 puntos -un punto por cada apartado-)**

#### **\*Este ejercicio no requiere de código de programación**

Eres el desarrollador principal de una aplicación web para una librería en línea. La aplicación necesita gestionar un catálogo de libros, las órdenes de los clientes y los detalles de cada usuario. Debes diseñar y seleccionar los servicios web adecuados para integrar estos componentes de manera eficiente y segura. Considera las siguientes tareas específicas:

#### **1. Gestionar el Catálogo de Libros:**

1.1. Obtener la lista completa de libros disponibles. (1 punto)

Método HTTP: GET

Justificación: GET es el método estándar para tomar datos del servidor sin hacer modificaciones

1.2. Añadir un nuevo libro al catálogo. (1 punto)

Método HTTP: POST

Justificación: POST se utiliza para enviar datos al servidor para crear nuevo recurso

1.3. Actualizar la información de un libro existente. (1 punto)

Método HTTP: PUT

Justificación: PUT se utiliza para actualizar recurso existente y reemplaza con los nuevos datos

1.4. Eliminar un libro del catálogo. (1 punto)

Método HTTP: DELETE

Justificación: DELETE se utiliza para eliminar un recurso del servidor

#### **2. Gestionar las Órdenes de Clientes:**

2.1. Crear una nueva orden de compra. (1 punto)

Método HTTP: POST

Justificación: POST sirve para enviar datos al servidor para crear una nueva orden de compra

2.2. Obtener los detalles de una orden específica. (1 punto)

Método HTTP: GET

Justificación: GET se utiliza para recuperar datos de una orden específica, sin modificar datos

2.3. Actualizar el estado de una orden. (1 punto)

Método HTTP: PATCH

Justificación: PATCH se utiliza para modificar parcialmente un recurso, como el estado de una orden

2.4. Eliminar una orden si es necesario. (1 punto)

Método HTTP: DELETE

Justificación: DELETE se utiliza para eliminar un recurso en específico, como una orden

### **3. Gestionar los Detalles del Usuario:**

3.1. Crear un nuevo perfil de usuario. (1 punto)

Método HTTP: POST

Justificación: POST se utiliza para enviar datos al servidor y crear un nuevo perfil

3.2. Obtener la información de un usuario específico. (1 punto)

Método HTTP: GET

Justificación: GET se utiliza para recuperar datos sobre un usuario específico, sin hacer modificaciones

3.3. Actualizar la información del usuario. (1 punto)

Método HTTP: PUT

Justificación: PUT se utiliza para actualizar recurso existente y reemplaza completamente con la nueva información

3.4. Eliminar un perfil de usuario. (1 punto)

Método HTTP: DELETE

Justificación: DELETE se utiliza para borrar un recurso específico del servidor, como un usuario