

## Ejercicio Evaluación continua MF0492\_3 - UF1846

**Adrián Sobrevela**

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

- A. Permiten la interoperabilidad entre diferentes sistemas**
- B. Aumentan la dependencia de sistemas monolíticos
- C. Restringen el uso de protocolos estándares
- D. Limitan la escalabilidad de las aplicaciones

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

- A. Son válidos indefinidamente sin necesidad de renovación
- B. Se utilizan únicamente para firmar correos electrónicos
- C. Permiten la autenticación y el cifrado en comunicaciones electrónicas**
- D. No requieren una entidad certificadora para ser válidos

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

- A. WS-Security**
- B. SSL
- C. TLS
- D. HTTP

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

- A. SMTP
- B. SOAP**
- C. FTP
- D. REST

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

- A. RBAC permite a los usuarios definir sus propios roles y permisos.
- B. RBAC asigna permisos a roles, y los usuarios se asignan a esos roles.**
- C. RBAC es un sistema de control de acceso basado en listas.
- D. RBAC solo se utiliza en sistemas operativos, no en aplicaciones web.

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

- A. MySQL
- B. Adobe
- C. NetBeans
- D. Postman**

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

- A. Criptografía simétrica
- B. Criptografía asimétrica**
- C. Criptografía de flujo
- D. Criptografía de bloque

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

- A. Facilita el descubrimiento y la publicación de servicios web.**
- B. Mejora la compatibilidad entre diferentes lenguajes de programación.
- C. Garantiza la encriptación de los datos transmitidos.
- D. Permite la ejecución de código en múltiples plataformas.

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

- A. Los servicios basados en mensajes permiten una comunicación segura, mientras que los basados en recursos no.
- B. Los servicios basados en mensajes son más adecuados para aplicaciones móviles, mientras que los basados en recursos son más adecuados para aplicaciones web.
- C. Los servicios basados en mensajes siempre utilizan el protocolo SOAP, mientras que los basados en recursos siempre utilizan REST.
- D. Los servicios basados en mensajes se enfocan en la comunicación de datos, mientras que los basados en recursos se enfocan en el estado de los recursos.**

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

- A. Los servicios SOAP permiten la comunicación asíncrona, mientras que los servicios REST no.
- B. Los servicios SOAP utilizan operaciones HTTP, mientras que los servicios REST utilizan protocolos personalizados.
- C. Los servicios SOAP requieren un contrato definido en WSDL, mientras que los servicios REST se basan en URIs y utilizan operaciones estándar HTTP.**
- D. Los servicios SOAP son siempre más eficientes en términos de rendimiento que los servicios REST.

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

- a) Crear y consumir servicios web
- b) Proporcionar y publicar servicios web**
- c) Monitorizar el tráfico de la red
- d) Gestionar bases de datos

**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);
}
```

- A) Es una entidad que representa la tabla ProductoPromocionPedido en la base de datos.
- B) Es un servicio que gestiona las operaciones de negocio relacionadas con ProductoPromocionPedido.
- C) Es una clase que define la clave primaria compuesta de la entidad ProductoPromocionPedido.**
- D) Es una clase que contiene la lógica de implementación para la entidad ProductoPromocionPedido.

### **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 El método GET está diseñado para recuperar información de manera segura y eficiente, y es idóneo para este tipo de operación según las especificaciones HTTP.*

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

*Método HTTP: POST El método POST está diseñado para enviar datos al servidor y realizar operaciones de creación, siguiendo los estándares HTTP.*

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

*Método HTTP: PUT El método PUT es ideal para actualizaciones completas de recursos, según las especificaciones HTTP.*

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

*Método HTTP: DELETE El método DELETE está diseñado específicamente para eliminar recursos, siguiendo los estándares HTTP.*

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

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

*Método HTTP: POST El método POST es ideal para enviar datos y realizar operaciones de creación, según los estándares HTTP.*

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

*Método HTTP: GET El método GET es idóneo para operaciones de lectura según las especificaciones HTTP.*

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

*Método HTTP: PUT El método PUT es ideal para actualizaciones completas de recursos, según las especificaciones HTTP.*

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

*Método HTTP: DELETE El método DELETE está diseñado específicamente para eliminar recursos, siguiendo los estándares HTTP.*

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

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

*Método HTTP: POST El método POST es ideal para enviar datos y realizar operaciones de creación, según los estándares HTTP.*

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

*Método HTTP: GET El método GET es idóneo para operaciones de lectura según las especificaciones HTTP.*

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

*Método HTTP: PUT El método PUT es ideal para actualizaciones completas de recursos, según las especificaciones HTTP.*

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

*Método HTTP: DELETE El método DELETE está diseñado específicamente para eliminar recursos, siguiendo los estándares HTTP.*

### **Tarea**

Para cada una de las tareas mencionadas arriba, especifica qué tipo de método HTTP (GET, POST, PUT, DELETE) utilizarías y justifica tu elección. Considera los siguientes aspectos en tu reflexión:

- La naturaleza de la operación (lectura, creación, actualización, eliminación).
- La necesidad de seguridad en la transmisión de datos.
- La estructura y claridad del diseño del servicio web.
- Las especificaciones y estándares de los protocolos web.