

1. En el contexto de la asignatura, indique la afirmación correcta:

- a) La arquitectura de integración propuesta está compuesta por dos capas: una capa de integración a nivel de plataforma y otra capa de integración a nivel de procesos de negocio.
- b) La arquitectura de integración propuesta está formada por dos capas: una capa de integración de datos y otra capa de integración de procesos de negocio.
- c) La arquitectura de integración propuesta está formada por tres capas: una capa de integración de datos, otra capa de integración a nivel de procesos de negocio y una capa de gestión de API.
- d) La arquitectura de integración propuesta está formada por cinco capas: a nivel inferior se encuentra la capa de integración a nivel de plataforma, otro nivel lo constituye la capa de integración de datos, sobre ella se encuentra la capa de integración a nivel de procesos de negocio, y después se encuentra la capa de gestión de API y la capa de presentación.

2. En base a lo estudiado a lo largo de la asignatura, un proceso de negocio ...

- a) Se divide en una serie de tareas o actividades individuales que se realizan en un orden no definido.
- b) Puede modelarse como un flujo, especificando cómo deben de colaborar entre sí las diferentes entidades participantes.
- c) No permite la ejecución paralela de tareas o actividades.
- d) No permite la realización de esperas asíncronas para la recepción de mensajes.

3. Considere los dos siguientes Partner Link Types del flujo BPEL de su práctica: a) el partnerLinkType1 define, entre otras, las operaciones de obtener la información de un pedido y cambiar el estado del mismo; b) el partnerLinkType2 define, entre otras, la operación de notificar al flujo que se ha realizado una modificación en el inventario de algún producto. Ambos Partner Link Types definen un único role con nombre "rolename". Indique la forma correcta de declarar los partners en el flujo BPEL:

- a)

```
<partnerLink name="partner1" partnerLinkType="partnerLinkType1"
  partnerRole="rolename"/>
<partnerLink name="partner2" partnerLinkType="partnerLinkType2"
  myRole="rolename"/>
```
- a)

```
<partnerLink name="partner1" partnerLinkType="partnerLinkType1"
  myRole="rolename"/>
<partnerLink name="partner2" partnerLinkType="partnerLinkType2"
  myRole="rolename"/>
```
- b)

```
<partnerLink name="partner1" partnerLinkType="partnerLinkType1"
  myRole="rolename"/>
<partnerLink name="partner2" partnerLinkType="partnerLinkType2"
  partnerRole="rolename"/>
```
- c)

```
<partnerLink name="partner1" partnerLinkType="partnerLinkType1"
  partnerRole="rolename"/>
<partnerLink name="partner2" partnerLinkType="partnerLinkType2"
  partnerRole="rolename"/>
```

4. Indica cuál de las siguientes afirmaciones referentes a actividades estructuradas de WS-BPEL es correcta:

- a) La actividad pick permite la ejecución de varias actividades en paralelo.
- b) La actividad flow permite a un proceso BPEL ejecutar uno de varios caminos alternativos, en función del evento que se produzca.
- c) La actividad sequence permite definir un conjunto de actividades a ejecutar en secuencia.
- d) Ninguna de las anteriores.

5. Es necesario crear un flujo BPEL para coordinar la invocación a diferentes partners remotos, p1, p2 y p3, de tal modo que la invocación a p1 se realice en paralelo a las invocaciones a p2 y p3 (p3 debe ejecutarse después de p2, lo antes posible). Indique la forma correcta de modelarlo teniendo en cuenta todas las posibilidades permitidas por el estándar WS-BPEL:

- a) Utilizar la actividad flow con links para indicar las dependencias entre las diferentes actividades invoke.
- b) Utilizar la actividad flow para ejecutar en paralelo una actividad invoke sobre p1 y una actividad sequence que agrupe dos actividades invoke sobre p2 y p3.
- c) Utilizar una actividad pick con links para indicar las dependencias entre las diferentes actividades invoke.
- d) a) y b).

6. Teniendo en cuenta el enunciado de la práctica, indique qué actividad inicia el conjunto de correlación y cuál la usa:

- a) La actividad invoke que obtiene la información de un pedido inicializa el conjunto de correlación, y la actividad receive que recibe la notificación de que hay más inventario, lo usa.
- b) La actividad receive que inicia el flujo inicializa el conjunto de correlación, y la actividad invoke que comprueba si hay inventario de un producto, lo usa.
- c) La actividad receive que inicia el flujo inicializa el conjunto de correlación, y la actividad receive que recibe la notificación de que hay más inventario, lo usa.
- d) La actividad invoke que obtiene la información de un pedido inicializa el conjunto de correlación, y la actividad invoke que comprueba si hay inventario de un producto, lo usa.

7. Indique cuál de las siguientes afirmaciones constituye una ventaja de la utilización de enlaces en la representación de recursos de servicios web REST:

- a) Proporcionan transparencia de localización, al ocultar detalles de implementación del servicio. El cliente debe entender la semántica del link, pero no el formato de los URLs a seguir.
- b) Permiten desacoplar a los clientes de detalles de interacción con el servicio, permitiendo que los clientes descubran nueva información sin entender las interfaces del mismo.
- c) Reducen los errores de transición de estados, al incluir solo los enlaces a los recursos alcanzables a partir de la representación de otro recurso.
- d) Todas las anteriores.

8. Suponga que está diseñando un servicio web según el enfoque REST y que desea modelar la funcionalidad de creación de múltiples críticas de una película a la vez. Sea `/movies/1/reviews` el URL que identifica el subrecurso que representa las críticas de la película con identificador 1. Nótese que es responsabilidad del servidor el asignar la clave primaria a cada nueva crítica creada. En base a los conocimientos adquiridos en la asignatura, ¿qué opción seleccionaría?

- a) PUT `/movies/1/reviews`, indicando en el cuerpo de la petición la lista de críticas.
- b) POST `/movies/1/reviews`, indicando en el cuerpo de la petición la lista de críticas. Como respuesta devolvería el código 201 (Created) y en la cabecera HTTP Location incluiría el URL que identifica el último recurso creado.
- c) POST `/movies/1/reviews`, indicando en el cuerpo de la petición la lista de críticas. Como respuesta devolvería el código 303 (See Other) y en la cabecera HTTP Location incluiría el URL a un recurso colección que contiene la lista de críticas que han sido creadas.
- d) PUT `/reviews/1`, siendo reviews un nuevo recurso expuesto por el servicio. La información de las críticas para la película con identificador 1 se especifica en el cuerpo de la petición.

9. Jersey es:

- La implementación de referencia de la API Jakarta RESTful Web Services (JAX-RS).
- Una API estándar para la creación de servicios web REST en Java, así como también de clientes.
- Una API estándar para implementar la parte cliente de servicios REST en Java.
- Una API no estándar que permite la creación de servicios web REST en Java, aunque también puede utilizarse para la creación de servicios web SOAP en Java.

10. En el contexto de Jakarta RESTful Web Services (JAX-RS), indique la afirmación correcta:

- El programador crea una clase implementando una interfaz de JAX-RS para cada recurso.
- El programador del servicio REST no tiene que ocuparse de complejidades como la recepción y envío de cabeceras HTTP, creación de Servlets o parsing de URLs.
- Se utilizan convenciones de nombrado para especificar cómo deben procesarse las peticiones sobre los recursos.
- Todas las respuestas son correctas.

11. Dado el siguiente fragmento de código, correspondiente a la clase `BookResource`, que representa a un recurso colección de libros, y suponiendo que NO utilizamos JAXB para serializar/deserializar objetos Java a/desde XML:

```
@Path("/books")
public class BookResource {

    @GET
    @Path("/{id : \\d+}")
    @Produces(MediaType.APPLICATION_XML)
    public Book findBookById(@PathParam("id") long id)
        throws InstanceNotFoundException {
        ...
    }
    ...
}
```

Y teniendo en cuenta que en caso de error el cuerpo de la respuesta se mostrará vacío y únicamente se implementa en la capa de servicios la operación `findBookById` del servicio web REST, indique cuál de las siguientes afirmaciones es correcta:

- Es necesario proporcionar un provider que implemente `MessageBodyWriter<Book>` y otro que implemente `ExceptionHandler<InstanceNotFoundException>`.
- Es necesario proporcionar un provider que implemente `MessageBodyReader<Book>` y otro que implemente `ExceptionHandler<InstanceNotFoundException>`.
- Es necesario proporcionar un provider que implemente `MessageBodyWriter<Book>` y otro que implemente `MessageBodyReader<Book>`.
- Únicamente es necesario proporcionar un provider que implemente `ExceptionHandler<InstanceNotFoundException>`.

12. En el contexto de Jakarta RESTful Web Services (JAX-RS) y Jakarta XML Binding (JAXB), indique cuál de las siguientes afirmaciones es correcta:

- Para que un servicio pueda realizar las transformaciones de objetos Java a XML de forma automática, además de anotar las clases devueltas con JAXB, es necesario generar un esquema XML a partir de las clases Java anotadas con JAXB.
- Para implementar un cliente de JAX-RS que utiliza JAXB, es necesario indicar un fichero de bindings.
- En la asignatura utilizamos un plugin de JAXB para Apache Maven para generar clases java anotadas con JAXB a partir de un esquema XML.
- Si se utiliza JAXB para anotar las clases Java devueltas por un servicio REST implementado con JAX-RS, es necesario utilizar también JAXB en la implementación de un cliente Java con JAX-RS para dicho servicio.

13. Dada la siguiente clase anotada con Jakarta XML Binding (JAXB):

```
@XmlRootElement(name = "book-instance")
@XmlType(name = "bookType", propOrder = { "title", "author" })
public class Book {

    @XmlAttribute(required = true)
    private long id;
    @XmlElement(name = "book-title", required = true)
    private String title;
    @XmlElement
    private String author;

    // Constructores, métodos get/set y método toString()
    ...
}
```

Indique cuál de los siguientes esquemas se generaría al ejecutar sobre ella el generador de esquemas de JAXB:

- a)
- ```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema ...>
 <xs:element name="book-instance" type="tns:book"/>

 <xs:complexType name="book">
 <xs:sequence>
 <xs:element name="book-title" type="xs:string"/>
 <xs:element name="author" type="xs:string"/>
 </xs:sequence>
 <xs:attribute name="id" type="xs:long" use="required"/>
 </xs:complexType>
</xs:schema>
```
- b)
- ```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema ...>
  <xs:element name="book" type="tns:bookType"/>

  <xs:complexType name="bookType">
    <xs:sequence>
      <xs:element name="book-title" type="xs:string" minOccurs="0"/>
      <xs:element name="author" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:long" use="required"/>
  </xs:complexType>
</xs:schema>
```
- c)
- ```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema ...>
 <xs:element name="book-instance" type="tns:bookType"/>

 <xs:complexType name="bookType">
 <xs:sequence>
 <xs:element name="author" type="xs:string" minOccurs="0"/>
 <xs:element name="book-title" type="xs:string" minOccurs="0"/>
 </xs:sequence>
 <xs:attribute name="id" type="xs:long" use="required"/>
 </xs:complexType>
</xs:schema>
```



```
d)
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema ...>
 <xs:element name="book-instance" type="tns:bookType"/>
 <xs:complexType name="bookType">
 <xs:sequence>
 <xs:element name="book-title" type="xs:string"/>
 <xs:element name="author" type="xs:string" minOccurs="0"/>
 </xs:sequence>
 <xs:attribute name="id" type="xs:long" use="required"/>
 </xs:complexType>
</xs:schema>
```

14. Supóngase una clase `CustomerResource` que representa a un recurso colección de clientes, a la que se quiere añadir una operación `addCustomer` que nos permita crear un nuevo cliente (clase `CustomerDtoJaxb`). Además, la clase `CustomerDtoJaxb` se ha anotado con JAXB y se desea que la operación sea capaz de crear clientes utilizando tanto representaciones XML como JSON. Indique cuál de las siguientes alternativas sería correcta en el contexto de JAX-RS:

- a)
- ```
@POST
@Consumes({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
@Produces({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
public CustomerDtoJaxb addCustomer(@FormParam CustomerDtoJaxb customer, @Context
final UriInfo ui) { ... }
```
- b)
- ```
@POST
@Consumes({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
@Produces({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
public CustomerDtoJaxb addCustomer(CustomerDtoJaxb customer, @Context final
UriInfo ui) { ... }
```
- c)
- ```
@POST
@Consumes({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
@Produces({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
public CustomerDtoJaxb addCustomer(@QueryParam CustomerDtoJaxb customer, @Context
final UriInfo ui) { ... }
```
- d)
- ```
@PUT
@Consumes({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
@Produces({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
public CustomerDtoJaxb addCustomer(CustomerDtoJaxb customer, @Context final
UriInfo ui) { ... }
```

15. Teniendo en cuenta las recomendaciones explicadas en la asignatura para el modelado de enlaces hipertexto en servicios JAX-RS, indique cuál de las afirmaciones es correcta:

- a) Recomendamos utilizar la clase `Link` de JAX-RS tanto para enlaces navegacionales en las cabeceras HTTP de las respuestas como para enlaces estructurales en la representación de los recursos.
- b) Recomendamos utilizar la clase `Link` de JAX-RS para enlaces navegacionales en las cabeceras HTTP de las respuestas, pero una clase Java adecuadamente anotada con JAXB para enlaces estructurales en la representación de los recursos.
- c) Recomendamos utilizar la clase `Link` de JAX-RS para enlaces estructurales en la representación de los recursos, pero una clase Java adecuadamente anotada con JAXB para enlaces navegacionales en las cabeceras HTTP de las respuestas.
- d) Recomendamos utilizar una clase Java adecuadamente anotada con JAXB tanto para enlaces navegacionales en las cabeceras HTTP de las respuestas como para enlaces estructurales en la representación de los recursos, en lugar de utilizar la clase `Link` de JAX-RS.