

Completado

Estado Finalizado

Finalizado en

viernes, 26 de mayo de 2023, 16:58

Tiempo empleado

1 hora

Calificación

1,00 de 1,50 (67%)

Pregunta 1

Finalizado

Puntuación

0,00 sobre 0,10

Marcar pregunta

¿Qué patrón de diseño de arquitecturas basadas en microservicios tiene como misión conocer si un servicio de la arquitectura está activo y funcionando correctamente?

Respuesta:

Patrón para pruebas

Pregunta 2

Incorrecta

Puntuación

0,00 sobre 0,10

Marcar pregunta

En una empresa usan Glassfish para desplegar sus aplicaciones web. Escribe el nombre de un Framework de desarrollo web que podamos usar para desarrollar una aplicación web para esta empresa.

Respuesta:

Edpsa

Pregunta 3

Finalizado

Puntuación

0,10 sobre 0,10

Marcar pregunta

En el modelo "push" del patrón de configuración externa para arquitecturas basadas en microservicios, ¿Cómo se le pueden pasar los datos de configuración a los servicios?

Se le pasan mediante variables de entorno.

Pregunta 4

Finalizado

Puntuación

0,10 sobre 0,10

Marcar pregunta

¿Cuándo es obligatorio el uso de la anotación @Temporal de JPA?

Cuando tengamos que usar tipo Date (de java.util.Date o java.util.Calendar) habrá que usar la anotación @Temporal delante y especificar el tipo de valor que contendrá, por ejemplo: @Temporal(TemporalType.DATE) vía antes de la variable private Date nacimiento;

Pregunta 5

Correcta

Puntuación

0,10 sobre 0,10

Marcar pregunta

¿Qué anotación de JPA se puede usar para cambiar el nombre de la columna del discriminador en la estrategia de herencia de tablas vinculadas?

Respuesta:

@DiscriminatorColumn

Pregunta 6

Incorrecta

Puntuación

0,00 sobre 0,10

Marcar pregunta

¿Qué anotación de Spring Web MVC permite señalar el método que debe responder a una petición DELETE de un endpoint en una API RESTful?

Respuesta:

@RestController

Pregunta 7

Correcta

Puntuación

0,10 sobre 0,10

Marcar pregunta

Dadas las dos clases entidad de abajo, ¿quién es la propietario de la relación entre ambas?

```
@Entity
public class A {
    @Id
    private Long id;
    // Getters and setters...
}
```

```
@Entity
public class B {
    @Id
    private Long id;
    @ManyToOne
    private A a;
    // Getters and setters...
}
```

☒ a. La clase B

☐ b. La clase A

Pregunta 8

Finalizado

Puntuación

0,10 sobre 0,10

Marcar pregunta

Indique un principio de las arquitecturas orientadas a servicios y explique.

Autonomía: los servicios de las SOA (Service Oriented Architecture) tienen control total sobre su función, siendo independientes de cualquier tecnología (aunque pueden usar algunas, como por ejemplo, la Web)

Pregunta 9

Finalizado

Puntuación

0,00 sobre 0,10

Marcar pregunta

Dadas las siguientes clases que se usan para almacenar en una base de datos la información de los empleados de una empresa:

```
@Entity
public class Empleado {
    @Id
    private Long id;
    private String nombre;
    private String primerApellido;
    private String segundoApellido;
    @Embedded
    private Direccion domicilio;
    // Getters and setters ...
}
```

```
@Embeddable
public class Direccion {
    private String ciudad;
    private Long codigoPostal;
    private String detalles1;
    private String detalles2;
    private String pais;
    // Getters and setters...
}
```

Escriba una consulta JPQL en la plantilla de código de abajo (sustituya el texto resaltado en amarillo) para obtener todos los empleados que viven en la población indicada por la variable ciudad usando parámetros con nombre.

```
public List<Empleado> empleadosPorCiudad(String ciudad) {
    TypedQuery<Empleado> query = entityManager.createQuery("SELECT e FROM Empleado e WHERE e.domicilio.ciudad = :ciudad", Empleado.class);
    query.setParameter("ciudad", ciudad);
    return query.getResultList();
}
```

Pregunta 10

Finalizado

Puntuación

0,10 sobre 0,10

Marcar pregunta

¿Qué efecto tiene la anotación @Transactional cuando se aplica a una clase de servicio en Spring?

El efecto es que los métodos se ejecutarán en una transacción. Se escribe después del @Service y antes del nombre de la clase Java.

Pregunta 11

Finalizado

Puntuación

0,10 sobre 0,10

Marcar pregunta

¿Para qué sirve la anotación @RequestBody en el contexto de los controladores REST de Spring?

La anotación @RequestBody deserializa el cuerpo de la petición a un objeto Java.

Pregunta 12

Incorrecta

Puntuación

0,00 sobre 0,10

Marcar pregunta

¿Qué niveles de la arquitectura de aplicaciones empresariales se ejecutan fuera de un servidor de aplicaciones?

Respuesta:

Pregunta 13

Correcta

Puntuación

0,10 sobre 0,10

Marcar pregunta

¿Qué nivel de aislamiento de transacciones puede conseguirse mediante el uso de atributos anotados con @Version en las entidades JPA?

Respuesta:

Lectura repetible

Pregunta 14

Finalizado

Puntuación

0,10 sobre 0,10

Marcar pregunta

Sea la clase

```
@XmlRootElement
public class Persona {
    private String nombre;
    private String apellidos;
    private String nacimiento;
}
```

Creamos un objeto de dicha clase con los siguientes valores para sus atributos:

nombre = "Tini"

apellidos = "Berners-Lee"

nacimiento = "76/6/1955"

Escriba el documento XML generado al serializar dicho objeto.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes">
<Persona>
<nombre>Tini</nombre>
<apellidos>Berners-Lee</apellidos>
<nacimiento>76/6/1955</nacimiento>
```

Pregunta 15

Finalizado

Puntuación

0,10 sobre 0,10

Marcar pregunta

Escriba una expresión Java con ayuda de la clase ResponseEntity para construir una respuesta HTTP con código 200 (OK).

return ResponseEntity.ok().build();