

Examen ASO B1

1. ¿Cuál es el comando utilizado para deshacer el último commit en git?
 - a) **git reset**
 - b) git revert
 - c) git amend
 - d) git checkout
2. ¿Cuál es la diferencia entre una clase abstracta y una interfaz en java 8?
 - a) Una clase abstracta puede contener implementaciones de métodos, mientras que una interfaz no puede.
 - b) **Una clase abstracta puede contener variables de instancia, mientras que una interfaz no puede.**
 - c) Una interfaz puede contener implementaciones de métodos, mientras que una clase abstracta no puede.
 - d) Una interfaz solo puede heredar de una clase, mientras que una clase abstracta puede heredar de múltiples interfaces.
3. De los siguientes, ¿Qué tipos de declaraciones se deben usar para contar la cantidad de monedas de cinco centavos en una matriz de cadenas de varias monedas?
 - a) Conditional
 - b) Assertion
 - c) Assignment
 - d) **Iteration**
4. ¿Qué es un archivo JAR en java?
 - a) Un archivo que contiene un archivo de configuración Maven
 - b) Un archivo que contiene un archivo de configuración Git.
 - c) **Un archivo que contiene una clase Java compilada**
 - d) Un archivo que contiene un archivo de configuración de Spring
5. ¿Qué es la sobrecarga de métodos en Java?
 - a) Cuando un método tiene múltiples definiciones con el mismo nombre y tipo de parámetros.
 - b) Cuando un método tiene múltiples definiciones con diferentes nombres y cantidades de parámetros.
 - c) **Cuando un método tiene múltiples definiciones con diferentes tipos y cantidades de parámetros.**
 - d) Cuando un método tiene múltiples definiciones con diferentes nombres y tipos de parámetros.
6. ¿Cuál es la diferencia entre un ArrayList y un LinkedList en Java?
 - a) ArrayList es más rápido que un LinkedList para agregar y eliminar elementos.
 - b) ArrayList es más eficiente en el uso de memoria que LinkedList.
 - c) LinkedList es una clase abstracta mientras que ArrayList es una clase concreta.
 - d) **LinkedList es más rápido que ArrayList para acceder a elementos aleatorios.**
7. ¿Cuándo se debe usar un bloque finally en una declaración try regular (no un try-with-resources)?
 - a) **Cuando no hay bloques catch en una declaración try.**
 - b) Nunca.
 - c) Cuando hay dos o más bloques catch en una sentencia try.
 - d) Cuando hay exactamente un bloque catch en una sentencia try.

e) Cuando el código del programa no termina por sí solo.

8. ¿Cuál es el propósito principal de los test unitarios?

- a) Comprobar la eficiencia del hardware.
- b) Medir la velocidad de la aplicación.
- c) Ahorrar tiempo en el desarrollo.
- d) Asegurar la calidad del software.

9. Selecciona la respuesta correcta con respecto al resultado del bloque de código.

```
public class Test3 {  
    public static void main(String[] args) {  
        String cad1 = "hola";  
        String cad2 = new String( original: "hola");  
        String cad3 = "hola";  
  
        if (cad1 == cad2)  
            System.out.println("ca1 es igual a cad2");  
        else System.out.println("cad1 diferente a cad2");  
  
        if (cad1 == cad3)  
            System.out.println("cad1 es igual a cad3");  
        else  
            System.out.println("cad1 diferente a cad3");  
    }  
}
```

- a) cad1 diferente a cad2
cad1 es igual a cad3
- b) ca1 es igual a cad2
ca1 es igual a cad3
- c) No compila
- d) cad1 diferente a cad2
cad1 diferente a cad3

10. ¿Cuál es la salida al ejecutar el siguiente código?

```
1: class Mammal {  
2:     public Mammal(int age) {  
3:         System.out.print("Mammal");  
4:     }  
5: }  
6: public class Platypus extends Mammal {  
7:     public Platypus() {  
8:         System.out.print("Platypus");  
9:     }  
10:     public static void main(String[] args) {  
11:         new Mammal(5);  
12:     }  
13: }
```

- a) Mammal.
- b) MammalPlatypus.
- c) El código no se compila en la línea 11.
- d) El código no compila en la línea 7, ya que no se hace una llamada correcta al constructor de Mammal en la clase Platypus.

11. ¿Cómo se manejan las excepciones en java?

- a) Con la instrucción try-catch.
- b) Con la instrucción if-else.
- c) Con la instrucción for.
- d) Con la instrucción while.

12. ¿La anotación @Ignore es usada para omitir un test por lo que no se ejecuta?

- a) Verdadero
- b) Falso

13. ¿Cuál es el resultado de compilar y ejecutar el siguiente código?

```
public class Tester {  
    static {  
        int x = 3;  
    }  
    2 usages  
    static int x;  
    public static void main(String[] args) {  
        x--; // line 7  
        System.out.println(x);  
    }  
}
```

- a) Error de compilación en la línea 7, x no se inicializa.
- b) -1
- c) -2
- d) 0

14. ¿Qué es un operador de short circuit?

- a) Es un patrón de arquitectura de microservicios que nos permite evitar el consumo de servicios que están en mantenimiento.
- b) Sirve para realizar más eficientes las operaciones condicionales evitando ejecutar operaciones si estas ya no son necesarias.
- c) Sirve para lanzar una excepción personalizada.
- d) Operador que nos sirve para crear una nueva clase anónima.

15. ¿Qué es el patrón de diseño DAO y cómo se implementa en Java?

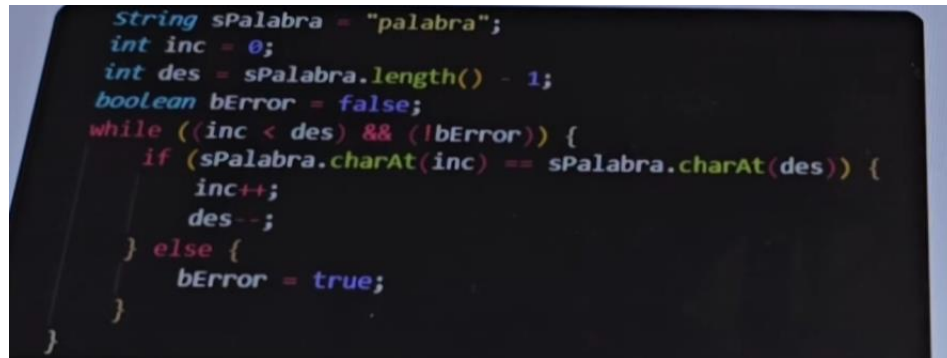
- a) El patrón de diseño DAO es un patrón que se utiliza para abstraer la capa de negocios de una aplicación. Se puede implementar en Java utilizando clases abstractas y métodos estáticos.
- b) El patrón de diseño DAO es un patrón que se utiliza para abstraer la capa de presentación en una aplicación. Se puede implementar en Java utilizando interfaces y clases concretas.

- c) El patrón de diseño DAO es un patrón que se utiliza para abstraer la capa de acceso a datos en una aplicación. Se puede implementar en Java utilizando interfaces y clases concretas.
- d) El patrón de diseño DAO es un patrón que se utiliza para abstraer la capa de infraestructura en una aplicación. Se puede implementar en Java utilizando excepciones y bloques try-catch.

16. ¿Qué es un endpoint en una API REST?

- a) Un endpoint es la URL que se utiliza para acceder a una API REST.
- b) Un endpoint es un método que se utiliza para procesar datos en una API REST.
- c) Un endpoint es un controlador que se utiliza para administrar una API REST.
- d) Un endpoint es un objeto que se utiliza para almacenar datos en una API REST.

17. ¿Qué hace el siguiente programa?



```
String sPalabra = "palabra";
int inc = 0;
int des = sPalabra.length() - 1;
boolean bError = false;
while ((inc < des) && (!bError)) {
    if (sPalabra.charAt(inc) == sPalabra.charAt(des)) {
        inc++;
        des--;
    } else {
        bError = true;
    }
}
```

- a) El programa no compila.
- b) Cuenta las letras que hay en una palabra.
- c) Verifica si una palabra es un palíndromo.

18. ¿Cuál de las siguientes opciones son verdaderas? (elijas todas las correctas)

- a) Java es un lenguaje orientado a objetos.
- b) El código Java compilado en Windows puede ejecutarse en Linux.
- c) Java permite la sobrecarga de operadores
- d) Java es un lenguaje de programación funcional.
- e) Java es un lenguaje procedimental.
- f) Java tiene punteros a ubicaciones específicas en la memoria.

19. ¿Qué es Maven y para qué se utiliza en el desarrollo de aplicaciones?

- a) Maven es un lenguaje de programación. Se utiliza en el desarrollo de aplicaciones Java para escribir...
- b) Maven es un servidor de base de datos. Se utiliza en el desarrollo de aplicaciones java para alojar...
- c) Maven es un sistema de control de versiones. Se utiliza en el desarrollo de aplicaciones java para...
- d) Maven es una herramienta de gestión de dependencias. Se utiliza en el desarrollo de aplicaciones...

20. ¿Cuál de lo siguiente es cierto? (elijas todas las correctas)

- a) javac compila un archivo .java en un archivo .bytecode.
- b) Java toma el nombre del archivo .bytecode como parámetro.
- c) javac compila un archivo .java en un archivo .class
- d) Java toma el nombre de la clase como parámetro.

e) Java toma el nombre del archivo .class como parámetro.

f) javac compila un archivo .class como archivo .java.

21. ¿Qué es Git y cuáles son algunos de sus comandos básicos?

a) Git es un lenguaje de programación. Algunos comandos básicos de Git incluyen “print” e “if-else”.

b) Git es una herramienta para el análisis de código. Algunos comandos básicos de Git incluyen “analyze” y “debug”.

c) Git es un sistema de control de versiones. Algunos comandos básicos de Git incluyen “commit” y “push”.

d) Git es una herramienta para realizar pruebas de software. Algunos comandos básicos de Git incluyen “run” y “test”.

22. Dados los siguientes segmentos de código, ¿Qué respuesta no es una implementación de java válida?

a) `int variableA = 10;`

`float variableB = 10.5f;`

`int variableC = variableA + variableB;`

b) `byte variableA = 10;`

`double variableB = 10.5f;`

`double variableC = variableA + variableB;`

c) `byte variableA = 10;`

`float variableB = 10.5f;`

`float variableC = variableA + variableB;`

23. ¿Qué escenario es el mejor uso de una excepción?

a) La computadora se incendió.

b) No sabe cómo codificar un método.

c) No se encuentra un elemento al buscar en una lista.

d) Se pasa un parámetro inesperado a un método.

e) Quiere recorrer una lista.

24. ¿Qué es un bean en Spring?

a) Una herramienta de inyección de dependencias que se utiliza para inyectar dependencias en una clase.

b) Una instancia de una clase que se administra por el contenedor de Spring.

c) Un archivo de configuración XML que se utiliza para definir la estructura de una tabla de base de datos.

d) Una clase que se utiliza para configurar la conexión a una base de datos.

25. Selecciona la respuesta con respecto al resultado del bloque de código.

```
public class Test1 extends Concreate{
    1 usage
    Test1(){
        System.out.println("t ");
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        new Test1();
    }
}

1 usage 1 inheritor
class Concreate extends Send{
    1 usage
    Concreate(){
        System.out.println("c ");
    }
    private Concreate(String s){

    }
}

1 usage 2 inheritors
abstract class Send{
    2 usages
    Send(){
        System.out.println("s ");
    }
}
```

- a) t,s,c
- b) Error en tiempo de ejecución.
- a) s, c, t

26. ¿Cuáles de las siguientes afirmaciones sobre el polimorfismo son verdaderas? (Elija todas las correctas).

- a) Un método que toma un parámetro con tipo java.lang.Object tomará cualquier referencia.
- b) Una referencia a un objeto se puede convertir a una subclase del objeto sin una conversión explícita.
- c) Todas las excepciones de conversión se pueden detectar en tiempo de compilación.
- d) Al definir un método de instancia pública en la superclase, garantiza que el método específico se llamará al método en la clase principal en tiempo de ejecución.
- e) Si un método toma una superclase de tres objetos, cualquiera de esas clases puede pasarse como parámetro del método.

27. Son patrones de diseño de software estructural

- a) Adapter, Proxy, Prototype y Bridge.
- b) Adapter, Bridge, Proxy y Composite.

- c) Agile, Builder, Singleton y Prototype.
- d) Builder, Singleton y Prototype y Abstract Factory.

28. Seleccione la respuesta que considere correcta dado el siguiente bloque de código.

```
import java.util.Arrays;
import java.util.List;

public class Example {
    public static void main(String[] args) {
        List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);
        double result = numbers.stream()
            .mapToInt(n -> n)
            .average()
            .orElse(0);
        System.out.println(result);
    }
}
```

- a) 3.0
- b) 2.5
- c) 1
- d) 5

29. ¿Qué son las pruebas de integración?

- a) Pruebas que comprueben el rendimiento de la aplicación.
- b) Pruebas que comprueben el funcionamiento de la interfaz de usuario.
- c) Pruebas que comprueban el funcionamiento de varias unidades de código juntas.
- d) Pruebas que comprueban el funcionamiento de una sola unidad de código.

30. ¿Qué comando se utiliza para enviar los cambios confirmados en un repositorio local al repositorio remoto?

- a) git push
- b) it pull
- c) git commit
- d) git add

32. ¿Cuál es el comando utilizado para crear una nueva rama en Git?

- a) git branch
- b) git merge
- c) git commit
- d) git push

33. ¿Cuál es el resultado de compilar la siguiente clase?

```
public class Book {
    3 usages
    private int ISBN;
    private String title, author;
    private int pageCount;

    public int hashCode(){
        return ISBN;
    }

    public boolean equals(Object obj){
        if(!(obj instanceof Book)){
            return false;
        }
        Book other = (Book) obj;
        return this.ISBN == other.ISBN;
    }
}
```

- a) Línea 15 no compila porque other.ISBN es un atributo con modificador de acceso private.
- b) Línea 14 no compila porque no está declarada o manejada ClassCastException
- c) La clase compila satisfactoriamente.

34. ¿Cuál es la primera línea en fallar al compilar?

```
class Tool {
    private void repair(){} //r1
    1 override
    void use(){}
}

class Hammer extends Tool{
    private int repair(){return 0;} //r3
    private void use(){} //r4
    public void bang(){} //r5
}
```

- a) r5
- b) r4
- c) r3
- d) Ninguna de las anteriores.

35. ¿Qué es Git?

- a) Una herramienta de automatización de compilación que se utiliza para compilar y construir un proyecto.
- b) Una herramienta de gestión de software de dependencias que se utiliza para descargar bibliotecas y paquetes en un proyecto de Java.
- c) Una herramienta de generación de informes que se utiliza para generar informes sobre el rendimiento de una aplicación.
- d) Una herramienta de control de versiones que se utiliza para almacenar y administrar el código fuente de un proyecto.

36. ¿Cuál de las siguientes excepciones lanza la JVM? (Elija todas las correctas)

- a) `ArrayIndexOutOfBoundsException`
- b) `NumberFormatException`
- c) `ExceptionInInitializerError`
- d) `Java.io.IOException`
- e) `NullPointerException`

38. ¿Cuál es el comando utilizado para fusionar una rama en Git?

- a) `git Branch`
- b) `git merge`
- c) `git push`
- d) `git pull`

39. ¿Qué es REST y cuál es su relación con las API web?

- a) REST es un protocolo de comunicación. Su relación con las API web es que se utiliza para definir los endpoints de una API.
- b) REST es un estilo de arquitectura para aplicaciones web. Su relación con las API web es que se utiliza para definir la estructura y funcionalidades de una API
- c) REST es un servicio en la nube. Su relación con las API web es que se utiliza para alojar las aplicaciones web.
- d) REST es un lenguaje de programación. Su relación con las API web es que se utiliza para crear aplicaciones web.

40. ¿Cuál es el comando utilizado para actualizar la rama local con los cambios de la rama remota en Git?

- a) `git checkout`
- b) `git clone`
- c) `git push`
- d) `git pull`

Examen ASO E

1. ¿Qué es un “code smell”?

- a) Un componente de la biblioteca estándar de Java
- b) Un error en tiempo de compilación que se produce en Java
- c) Un indicador de que puede haber un problema en el código que puede ser difícil de detectar o que podría ser una fuente potencial de errores o problemas de mantenimiento en el futuro.
- d) Una práctica de programación recomendada en Java.

2. ¿Cuál de las siguientes afirmaciones son verdaderas? (Elije todas las correctas)

- a) Las excepciones de tiempo de ejecución son lo mismo que las excepciones comprobadas (checked).
- b) Las excepciones en tiempo de ejecuciones son lo mismo que las excepciones no comprobadas (unchecked).
- c) Solo pueden manejar subclases de exception.
- d) Solo puede declarar excepciones comprobadas (checked).
- e) Puede declarar solo excepciones no comporbadas (unchecked).

3. ¿Cuáles son las salidas del siguiente código? (Elije todas las correctas)

```
public class StrinBuilders {  
    1 usage  
    public static StringBuilder work(StringBuilder a, StringBuilder b){  
        a = new StringBuilder("a");  
        b.append("b");  
        return a;  
    }  
  
    public static void main(String[] args) {  
        StringBuilder s1 = new StringBuilder("s1");  
        StringBuilder s2 = new StringBuilder("s2");  
        StringBuilder s3 = work(s1,s2);  
        System.out.println("s1 = " + s1);  
        System.out.println("s2 = " + s2);  
        System.out.println("s3 = " + s3);  
    }  
}
```

- a) El código no compila.
- b) s3 = a
- c) s1 = a
- d) s2 = s2
- e) s1 = s1
- f) s3 = null
- g) s2 = s2b

6. ¿Cuál es la función del operador de doble dos puntos (::) en Java 8?

- a) El operador doble dos puntos se utiliza para crear una nueva instancia de una clase en Java 8.
- b) EL operador de doble dos puntos no se utiliza en java 8.
- c) El operador de doble dos puntos se utiliza para acceder a métodos estáticos en Java 8.
- d) El operador de doble dos puntos se utiliza para acceder a métodos no estáticos en Java 8.

7. ¿Qué significa el acrónimo CRUD en una API REST?

- a) Code, Register, Update, Debug
- b) Create, Read, Update, Delete
- c) Call, Receive, Use, Debug
- d) Customize, Request, Use, Debug

10. ¿Para qué nos sirve utilizar un profile dentro del archivo pom.xml?

- a) Etiqueta por la cual podemos definir las versiones de nuestras dependencias.
- b) Es la etiqueta por la cual podemos definir las características que tendrá nuestro proyecto al ser compiladas.
- c) Etiqueta por la cual definimos los parámetros de conexión a un repositorio.
- d) No existe esta etiqueta en Maven.

12. Dadas las siguientes clases Vehicle y Car

```
package my.vehicles;

public class Vehicle {
    public String make;
    protected String model;
    private int year;
    int mileage;
}

package my.vehicles.cars;

import my.vehicles.*;

public class Car extends Vehicle {
    public Car() {
        //línea 7
    }
}
```

¿Cuál de las siguientes instrucciones pueden colocarse en la línea 7 para que la clase Car compile correctamente? (Seleccione las que apliquen)

- a) mileage = 15285;
- b) Ninguna de las anteriores.
- c) make = "Honda";
- d) year = 2009;
- e) model = "Pilot";

13. Enumere cuatro interfaces de la API colecciones

- a) List, Map, Set, Queue.
- b) ArrayList, Map, Set, Queue.
- c) List, HashMap, HashSet, PriorityQueue.
- d) List, Map, HashSet, PriorityQueue.

15. Selecciona la respuesta correcta con respecto al resultado del bloque de código.

```
public class Test5 {  
    public static void main(String args[]) {  
        Side primerIntento = new Head();  
        Tail segundoIntento = new Tail();  
        Coin.overload(primerIntento);  
        Coin.overload((Object)segundoIntento);  
        Coin.overload(segundoIntento);  
        Coin.overload((Side)primerIntento);  
    }  
}  
  
interface Side { String getSide();}  
  
class Head implements Side {  
    public String getSide() { return "Head ";}  
}  
  
class Tail implements Side {  
    public String getSide() { return "Tail ";}  
}  
  
class Coin {  
    public static void overload(Head side) {System.out.println(side.getSide());}  
    public static void overload(Tail side) {System.out.println(side.getSide());}  
    public static void overload(Side side) {System.out.println("Side ");}  
    public static void overload(Object side) {System.out.println("Object ");}  
}
```

- a) Head
Object
Tail
Side
- b) No compila
- d) Side
Object
Tail
Side
- e) Head
Head
Tail
Tail
- e) Side
Head
Tail
Side

16. ¿Cuál es la salida al ejecutar el siguiente código?

```
public class Lion {  
    public void roar(String roar1, StringBuilder roar2) {  
        roar1.concat("!!!");  
        roar2.append("!!!");  
    }  
    public static void main(String[] args) {  
        String roar1 = "roar";  
        StringBuilder roar2 = new StringBuilder("roar");  
        new Lion().roar(roar1, roar2);  
        System.out.println(roar1 + " " + roar2);  
    }  
}
```

- a) roar roar!!!
- b) roar!!! Roar
- c) Se lanza una excepción
- d) roar!!! roar!!!
- e) roar roar
- f) El código no compila

17. ¿Cuál de los siguientes es cierto acerca de una subclase concreta?

- a) Una subclase concreta no se puede marcar como final.
- b) Una subclase concreta debe implementar todos los métodos definidos en una interfaz heredada.
- c) Una subclase concreta debe implementar todos los métodos abstractos heredados.
- d) Una subclase concreta puede declararse como abstracta.
- e) Los métodos abstractos no pueden ser anulados por una subclase concreta.

18. ¿Cuál es la salida del siguiente código?

```
1 public abstract class Catchable {
2     protected abstract void catchAnObject(Object x);
3
4     public static void main(String [] args) {
5         java.util.Date now = new java.util.Date();
6         Catchable target = new MyStringCatcher();
7         target.catchAnObject(now);
8     }
9 }
10
11 class MyStringCatcher extends Catchable {
12     public void catchAnObject(Object x) {
13         System.out.println("Caught object");
14     }
15
16     public void catchAnObject(String s) {
17         System.out.println("Caught string");
18     }
19 }
```

- a) Error de compilación línea 12
- b) Error compilación línea 16
- c) Caught string
- d) Error compilación línea 2
- e) Caught object

19. Seleccione la respuesta que considere correcta, dado el siguiente bloque de código.

```
1 import java.util.Arrays;
2 import java.util.List;
3
4 public class Example {
5
6     public static void main(String[] args) {
7         List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);
8
9         int result = numbers.stream()
10             .filter(n -> n % 2 == 0)
11             .reduce(0, (a, b) -> a + b);
12
13         System.out.println(result);
14     }
15 }
16 }
```

- a) 3
- b) 9
- c) 14
- d) 6

20. ¿Qué declaración representa una declaración válida que permitirá la inclusión de clases del paquete java.util?

- a) #include java.util.*;
- b) #include java.util;
- c) import java.util.*;
- d) import java.util;

21. ¿Qué es la cobertura de código?

- a) La cantidad de veces que se ejecuta una línea de código.
- b) La cantidad de errores detectados por una prueba.
- c) La cantidad de código que se ejecuta durante una prueba.
- d) La cantidad de tiempo que tarda una prueba en ejecutarse.

23. ¿Cuál es el formato correcto para hacer un commit en Git?

- a) Descripción breve del cambio y nombre del autor.
- b) Tipo de cambio, descripción breve, cuerpo opcional y notas de pie de página.
- c) Solo se necesita una breve descripción del cambio.
- d) Nombre de la rama, descripción detallada del cambio y fecha.

24. ¿Qué es el patrón de diseño Singleton y cómo se implementa en Java 8?

- a) El patrón de diseño Singleton es un patrón que se utiliza para garantizar que una clase tenga una única instancia en todo el sistema. Se implementa utilizando una variable estática y un constructor privado.
- b) El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de infraestructura en una aplicación. Se implementa utilizando excepciones y bloques try-catch.
- c) El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de presentación en una aplicación. Se implementa con interfaces y clases concretas.
- d) El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de negocios en una aplicación. Se implementa utilizando clases abstractas y métodos estáticos.

26. ¿Qué es un Microservicio?

- a) Ninguna de las anteriores
- b) Son componentes que se pueden desplegar de forma independiente en función múltiple, es decir, englobando endpoint que no necesariamente están relacionados.
- c) Es el conjunto de endpoints contenidos en múltiples desarrollos que se despliegan en conjunto y que están estrechamente relacionados.
- d) Son componentes que se pueden desplegar de forma independiente y que suelen ser de función única, es decir, englobando endpoint y están estrechamente relacionandos.

27. ¿Cuál no es un objetivo de Maven?

- a) Debug
- b) Clean

- c) Package
- d) Install

28. En los verbos REST ¿Cuál es la diferencia en el uso de PATCH y PUT?

- a) Son exactamente iguales, no hay diferencia de uso.
- b) PATCH requiere se le envíe la entidad completa mientras que PUT solo los atributos a modificar.
- c) PUT requiere se le envíe la entidad completa mientras que PATCH solo los atributos a modificar.
- d) PATCH es un verbo deprecado sustituido por PUT.

29. ¿Qué es una expresión lambda en Java 8?

- a) Una expresión lambda es una forma de escribir una clase anónima en Java 8.
- b) Una expresión lambda es una forma de escribir una función anónima en Java 8.
- c) Una expresión lambda es un método que se llama automáticamente cuando se crea un objeto.
- d) Una expresión lambda es un método que se llama de forma explícita desde el código.

31. ¿Cuál es la diferencia entre las anotaciones: @RestController, @Component, @Service y @Repository?

- a) @Controller es una anotación que nos ayuda a construir una API REST mientras que @Service, @Component y @Repository solo marcan las clases que se deben de inicializar.
- b) @Controller, @Component son anotaciones que crean bean y exponen la serialización de las clases mientras que @Service y @Repository requieren de una inicialización manual.
- c) No existe diferencia funcional entre ellas sino semántica, las 4 son anotaciones de Spring que crean un bean y lo agregan al contexto de Spring.
- d) @Service y @Repository son anotaciones que crean bean y exponen la serialización de las clases mientras que @Controller. @Component requiere de una inicialización manual.

32. ¿Cuál es una buena práctica al escribir pruebas unitarias?

- a) Ejecutar pruebas con poca frecuencia.
- b) Asegurarse de que las pruebas sean claras y concisas.
- c) Probar solo una pequeña parte de una función.
- d) Hacer que las pruebas dependan de otras pruebas.

33. ¿Cuál es la ventaja de usar APIs REST sobre otros tipos de servicios web?

- a) Mayor seguridad.
- b) Mayor facilidad de implementación.
- c) Mayor velocidad de transferencia de datos.
- d) Mayor compatibilidad con diferentes plataformas.

34. Selecciona la respuesta correcta con respecto al resultado del bloque de código.

```
public class Test4 {  
  
    public static void main(String[] args) {  
        List list = Arrays.asList(25,7,25,67);  
        System.out.println(list);  
        System.out.println(new HashSet(list));  
        System.out.println(new TreeSet(list));  
        System.out.println(new HashSet(list));  
        System.out.println(new ConcurrentSkipListSet(list));  
    }  
}
```


- a) No compila
- b) [25, 7, 25, 67]
[67, 7, 25]
[7, 25, 67]
[67, 7, 25]
[7, 25, 67]
- c) [25, 7, 67]
[67, 7, 25]
[7, 25, 67]
[67, 7, 25]
[7, 25, 67]
- d) [67, 7, 25]
[67, 7, 25]
[67, 7, 25]
[67, 7, 25]
[67, 7, 25]
- e) [25, 7, 25, 67]
[7, 25, 67]
[67, 7, 25]
[7, 25, 67]
[67, 7, 25]

35. ¿Cuál es la salida al ejecutar el siguiente código?

```
public class Fish {
    public static void main(String[] args) {
        int numFish = 4;
        String fishType = "tuna";
        String anotherFish = numFish + 1;
        System.out.println(anotherFish + " " + fishType);
        System.out.println(numFish + " " + 1);
    }
}
```

- a) 51tuna
- b) 5tuna
- c) 5
- d) 41
- e) 5 tuna
- f) 4 1
- g) El código no compila

36. ¿Cuáles son los 4 pilares de la programación orientada a objetos?

- a) Polimorfismo, Coerción, Herencia y Encapsulamiento.
- b) Encapsulamiento, Coerción, Polimorfismo y Abstracción.
- c) Polimorfismo, Herencia, Encapsulamiento y Sincronía.
- d) Polimorfismo, Abstracción, Herencia y Encapsulamiento.

37. ¿Cuál es el comando utilizado para ver el historial de cambios en git?

- a) git diff
- b) git status
- c) git log
- d) git commit

38. ¿Qué utilidad de línea de comandos basada en MS Windows le permitirá ejecutar el intérprete de Java sin abrir la ventana de la consola?

- a) jconsole
- b) javaw
- c) interpw
- d) java -wo