

## Exámenes Completos

1. **¿Cuál es el comando utilizado para deshacer el último commit en git?**

Existen dos comandos que pueden realizar esta acción, git revert y git reset, sin embargo, los propósitos cambian entre cada comando, git revert es útil cuando se quieren deshacer cambios en el historial de confirmaciones sin modificar el historial existente, mientras que git reset es más adecuado cuando se necesita modificar el historial de confirmaciones como eliminar confirmaciones o deshacer cambios en el árbol de trabajo.

2. **¿Cuál es la diferencia entre una clase abstracta y una interfaz en java 8?**

Una clase abstracta y una interfaz tienen propósitos similares, proporcionar un molde de un objeto sin la posibilidad de instanciar, sin embargo, la diferencia principal radica en que una clase abstracta puede contener métodos abstractos y/o métodos concretos, los cuales pueden definir algún comportamiento específico.

Desventaja de clase abstracta: La clase ya no puede heredar de ninguna otra clase, pues en Java se maneja una sola herencia

Ventaja de clases abstractas: Se pueden definir constructores en las clases abstractas

Ventaja de las interfaces: Permite la implementación de n interfaces

Desventaja de interfaces: No se pueden definir constructores en las interfaces

3. **De los siguientes ¿qué tipos de declaraciones se deben usar para contar la cantidad de monedas de 5 centavos en una matriz de cadenas de varias monedas? (Elije todas las correctas)**

4. **¿Qué es un archivo JAR en java?**

Un archivo JAR es un archivo que contiene una o más clases java empaquetadas compiladas, el archivo JAR es un tipo de archivo ejecutable.

5. **¿Qué es la sobrecarga de métodos en Java?**

La sobrecarga de métodos es cuando existe más de un método con el mismo nombre y tipo de dato que retorna, sin embargo, la diferencia debe de radicar ya sea en el orden de parámetros recibidos, en el tipo de dato de cada parámetro, o la cantidad de estos.

6. **¿Cuál es la diferencia entre un ArrayList y un LinkedList en Java?**

La diferencia entre estos es que, el ArrayList contiene los datos uno tras otro, pero el acceso puede darse de manera aleatoria accediendo al índice del elemento requerido, por otro lado, un LinkedList los datos se encuentran de la misma forma uno tras otro, pero el acceso a los datos es más lento, puesto que en un LinkedList cada elemento tiene la información de sus vecinos ligados.

7. ¿Cuándo se debe usar un bloque finally en una declaración try regular (no una prueba con recursos)?

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

Un test unitario es utilizado para probar una funcionalidad específica de la aplicación, la cual es aislada del sistema completo.

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

Debido a que la cadena1 y la cadena3 se crean a partir de una inicialización de String , estas se almacenan en el pool de strings, cadena2 se crea a partir de una instanciación, lo que esta cadena no se almacena en el pool de strings, sin embargo, lo que en realidad se compara con el operador == es la dirección de memoria.

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: }
```

Se genera un error de compilación en la línea 8, puesto que la clase padre Mammal no tiene un constructor vacío, y recordemos que cuando una clase hereda de otra clase, existe un `super()` implícito o explícito, por lo que en este caso, como está implícito, se genera el error de compilación.

11. ¿Cómo se manejan las excepciones en java?

Las excepciones en Java son manejadas por medio del bloque de código try-catch-finally, en el bloque try se inserta el código que puede generar una excepción, en caso de que esta se genere, entra el bloque catch, para tratar de manejar dicha excepción y evitar una interrupción en la ejecución del programa, el bloque finally se ejecuta se haya generado una excepción o no.

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

Si, la anotación `@Ignore`, indica que un caso de prueba específico debe de ser ignorado durante la ejecución de las pruebas en JUnit.

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);  
    }  
}
```

Debido a que dentro del bloque static se define una variable inicializada con el valor de 3, cuando se ejecuta la clase, esta se carga en memoria, sin embargo, al salir de este bloque la memoria se va a liberar, por lo que la variable dejará de existir, más abajo se define una variable x sin valor explícito, por lo que su valor inicial es de 0, al momento de ejecución, esta variable se le restará su valor en 1, por lo que el valor impreso será -1.

14. ¿Qué es un operador de short circuit?

Este es un operador lógico, el cual evita que se validen ambos lados de una expresión lógica, esto evita realizar validaciones innecesarias en ciertas situaciones, existen 2 operadores de short circuit: && (and) y || (or).

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

El patrón de diseño DAO (Data Access Object) es un patrón de diseño de software que se utiliza para abstraer y encapsular todos los accesos a datos dentro de una aplicación. Su propósito es separar la lógica de negocio de la lógica de acceso a datos, lo que facilita el cambio de la fuente de datos subyacente sin afectar a la capa de negocio. El patrón DAO proporciona una interfaz entre la lógica de negocio y la fuente de datos, lo que permite que la lógica de negocio opere en objetos de dominio sin conocer los detalles de cómo se accede o se almacenan los datos.

En Java, el patrón DAO se implementa generalmente con las siguientes partes:

Interfaz DAO (Data Access Object): Define métodos abstractos para acceder a los datos relacionados con un objeto de dominio específico. Esta interfaz proporciona una abstracción sobre la fuente de datos subyacente.

Clase DAO concreta: Implementa la interfaz DAO y proporciona la lógica concreta para acceder a los datos. Esta clase puede interactuar directamente con la base de datos u otro mecanismo de almacenamiento.

Objeto de dominio (POJO - Plain Old Java Object): Representa los datos que se manipularán en la aplicación. Esta clase generalmente no contiene lógica de negocio, solo propiedades y métodos de acceso.

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

Un endpoint es un punto de acceso específico dentro de una API, la cual le permite a los clientes (ya sea sitio web, aplicación o incluso otra API) comunicarse con la API y realizar operaciones específicas, tales como obtención de datos, actualización de recursos, eliminación de datos, entre otras.

Cada endpoint está asociado con una URL única y generalmente está vinculado a un método HTTP específico, como GET, POST, PUT o DELETE, que define la acción que se realizará en los datos.

#### 17. ¿Qué hace el siguiente programa?

```
public class Palabra {
    public static void main(String[] args) {
        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;
            }
        }
    }
}
```

El programa anterior verifica si una palabra es un palíndromo, es decir, verifica si la palabra puede leerse tanto de izquierda a derecha como de derecha a izquierda.

#### 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?**

Maven es un manejador de dependencias en proyectos Java, este permite administrar las dependencias que serán usadas en el proyecto, por lo que simplifica la creación de un proyecto, quitándole la carga de trabajo al desarrollador al dejar de preocuparse por las dependencias y sólo concentrarse en desarrollar la lógica del programa.

**20. ¿Cuál de lo siguiente es cierto? (elija 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"... (no se ve).
- 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... no se ve

**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) Un archivo de configuración XML que se utiliza para definir la estructura de una tabla de base de datos.
- b) Una instancia de una clase que se administra por el contenedor de Spring.
- c) Una herramienta de inyección de dependencias que se utiliza para inyectar dependencias en una clase.
- 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 Concrete{
    1 usage
    Test1(){
        System.out.println("t ");
    }

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

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

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

La respuesta correcta es que imprimirá s,c,t en ese orden debido a la herencia de clases.

- a) c,s,t
- b) t,s,c
- c) Error en tiempo de ejecución
- d) No compila
- e) Ninguna es correcta

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

- a) Si un método toma una superclase de 3 objetos, cualquiera de esas clases puede pasarse como parámetro del método
- b) Un método que toma un parámetro con tipo java.lang.object tomará cualquier referencia
- c) Una referencia a un objeto se puede convertir a una subclase de objetos en una conversión explícita

- d) Todas las excepciones de conversión se pueden detectar en tiempo de compilación
- e) Al definir un método de instancia pública en la súper clase, garantiza que el método específico se llamará al método en la clase principal en tiempo de ejecución

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.
1. **Adapter (Adaptador):** El patrón Adapter se utiliza para permitir que dos interfaces incompatibles trabajen juntas. Convierte la interfaz de una clase en otra interfaz que el cliente espera encontrar. Esto se logra mediante un objeto que envuelve a uno de los objetos y proporciona una interfaz estándar.
  2. **Bridge (Puente):** El patrón Bridge se utiliza para desacoplar una abstracción de su implementación, de modo que ambas puedan variar independientemente. Esto implica una estructura en la que las clases abstractas tienen una referencia a una implementación específica. Permite cambios en la implementación sin afectar la interfaz expuesta al cliente.
  3. **Proxy (Proxy):** El patrón Proxy se utiliza para proporcionar un sustituto o representante de otro objeto para controlar el acceso a él. Esto puede ser útil para realizar tareas como el control de acceso, el registro o la creación de objetos bajo demanda. El objeto proxy tiene la misma interfaz que el objeto original, permitiendo su intercambio transparente.
  4. **Composite (Compuesto):** El patrón Composite se utiliza para tratar las colecciones de objetos individuales y composiciones de objetos de manera uniforme. Permite que los clientes traten a los objetos individuales y a las composiciones de objetos de la misma manera, lo que simplifica el diseño y la manipulación de estructuras complejas. En este patrón, los objetos se organizan en una estructura de árbol para representar parte-todo jerárquico.

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) 1
- c) 5
- d) 2.5



29. **¿Qué son las pruebas de integración?**

Las pruebas de integración son aquellas que se encargan de testear los diferentes módulos por los que está conformada la aplicación, esto, para comprobar si los diferentes módulos entre sí, se comunican (integran de manera correcta).

- 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?**

El comando utilizado es “git push origin master” o dependiendo de cómo se llame la rama a la que se desean enviar los cambios.

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

31. Seleccione la respuesta correcta, dado el siguiente bloque de código.

```
class ClassX{
    7 usages
    static int y = 2;
    1 usage
    ClassX(int x){
        this();
        y = y * 2;
    }

    1 usage
    ClassX(){
        y++;
    }
}

1 usage
public class Class2 extends ClassX{
    1 usage
    Class2(){
        super(y);
        y = y + 3;
    }

    public static void main(String[] args) {
        new Class2();
        System.out.println(y);
    }
}
```

El resultado final es 9, esto es porque el flujo de ejecución es el siguiente, al entrar al main se crea una instancia de la clase Class2, este llama al constructor de su padre, quien es ClassX, el valor que manda como parámetro es el atributo y, el cual contiene un valor de 2 (definido en el padre), en este constructor, la primera instrucción es this() quien llama al constructor de esa misma clase sin parámetros, este constructor aumenta en 1 a y, por lo que y ahora vale 3, al regresar al constructor con parámetro, y se multiplica por 2, por lo que y ahora vale 6, finalmente, al regresar al constructor de Class2, se le suman 3, por lo que por esto el valor final es 9.

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

El comando para crear una nueva rama es “git branch nombreRama”.

- 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 primer línea en fallar al compilar?

```
1  class Tool {
2  private void repair() {} //r1
3  1 override
4  void use(){}
5  }
6  class Hammer extends Tool{
7  private int repair(){return 0; } //r3
8  private void use(){}//r4
9  public void bang(){}//r5
10 }
11 |
```

Falla la línea 8 //4 puesto que está reduciendo la visibilidad del método use().

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

35. ¿Qué es Git?

**Git es un controlador de versiones de código, este permite administrar las diferentes versiones de un software a lo largo de su ciclo de vida.**

- a) Una herramienta de automatización de compilación que se utiliza para compilar y construir un proyecto (no se ve lo demás).
- b) Una herramienta de gestión de software de dependencias que se utiliza para descargar bibliotecas y paquetes en... no se ve
- c) Una herramienta de generación de informes que se utiliza para generar informes sobre el rendimiento... no se ve
- d) Una herramienta de control de versiones que se utiliza para almacenar y administrar el código... no se ve.**

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**

37. ¿Cuál es la diferencia entre una clase abstracta y una interfaz en Java 8?

Una clase abstracta y una interfaz tienen propósitos similares, proporcionar un molde de un objeto sin la posibilidad de instanciar, sin embargo, la diferencia principal radica en que una clase abstracta puede contener métodos abstractos y/o métodos concretos, los cuales pueden definir algún comportamiento específico.

Desventaja de clase abstracta: La clase ya no puede heredar de ninguna otra clase, pues en Java se maneja una sólo herencia

Ventaja de clases abstractas: Se pueden definir constructores en las clases abstractas

Ventaja de las interfaces: Permite la implementación de n interfaces

Desventaja de interfaces: No se pueden definir constructores en las interfaces

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

El comando en Git usado para fusionar una rama, es “git merge nombreRama”.

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

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

El comando utilizado en Git para traer los cambios del repo remoto al local es “git pull”.

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

41. ¿Qué es un microservicio?

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

42. Dado el siguiente código:

```
public class Main {  
    public static void main(String[] args) {  
        int[] numeros = {1,2,3,4,5};  
        int suma = 0;  
        for (int i = 1; i <= numeros.length; i++){  
            suma += numeros[i];  
        }  
  
        System.out.println("La suma de los números es: " + suma);  
    }  
}
```

¿Este código compila sin errores?

- a) Si, compila sin errores.
- b) No, hay un error en el ciclo for.
- c) No, hay un error en la inicialización de la variable “suma”.
- d) No, hay un error en la declaración del arreglo.

43. ¿Qué método se utiliza para obtener el mensaje de una excepción en Java?

- a) getClass()
- b) printStackTrace()
- c) toString()
- d) getMessage()

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

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

45. ¿Cuál es el resultado de ejecutar el siguiente código?

```
String s = "hello";  
s.toUpperCase();  
System.out.println(s);
```

Convierte a mayúsculas la cadena completa, sin embargo, este no modifica la cadena original, por lo que al imprimir, se imprime la cadena original.

46. ¿Qué son las pruebas de integración?

Las pruebas de integración son aquellas que se prueba el sistema completo, cada uno de sus módulos y se valida el paso de información, si la integración de cada uno de sus módulos es correcta.

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

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

El comando es "git branch nombreRama"

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

48. ¿Cuál es el paquete de importación necesario para usar la clase ArrayList?

- a) import.java.net.\*;
- b) import.java.awt.\*;
- c) import.java.io.\*;
- d) import.java.util.\*;

49. ¿Cuál es el formato de los datos que se envían y reciben en una API REST?

- a) YAML
- b) XML
- c) JSON
- d) Todos los anteriores

50.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.

51. ¿Qué palabra clave se utiliza para definir una excepción personalizada en Java?

- a) try
- b) throw
- c) finally
- d) catch

52. ¿Qué es un operador de short circuit?

El operador de short circuit es un operador que limita la validación de una expresión lógica de ambos lados, esto permite disminuir la carga al sólo evaluar un lado en una expresión lógica.

Los tipos de operadores son dos and y or.

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

53. ¿Cuál de los siguientes comandos elimina el directorio target antes de iniciar el proceso de construcción?

- a) mvn site
- b) mvn build
- c) mvn answer
- d) mvn clean

54. ¿Cuál es el comando utilizado para ver el historial de cambios en Git?

El comando para ver el historial es “git log”.



55. ¿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.