

# GUIA ASO

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

**git reset HEAD-1**

Este comando deshace el ultimo commit realizado en la rama actual, git reset se utiliza para mover la punta de la rama actual a un commit específico y el HEAD-1 especifica el commit anterior al HEAD actual y se utiliza para correcciones de error, reorganización de historial de commits

**git revert** se utiliza para deshacer los efectos de un commit específico creando un nuevo commit que revierte los cambios introducidos por ese commit, este crea un nuevo commit que deshace los cambios del commit específico, manteniendo intacto el historial de commits.

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

Una clase abstracta puede tener métodos abstractos sin implementación y métodos con implementación concretos, esta puede contener campos, constructores y métodos no abstractos, una clase puede extender solo una clase abstracta y actúa como base por medio de la palabra "extends". Este tipo de clases pueden ser invocadas por otras clases. Sus atributos pueden ser de cualquier nivel (public, protected, default o private).

Una interfaz sus métodos son abstractos por defecto, lo que significa que no tienen implementación y una clase puede implementar múltiples interfaces utilizando la palabra "implements" lo que significa que significa herencia múltiple. Las interfaces no tienen constructores porque no se pueden instanciar directamente y son contratos que las clases pueden implementar. Sus atributos son automáticamente públicos y abstractos o al menos que sean definidos como estáticos o por defecto.

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)

La **iteración** es el proceso de repetir un conjunto de instrucciones un cierto número de veces o hasta que se cumpla una condición específica. En el caso de contar la cantidad de monedas de 5 centavos en una matriz de cadenas, necesitas iterar sobre cada elemento de la matriz para verificar si es igual a "5 centavos" y contar cuántas veces ocurre esta **condición**. Un bucle for o un bucle foreach son formas comunes de realizar esta iteración en Java.

4. ¿Qué es un archivo JAR en java?

Es un archivo que se utiliza para empaquetar y distribuir uno o más archivos Java, como clases, recursos, metadatos y archivos auxiliares en una única unidad. Estos son similares a los archivos Zip pero están diseñados específicamente para contener archivos y recursos relacionados con Java.

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

Es un concepto utilizado que permite definir múltiples métodos en una clase con el mismo nombre, pero con diferentes listas de parámetros es decir que estos pueden

ser diferentes entre ellos, lo que permite una funcionalidad similar, pero con diferentes invocaciones y adaptaciones a diferentes tipos de datos o situaciones.

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

ArrayList es un arreglo dinámico continuo de memoria que almacena elementos, y este se incrementa su tamaño automáticamente cuando alcanza su capacidad máxima, y su acceso a los elementos es más rápido, sus inserciones y eliminaciones son más lentas.

LinkedList es una lista doblemente enlazada para elementos, y estos se almacenan en un nodo que contiene una referencia al siguiente nodo y al nodo anterior, su acceso a sus elementos es más lento porque no hay un acceso aleatorio directo a sus elementos, se debe seguir un orden desde el nodo inicial o final a los nodos sucesivos para llegar al destino y sus inserciones y eliminaciones son más rápidas.

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

Se utiliza cuando necesitas asegurar de que ciertas acciones se realicen, independientemente si ocurre una excepción o no dentro del bloque "try" para limpieza de estado donde las operaciones de limpieza o restauración del programa antes del salir del bloque de try o para manejo de excepciones en donde este se ejecutara incluso si no estar dentro del try o si no hay un catch.

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

Se utilizan para validar la funcionalidad, detectar errores, facilitar la refactorización, documentación dinámica, promover la calidad del código y facilitar la colaboración. Los test unitarios son una parte fundamental de las prácticas de desarrollo de software modernas y juegan un papel crucial en la creación de sistemas robustos, fiables y mantenibles.

**9. 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("cad1 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");  
    }  
}
```

De acuerdo con como declaramos las variables de referencia y los objetos, podemos observar que cad1 y cad 3 son iguales, pero cad1 y cad2 son diferentes al declararlo como un new Strin("Hola") donde se esta creando un nuevo objeto y no hacemos que la variable de referencia apunte al mismo objeto que los otros string.

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

```
class Mammal{
    public Mammal(int age){
        System.out.println("Mammal");
    }
}
public class Platypus extends Mammal{
    public Platypus(){
        System.out.println("Platypus");
    }

    public static void main(String[] args) {
        new Mammal(5);
    }
}
```

En el IDE imprime en pantalla Mammal

11. ¿Cómo se manejan las excepciones en java?

Se utilizan los bloques **try** para contener el código propenso a excepciones, el bloque **catch** se utiliza para manejar excepciones específicas que pueden ocurrir en el bloque **try**, y el bloque **finally** se utiliza para contener código que se ejecuta siempre, independientemente de si se produce una excepción o no.

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

La anotación @Ignore en JUnit se utiliza para omitir la ejecución de un método de prueba específico o de una clase de prueba completa. Cuando un método de prueba está anotado con @Ignore, JUnit lo tratará como si estuviera deshabilitado y no lo ejecutará durante la ejecución de la suite de pruebas.

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);  
    }  
}
```

Ya que la inicialización del int x esta dentro de los corchetes al entrar a main esta al no estar inicializada tiene un valor de cero por default y cuando se decrementa su valor este cambia a -1 por eso se imprime -1 en pantalla.

14. ¿Qué es un operador de short circuit?

Es un operador en programación que no evalúa todos sus operandos si no es necesario, y detiene la evaluación tanto como el resultado final puede ser determinado y estos pueden && y ||.

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

DAO (Data Access Object) es un patrón de diseño de software que se utiliza para separar la lógica de acceso a datos de la lógica de negocio en una aplicación. Su objetivo es proporcionar una capa de abstracción entre la aplicación y la fuente de datos (como una base de datos, un servicio web, etc.), lo que facilita el cambio y la evolución de la capa de acceso a datos sin afectar a la lógica de negocio.

Se proporcionan implementaciones concretas de la interfaz DAO para interactuar con la fuente de datos real. Estas implementaciones contienen la lógica específica para realizar operaciones CRUD en la fuente de datos. Por ejemplo, para una base de datos relacional, las implementaciones del DAO podrían utilizar JDBC para interactuar con la base de datos.

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

Se refiere a un punto final específico (URL) en el servidor donde los clientes pueden interactuar con el sistema, cada uno de estos representa una operación o acción específica que el servidor puede realizar sobre los recursos que maneja. Tienen una url única utiliza los verbos de HTTP, devuelve datos en formatos como JSON, XML, HTML o dependiendo de la solicitud, y pueden requerir autenticación o autorización antes de ser accedidos o utilizados.

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;
            }
        }
    }
}
```

Realiza un bucle que se haga hasta que este sea de acuerdo con el contador menor a la longitud de la cadena y al poner el condicional compara los valores por cada una de las letras de la palabra para verificar que es un palíndromo o no.

18. ¿Cuál de las siguientes opciones son verdaderas?

- 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.

Java es un lenguaje de programación de propósito general que soporta múltiples paradigmas de programación, incluyendo el paradigma orientado a objetos, el paradigma imperativo (procedimental), y en cierta medida, el paradigma funcional. El código java en Windows puede ejecutarse en Linux siempre y cuando tenga independencia de plataforma, contenga una máquina virtual java y sea compatible con bibliotecas y dependencias

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

Maven es una herramienta de gestión de proyectos y construcción de software utilizada en el desarrollo de aplicaciones Java. Proporciona un conjunto de estándares y convenciones para la gestión de proyectos, la gestión de dependencias, la compilación, el empaquetado y la distribución de software.

Se utiliza en el desarrollo de app para la gestión de dependencias de un proyecto y manejar automáticamente la descarga e inclusión de las dependencias requeridas en el proyecto. Define un ciclo de vida de construcción estándar en fases predefinidas como compile, test, package, install, deploy. Integra repositorios de artefactos centralizados como el Repositorio Central de Maven, que permite búsqueda y descarga de dependencias.

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.

javac es el compilador de java que toma un archivo fuente .java como entrada y lo compila en un archivo bytecode .class . Java no puede tomar nombre del archivo bytecode como parámetro, la ejecución de un programa Java se realiza mediante el comando java. Java no toma el nombre de la clase como parámetro.

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

Es un sistema de control de versiones y sus comandos básicos son: `init, commit, clone, add, status, log, branch, checkout, merge, pull y push.`

**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?

El uso de excepciones en Java es apropiado para manejar cualquier situación excepcional que pueda interrumpir el flujo normal de ejecución del programa y que no pueda ser manejada de manera adecuada en el contexto del código que la originó. Es importante diseñar y manejar las excepciones de manera adecuada para garantizar que el programa sea robusto, seguro y fácil de depurar.

**24.** ¿Qué es un bean en Spring?

Es un objeto gestionado por el contenedor de Spring. Los beans son componentes fundamentales en la arquitectura de Spring, y representan los diferentes elementos de una aplicación que son gestionados por el contenedor de Spring

25. Selecciona la respuesta correcta 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 ");
    }
}
```

De acuerdo con el código este podría imprimir s,c,t de acuerdo con la herencia,

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

Un método toma una superclase como parámetro, puede pasar a cualquier superclase, así como un método de un parámetro de tipo Object puede tomar como referencia cualquier objeto ya que todas las clases en java son subclases de Object. Una referencia a un objeto se puede convertir en una subclase de objetos mediante una conversión explícita y es conocido como casting.

No todas las excepciones de conversión se pueden detectar en tiempo de ejecución y un método de instancia publica en una superclase no garantiza que el método específico se llame al método en la clase principal en tiempo de ejecución ya que estos se resuelven en tiempo de compilación del tipo estático de la referencia

27. ¿Son patrones de diseño de software estructural?

- **Adapter (Adaptador):** Permite que interfaces incompatibles trabajen juntas.
- **Bridge (Puente):** Desacopla una abstracción de su implementación para que puedan variar independientemente.
- **Composite (Composite):** Compone objetos en estructuras de árbol para representar jerarquías de parte-todo.
- **Decorator (Decorador):** Añade funcionalidad a los objetos dinámicamente.
- **Facade (Fachada):** Proporciona una interfaz simplificada para un conjunto más grande de clases.
- **Proxy (Proxy):** Proporciona un sustituto o marcador de posición para controlar el acceso a un objeto.

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);
    }
}
```

Se tiene una lista la cual contiene valores y se realiza el cálculo de promedio de esta lista y se imprime 3.0

29. ¿Qué son las pruebas de integración?

Son una fase de prueba en el desarrollo de software donde se prueban conjuntos de módulos o componentes de software ya integrados para asegurar que funcionen



correctamente juntos como un sistema completo. En lugar de probar los componentes individualmente (como se hace en las pruebas unitarias), las pruebas de integración se centran en probar cómo interactúan los diferentes módulos cuando se combinan.

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

**Git Push** envía los cambios confirmados en la rama local al repositorio remoto correspondiente. Por defecto, git push enviará los cambios de la rama local al mismo nombre de rama en el repositorio remoto.

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);
    }
}
```

Al no ponerle un valor en el constructor de la classX o de la class2 este inicializara el contructos que incrementa el valor de y y sera  $y=3$  y despues utilizara el constructor donde se ingresa el valor de y donde multiplica  $y * 2$  y quedara como  $y=6$  y entonces realizara la operación declarada en el constructor de la class2 lo que finalizara con el valor de  $y=9$ .

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

El comando es `git branch <branch name>`. Crea una nueva rama con el nombre especifico.

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;  
    }  
}
```

Ya que este no imprime nada en pantalla y las asignaciones y métodos son correctos este compilara correctamente.

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 |
```

Marca error en la línea r4 ya que no se realiza el override en las clases .

35. ¿Qué es Git?

Git es un sistema de control de versiones distribuido, diseñado para rastrear cambios en archivos y coordinar el trabajo en proyectos de desarrollo de software.

36. ¿Cuáles son las excepciones para lanza la JVM?

IOException, FileNotFoundException, ParseException, SQLException,  
ClassNotFoundException, NullPointerException,  
ArrayIndexOutOfBoundsException, ArithmeticException,  
IllegalArgumentException, ClassCastException.

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

El comando utilizado para fusionar una rama es **git merge**. Este comando permite combinar los cambios de una rama específica en la rama actual en la que te encuentras.

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

REST (Representational State Transfer) es un estilo de arquitectura para diseñar sistemas distribuidos, particularmente servicios web. Una API web (Application Programming Interface) es un conjunto de reglas y mecanismos que permite que diferentes programas se comuniquen entre sí a través de la web. Cuando se diseña una API web siguiendo los principios de REST, se le denomina API RESTful.

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

En Git, el comando utilizado para actualizar la rama local con los cambios de la rama remota es **git pull**. Este comando realiza dos acciones: primero, recupera (fetch) los cambios desde el repositorio remoto y luego los fusiona (merge) con la rama local actual.

**40. ¿Qué es un microservicio?**

Es una arquitectura de software que estructura una aplicación como una colección de servicios pequeños, independientes y autónomos. Cada microservicio es responsable de una funcionalidad específica y puede ser desarrollado, desplegado y escalado de manera independiente. Esta arquitectura es una evolución de la arquitectura monolítica, donde todos los componentes de una aplicación están integrados en un solo código base.

**41. 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);  
    }  
}
```

Compila sin errores e imprimirá la suma del arreglo de numero.

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

Es el método **getMessage()** el cual pertenece a la clase Throwable que es la clase base para todas las excepciones y errores en java.

**43. ¿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

En Java las excepciones no comprobadas son subclases de RuntimeException y no necesitan ser declaradas en el método con throws. Las excepciones en tiempo de ejecución son lo mismo que las excepciones no comprobadas. Las excepciones en tiempo de ejecución no son lo mismo que las excepciones comprobadas, y estas

deben ser declaradas en la firma del método y manejadas explícitamente. Pueden declararse excepciones comprobadas como no comprobadas con el método usando throws. Se pueden usar subclasses de Throwable, que incluye Error no es recomendable porque representa errores serios que una aplicación típica no debería intentar manejar.

**44.** ¿Cuál es el resultado de ejecutar el siguiente código?

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

Imprimira en pantalla hello ya que aunque convertimos las cadena en mayusculas no hacemos la asignacion para que apunte a ese objeto donde hello esta en mayusculas.

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

Para usar la clase ArrayList en Java debemos de importar el paquete `java.util`.

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

Usan típicamente `JSON` pero también puede usar formatos como `XML`, `YAML` o incluso `plain text` dependiendo de la especificación de la API y las necesidades del cliente y servidor.

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

Es un operador de referencia de método, y de una forma concisa de referencia de métodos y constructores sin tener que invocarlos explícitamente.

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

Para definir una excepción personalizada en Java, se utiliza la palabra clave `class`. Una vez definida la excepción personalizada, puedes lanzarla utilizando la palabra clave `throw` y manejarla con un bloque try-catch.

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

En el uso en un archivo de construcción de Maven se utiliza el comando `mvn clean` para eliminar el directorio target

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

Se utiliza el comando `git log`, que muestra una lista de commits realizados en el repositorio junto con los detalles como el hash del commit, autor, fecha y mensaje del commit.

**51.** ¿Qué es una expresión lambda en Java 8?

Una expresión lambda en Java 8 es una función anónima que se puede utilizar para proporcionar una implementación concisa de una interfaz funcional, es decir, una interfaz con un único método abstracto. Las expresiones lambda permiten tratar la

funcionalidad como un argumento del método, o sea, pueden pasar bloques de código como si fueran datos.

52. ¿Qué muestra el siguiente código fuente por pantalla?

```
int x = 1;
switch (x){
    case 1:
        System.out.println("Uno");
    case 2:
        System.out.println("Dos");
    case 3:
        System.out.println("Tres");
    default:
        System.out.println("Otro número");
}
```

Imprimira todos los mensajes de los case , ya que ninguno contiene un break en cada uno de los case.

53. De los siguientes paquetes, ¿cuáles contienen clases para construir una interfaz gráfica? (Elije todas las que correspondan)

Las principales clases y paquetes que se utilizan para construir interfaces gráficas en Java se encuentran en el paquete `java.awt` y en el paquete `javax.swing`.

54. ¿Cuál de las siguientes líneas deben ir en el espacio en blanco para que el código compile?  
public class News < \_\_\_\_ > { }

Indica un parámetro de tipo genérico, donde puede definir clases genéricas que toman uno o más parámetros de tipo. Puede ser `T` o como identificar la variable, Se puede utilizar `News` pero estamos limitando la clase News a que trabaje solo son instancias News o subclases de este y para utilizar `Object` estamos permitiendo que la clase News trabaje como cualquier tipo de objeto, ya que es de Object es de la clase base de todas las clases en Java.

55. ¿Qué es un stream en Java 8 y para qué se utiliza?

Stream es una secuencia de elementos que permite realizar operaciones de forma secuencial o paralela, proporcionando una forma más declarativa y funcional de trabajar con colecciones de datos. Realiza operaciones de transformación, filtrado, mapeo, ordenamiento entre otras de una manera eficiente y concisa. Trabaja con volúmenes de datos de manera eficiente, facilita la programación paralela al permitir el procesamiento en paralelo en sistemas multinúcleo.