­

**Preguntas de Examen**

Luis Alberto García Reyes

17/05/2024

# Preguntas

## 1.- ¿Cuál es el comando utilizado paradeshacer el último commit en git?

R= Al comando git revert se le puede considerar un comando para "deshacer", pero lo cierto es que no lo es en el sentido tradicional. En lugar de eliminar la confirmación del historial del proyecto, resuelve cómo invertir los cambios introducidos por la confirmación y añade una nueva con el contenido inverso resultante. Así, se evita que Git pierda el historial, lo cual resulta importante para la integridad del historial de revisiones y para que la colaboración sea fiable

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

R= Una interfaz es una colección de métodos abstractos, constantes y firmas de métodos que define un contrato para las clases que la implementen, mientras que una clase abstracta puede contener métodos concretos e implementados, así como métodos abstractos y variables de instancia. En Java 8, la principal diferencia entre una interfaz y una clase abstracta es:

Una interfaz puede tener métodos default (con implementación) y métodos static, mientras que una clase abstracta no puede tener métodos static a nivel de clase (sólo métodos de instancia).

## 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)

R=

Conditional (Condicional):

* Declaraciones condicionales (if, switch) se utilizan para tomar decisiones en el código. En este caso, se utilizarían para verificar si una moneda es de 5 centavos.

Assertion (Aserción):

* Las aserciones (assert) se utilizan principalmente para pruebas y comprobaciones durante el desarrollo, para asegurar que ciertas condiciones se cumplan. No son adecuadas para la lógica principal del programa en este contexto.

Assignment (Asignación):

* Las declaraciones de asignación (=) se utilizan para asignar valores a las variables. Aquí se utilizarían para incrementar un contador cuando se encuentra una moneda de 5 centavos.

Iteration (Iteración):

* Las declaraciones de iteración (for, while) se utilizan para recorrer la matriz y acceder a cada moneda. Son esenciales para verificar todas las monedas en la matriz.

## 4.- ¿Qué es un archivo JAR en java?

R= La extensión .jar es propia del lenguaje de programación Java y puede tener dos usos principales. Primero, un archivo .jar puede ser una aplicación de Java, es decir, un programa ejecutable. Segundo, más comúnmente, un archivo .jar actúa como una biblioteca que contiene múltiples archivos Java (archivos .class) y metadatos, comprimidos de manera similar a un archivo .zip. Estos archivos pueden incluir imágenes, archivos de audio y otros formatos. Las bibliotecas .jar simplifican la programación de aplicaciones y applets de Java al permitir la integración de diversos recursos como metadatos, archivos .xml o .json, gráficos y audio. Un componente clave de estas bibliotecas es el manifiesto, que incluye información sobre la versión y el autor del código.

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

R= La sobrecarga de métodos en Java es una característica que permite a una clase tener más de un método con el mismo nombre, pero con diferentes parámetros. Esta técnica se utiliza para aumentar la legibilidad y la reutilización del código, permitiendo que un solo método maneje diferentes tipos o cantidades de datos.

Las condiciones para la sobrecarga de métodos son:

* Diferentes tipos de parámetros: Los métodos deben diferir en el tipo de los parámetros.
* Diferente número de parámetros: Los métodos deben diferir en la cantidad de parámetros.
* Diferente orden de parámetros: Si los métodos tienen el mismo número de parámetros, el orden de estos debe ser diferente.

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

|  |  |  |
| --- | --- | --- |
|  | ArrayList | LinkedList |
| Estructura Interna | * Utiliza un array dinámico para almacenar los elementos. * El tamaño del array puede cambiar dinámicamente a medida que se añaden o eliminan elementos. * Ideal para acceso rápido a elementos por índice. | * Utiliza una estructura de lista enlazada doblemente. * Cada elemento (nodo) contiene una referencia al anterior y al siguiente nodo. * Ideal para inserciones y eliminaciones frecuentes. |
| Complejidad de Operaciones | * Acceso por índice: O(1) (tiempo constante). * Inserción/Eliminación: O(n) (lineal) en el peor caso, especialmente cuando se inserta o elimina en el medio del array. * Añadir al final: O(1) amortizado (puede ser O(n) si se requiere redimensionar el array). | * Acceso por índice: O(n) (lineal) ya que se debe recorrer la lista para acceder a un elemento específico. * Inserción/Eliminación: O(1) (tiempo constante) para operaciones en los extremos de la lista (inicio o fin) y O(n) para acceso en posiciones intermedias. |
| Uso de Memoria | * Menor sobrecarga de memoria ya que solo almacena los datos sin referencias adicionales. * Puede desperdiciar memoria si la capacidad del array es mayor que el número de elementos almacenados. | * Mayor sobrecarga de memoria debido a las referencias adicionales (pointers) en cada nodo. * Más eficiente en el uso de memoria si se realizan muchas inserciones y eliminaciones. |
| Aplicaciones y Casos de Uso | * Adecuado para aplicaciones donde el acceso rápido a los elementos por índice es crucial. * Útil cuando las inserciones y eliminaciones son menos frecuentes comparadas con las lecturas. | * Adecuado para aplicaciones donde se realizan muchas inserciones y eliminaciones, especialmente en los extremos de la lista. * Útil cuando la secuencia de acceso a los elementos no es tan importante. |

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

R= En Java, el bloque finally se utiliza junto con un bloque try y opcionalmente un bloque catch para asegurar que un conjunto específico de instrucciones se ejecute independientemente de si se lanza o no una excepción en el bloque try. El bloque finally es útil para liberar recursos que se deben liberar pase lo que pase, como cerrar archivos, liberar conexiones de base de datos, o liberar memoria.

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

R= Una prueba unitaria es un bloque de código que verifica la precisión de un bloque más pequeño y aislado de código de aplicación, normalmente una función o un método. La prueba unitaria está diseñada para verificar que el bloque de código se ejecuta según lo esperado (Según lo esperado se refiere al estándar de calidad del software), de acuerdo con la lógica teórica del desarrollador. La prueba unitaria solo interactúa con el bloque de código a través de entradas y salidas (verdaderas o falsas) capturadas afirmadas.

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

R=

Texto

Descripción generada automáticamentecad1 diferente a cad2

cad1 es igual a cad3

Analisis

**cad1 == cad2:**

* cad1 es un literal de cadena.
* cad2 es un nuevo objeto String.
* No son el mismo objeto.
* Resultado: cad1 diferente a cad2

**cad1 == cad3:**

* cad1 y cad3 son el mismo literal de cadena.
* Son el mismo objeto.
* Resultado: cad1 es igual a cad3

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

R=Suponiendo que el código esta numerado línea por línea como aparece en la imagen, entonces el código no va a compilar porque el constructor que hereda Platypus es de Mammal entonces debe de ser declarado como lo dicta el constructor padre, con un argumento tipo int dado que Platypus tiene su constructor propio est debería de llamar al padre pero no lo hace por lo tanto no compila en esa línea linea 8 o 7 dependiendo como se vea pero es public Platypus() lo que esta mal

Texto

Descripción generada automáticamente

## 11.- ¿Cómo se manejan las excepciones en java?

R= El manejo de excepciones en Java es fundamental. Java maneja las excepciones con los bloques try- catch y ocasionalmente finally.

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

R= La anotación de prueba @Ignore se utiliza para ignorar pruebas particulares o grupos de pruebas para evitar el error de compilación.

@Ignorar La anotación se puede utilizar en dos escenarios como se indica a continuación:

* Si desea ignorar un método de prueba, utilice @Ignore junto con la anotación @Test.
* Si desea ignorar todas las pruebas de clase, utilice la anotación @Ignore a nivel de clase.

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

Texto

Descripción generada automáticamenteR=-1

La clave aquí es entender que el bloque estático inicial no afecta la variable estática x declarada a nivel de clase, ya que int x = 3; dentro del bloque estático es una variable local al bloque, no tiene ningún impacto en la variable x estática de la clase.

## 14.- ¿Qué es un operador de short circuit?

R= En Java, los operadores de cortocircuito (short-circuit operators) son aquellos que, al evaluar una expresión lógica, pueden determinar el resultado final sin necesidad de evaluar todos los operandos. Esto se debe a que, una vez que el resultado de la expresión se puede determinar, la evaluación se detiene. Los operadores de cortocircuito en Java son && (AND lógico) y || (OR lógico).

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

R= El patrón de diseño DAO (Data Access Object) es un patrón estructural que proporciona una abstracción para las operaciones de acceso a datos, aislando la lógica de negocio de la lógica de persistencia. Esto facilita el mantenimiento y la escalabilidad del código al separar las responsabilidades. El DAO se encarga de las operaciones CRUD (Create, Read, Update, Delete) sobre la base de datos o cualquier otro almacenamiento de datos. Se puede implementar en Java utilizando interfaces y clases concretas.

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

R= Un endpoint en una API REST (Representational State Transfer) es una URL específica que permite acceder a un recurso determinado y realizar operaciones sobre ese recurso. Los endpoints son puntos de acceso a la API y definen cómo se puede interactuar con los datos del servidor.

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

Texto

Descripción generada automáticamenteR=Este programa es una implementación de un algoritmo para verificar si una cadena de texto es un palíndromo o no.

Un palíndromo es una palabra, frase, número o secuencia de caracteres que se lee igual hacia adelante que hacia atrás. Por ejemplo, "anita lava la tina" es un palíndromo.

## 18.- ¿Cuál de las siguientes opciones son verdaderas? (elija todas las correctas)

a) Java es un lenguaje orientado a objetos.

* Verdadero: Java es principalmente un lenguaje de programación orientado a objetos, ya que se basa en la definición y manipulación de objetos y clases.

b) El código Java compilado en Windows puede ejecutarse en Linux.

* Verdadero: El código Java se compila en bytecode que se ejecuta en la Java Virtual Machine (JVM). La JVM es multiplataforma, por lo que el bytecode compilado en una plataforma (como Windows) puede ejecutarse en otra plataforma (como Linux) siempre que haya una JVM disponible.

c) Java permite la sobrecarga de operadores.

* Falso: Java no permite la sobrecarga de operadores. A diferencia de C++, Java no permite que los operadores sean redefinidos para usarse con diferentes tipos de datos.

d) Java es un lenguaje de programación funcional.

* Falso: Java no es un lenguaje de programación funcional en su esencia. Sin embargo, con la introducción de lambdas y el soporte para algunos conceptos funcionales a partir de Java 8, Java ha incorporado características de programación funcional, pero sigue siendo principalmente un lenguaje orientado a objetos.

e) Java es un lenguaje procedimental.

* Falso: Java no es un lenguaje procedimental. Aunque Java permite la escritura de métodos (procedimientos) y su invocación, su enfoque principal es la programación orientada a objetos.

f) Java tiene punteros a ubicaciones específicas en la memoria.

* Falso: Java no tiene punteros explícitos como C o C++. La gestión de memoria y el acceso directo a ubicaciones específicas en la memoria están manejados por la JVM y el recolector de basura (garbage collector), lo que hace que el uso de punteros sea innecesario y, por diseño, se evita para garantizar la seguridad y la simplicidad del manejo de memoria.

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

R=Maven es una herramienta esencial en el ecosistema de desarrollo de Java que proporciona una forma estandarizada y eficiente de gestionar proyectos, manejar dependencias y automatizar tareas de construcción y distribución de software.

## 20.- ¿Cuál de lo siguiente es cierto? (elija todas las correctas)

a): Incorrecta. javac compila un archivo .java en un archivo .class, no en un archivo .bytecode.

b): Incorrecta. Java no toma el nombre del archivo .bytecode como parámetro. javac toma el nombre del archivo .java como parámetro.

c): Correcta. javac es el compilador de Java y se utiliza para compilar archivos fuente de Java (.java) en archivos de clase (.class), que contienen bytecode Java ejecutable por la JVM (Java Virtual Machine).

d): Correcta. Para compilar un archivo Java (.java) con javac, se utiliza el nombre del archivo .java como parámetro. Además, dentro del archivo .java, el nombre de la clase principal debe coincidir con el nombre del archivo .java.

e): Incorrecta. Java no toma el nombre del archivo .class como parámetro. Además, para ejecutar un archivo .class, se utiliza el comando java, no javac.

f): Incorrecta. javac compila archivos fuente (.java) en archivos de clase (.class), no al revés

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

R=Git es un sistema de control de versiones distribuido que se utiliza para el seguimiento de cambios en el código fuente durante el desarrollo de software.

Comandos básicos de git: git init, git clone, git add, git commit, git status, git log, git Branch, git checkout, git merge, git pull, git push

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

R=

1. int variableA = 10;  
   float variableB = 10.5f;  
   int variableC = variableA + variableB;

En Java, cuando se realiza una operación aritmética entre un tipo int y un tipo float, el resultado es un tipo float. Por lo tanto, en este caso, variableC se está declarando como un int, lo cual generará un error de compilación porque no se puede asignar un tipo float a un tipo int sin una conversión explícita.

1. byte variableA = 10;  
   double variableB = 10.5f;  
   double variableC = variableA + variableB;
2. byte variableA = 10;  
   float variableB = 10.5f;  
   float variableC = variableA + variableB;

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

R=

a) La computadora se incendió: Este escenario no es controlable desde el código y está más allá del ámbito de las excepciones de Java.

b) No sabe cómo codificar un método: Esto no es una excepción en el sentido de Java. En su lugar, podría requerir una investigación adicional o un aprendizaje para comprender cómo codificar el método correctamente.

c) No se encuentra un elemento al buscar en una lista: Esto es una situación común y esperada en la programación. Normalmente se maneja verificando si el elemento está presente antes de intentar acceder a él.

d) Se pasa un parámetro inesperado a un método.

Cuando se pasa un parámetro inesperado a un método, esto puede resultar en un comportamiento impredecible o incorrecto del método. En este caso, lanzar una excepción (por ejemplo, IllegalArgumentException) es una forma adecuada de manejar la situación y notificar al llamador del método de que ha ocurrido un error debido a un parámetro inesperado.

e) Quiere recorrer una lista: Recorrer una lista es una operación estándar en la programación y no representa una situación excepcional que requiera el uso de excepciones.

## 24.- ¿Qué es un bean en Spring?

R= En el contexto de Spring Framework, un bean es un objeto que es administrado por el contenedor de Spring. El término "bean" se refiere a un componente en la aplicación que es instanciado, configurado y ensamblado por el contenedor de Spring, en base a la configuración proporcionada por el desarrollador.

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

R= Ninguna opción es correcta dado que el programa devuelve:

s

c

t

En todo caso la opción mas cercana es la opción a) pero no es una garantía.

Texto

Descripción generada automáticamente

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

R=

a) Si un método toma una superclase de 3 objetos, cualquiera de esas clases puede pasarse como parámetro del método.

* Verdadero. En Java, si un método tiene un parámetro que es una superclase, puedes pasarle cualquier objeto que sea una instancia de esa superclase o de cualquiera de sus subclases. Esto se debe a la compatibilidad de tipos y el polimorfismo en la orientación a objetos.

b) Un método que toma un parámetro con tipo java.lang.Object tomará cualquier referencia.

* Verdadero. java.lang.Object es la clase base de todas las clases en Java, por lo que un método que acepta un parámetro de tipo Object puede recibir cualquier objeto, sin importar su tipo.

c) Una referencia a un objeto se puede convertir a una subclase de objetos en una conversión explícita.

* Verdadero, pero con precaución. En Java, puedes realizar una conversión explícita (cast) de una referencia a un objeto a una subclase. Sin embargo, esto solo funcionará si la referencia en realidad es una instancia de la subclase a la que se está convirtiendo. De lo contrario, se lanzará una ClassCastException en tiempo de ejecución.

d) Todas las excepciones de conversión se pueden detectar en tiempo de compilación.

* Falso. Las excepciones de conversión, como ClassCastException, ocurren en tiempo de ejecución y no pueden ser detectadas en tiempo de compilación. El compilador no puede garantizar que una conversión será segura si depende del tipo real del objeto en tiempo de ejecución.

e) 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.

* Falso. En Java, si una subclase sobrescribe un método de la superclase, el método sobrescrito en la subclase será llamado en tiempo de ejecución, no el de la superclase. Esto se debe al mecanismo de vinculación dinámica o dispatch dinámico.

## 27.- ¿Son patrones de diseño de software estructural?

R=Son principales patrones de diseño estructural: Adapter (Adaptador), Bridge (Puente), Composite (Compuesto), Decorator (Decorador), Facade (Fachada), Flyweight (Peso Ligero), Proxy (Proxy).

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

R=Este código calcula el promedio de los números enteros 1, 2, 3, 4, 5 utilizando la programación funcional y las operaciones de flujo de datos en Java 8. El resultado es 3.0

Texto

Descripción generada automáticamente

## 29.- ¿Qué son las pruebas de integración?

R=Las pruebas de integración son un tipo de pruebas de software que se centran en verificar la interacción y el funcionamiento conjunto de varios componentes o módulos del sistema. A diferencia de las pruebas unitarias, que prueban componentes individuales de forma aislada, las pruebas de integración aseguran que los componentes funcionan correctamente cuando se combinan.

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

R= git push Este comando toma todos los cambios confirmados en tu repositorio local y los envía al repositorio remoto que has configurado. Si deseas especificar una rama particular, puedes añadir el nombre de la rama después del comando push.

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

R= El código manipula el valor de la variable estática y a través de varias llamadas a constructores anidados. Primero se inicializa en 2, luego se incrementa a 3, se multiplica por 2 para obtener 6, y finalmente se incrementa en 3 más para obtener el valor final de 9, que se imprime en la consola.

Texto

Descripción generada automáticamente

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

R=El comand para crear una nueva rama en git es git branch nombre-de-la-rama este comando crea una nueva rama llamada nombre-de-la-rama en tu repositorio local, pero no cambia a esa rama automáticamente.

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

R= el código compila satisfactoriamente como no tiene método main no muestra nada.

Texto

Descripción generada automáticamente

## 34.- ¿Cuál es la primer línea en fallar al compilar?

R=el método use no puede ser private en la declaración r4

Texto

Descripción generada automáticamente

## 35.- ¿Qué es Git?

R=Git es un sistema de control de versiones distribuido que permite a múltiples desarrolladores trabajar en un proyecto de manera concurrente sin interferir en el trabajo de los demás. Sus funcionalidades permiten a los desarrolladores trabajar en colaboración, mantener un historial detallado de todos los cambios realizados en el código, y gestionar diferentes versiones del software de manera eficiente.

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

R= La Máquina Virtual de Java (JVM) lanza varias excepciones que son subclases de java.lang.RuntimeException y java.lang.Error. Estas excepciones pueden ser categorizadas en dos grupos principales: excepciones y errores.

**Excepciones (RuntimeException)**

* NullPointerException:
* ArrayIndexOutOfBoundsException
* ClassCastException:
* IllegalArgumentException:
* IllegalStateException:
* ArithmeticException:
* NumberFormatException
* UnsupportedOperationException:

**Errores (Error)**

* StackOverflowError:
* OutOfMemoryError:
* VirtualMachineError:
  + OutOfMemoryError
  + StackOverflowError
  + InternalError
  + UnknownError
* LinkageError:
  + NoClassDefFoundError
  + ClassFormatError
  + UnsupportedClassVersionError
  + ExceptionInInitializerError

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

R= El comando utilizado para fusionar una rama en Git es:

git merge nombre-de-la-rama

Este comando combina los cambios de la rama especificada (nombre-de-la-rama) en la rama actual en la que te encuentras.

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

R= REST es un estilo arquitectónico que se utiliza para construir servicios web escalables y mantenibles. Las API RESTful siguen los principios de REST y permiten que los sistemas se comuniquen de manera eficiente utilizando los métodos estándar de HTTP y una arquitectura basada en recursos identificados por URLs. REST ha sido ampliamente adoptado debido a su simplicidad y flexibilidad, convirtiéndolo en una elección popular para desarrollar APIs web.

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

R=El comando git pull es la manera más sencilla y común de actualizar una rama local con los cambios de su contraparte remota en Git. Combina los pasos de git fetch y git merge en un solo comando, simplificando el flujo de trabajo para mantener tu rama local sincronizada con el repositorio remoto.

## 40.- ¿Qué es un microservicio?

R= Un microservicio es un componente autónomo que se puede desplegar de forma independiente. Está diseñado para realizar una función específica del negocio y expone su funcionalidad a través de endpoints, generalmente mediante APIs. Los microservicios se comunican entre sí de manera desacoplada, permitiendo una gran flexibilidad y escalabilidad en el desarrollo y despliegue de aplicaciones.

## 41.- Dado el siguiente código:

Texto

Descripción generada automáticamente

¿Este código compila sin errores?

R=El código compla sin errores, las declaraciones están escritas correctamente.

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

R=En Java, el método que se utiliza para obtener el mensaje de una excepción es getMessage()

## 43.- ¿Cuál de las siguientes afirmaciones son verdaderas? (Elije todas las correctas)

R=

a) Puede declarar solo excepciones no comprobadas.

* Falso. Puedes declarar tanto excepciones comprobadas (checked) como no comprobadas (unchecked).

b) Las excepciones en tiempo de ejecución son lo mismo que las excepciones no comprobadas.

* Verdadero. Las excepciones en tiempo de ejecución (runtime exceptions) son un tipo de excepciones no comprobadas (unchecked).

c) Las excepciones en tiempo de ejecución son lo mismo que las excepciones comprobadas.

* Falso. Las excepciones en tiempo de ejecución son no comprobadas (unchecked), no comprobadas (checked).

d) Solo puede declarar excepciones comprobadas.

* Falso. Puedes declarar tanto excepciones comprobadas como no comprobadas.

e) Solo puede manejar subclases de Exception.

* Verdadero. En Java, puedes manejar cualquier subclase de Exception, incluyendo RuntimeException (excepciones no comprobadas) y las comprobadas.

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

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

R= El código se compilará y ejecutará sin errores, y la salida en la consola será: Hello

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

R= Para utilizar la clase ArrayList, debes importar el paquete java.util con la siguiente línea de código:

import java.util.ArrayList;

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

R= El formato más común y estándar para los datos en una API REST es JSON debido a su simplicidad y compatibilidad con la mayoría de los lenguajes de programación y herramientas. Otros formatos como XML, Form Data, Plain Text, y Protocol Buffers también pueden ser utilizados dependiendo de los requisitos específicos de la aplicación. YAML es un formato válido para enviar y recibir datos en una API REST. Aunque no es tan común como JSON o XML, puede ser utilizado cuando la legibilidad y la simplicidad son importante

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

R=El operador de doble dos puntos (::) en Java 8 se conoce como el operador de referencia a método (method reference operator). Este operador proporciona una forma concisa y legible de referenciar un método o un constructor existente sin tener que invocarlo explícitamente. Se utiliza en el contexto de expresiones lambda y es especialmente útil cuando la expresión lambda simplemente llama a un método ya existente.

Se usa para: hacer referencia a un Método Estático aunque el operador de doble dos puntos (::) en Java 8 puede hacer referencia a métodos no estáticos (métodos de instancia). Referencia a un Método de Instancia de un Objeto Particular, Referencia a un Método de Instancia de un Objeto Arbitrario de un Tipo Particular y Referencia a un Constructor

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

R=throw se utiliza en Java para lanzar una excepción, no para definir una excepción personalizada. La palabra clave throw se usa dentro de un método para indicar que se está lanzando una instancia de una excepción. Para definir una excepción personalizada, se utiliza la palabra clave class para crear una nueva clase que extienda Exception o RuntimeException.

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

R=El comando mvn clean limpia el proyecto, eliminando los archivos generados durante la compilación previa. Esto incluye la eliminación del directorio target, que es donde Maven coloca todos los archivos compilados, artefactos y otros resultados de la construcción.

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

R=El comando git log muestra una lista de los commits en el repositorio, proporcionando detalles sobre cada commit, como el identificador del commit (SHA-1), el autor, la fecha y el mensaje del commit.

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

R= Una expresión lambda en Java es una función (Ojo nos refermos a funcionalidad, dado que en java no se manejan funciones sino clases) anónima, es decir, una clase sin nombre, que puede ser tratada como un valor. Se introdujeron en Java 8 como una forma de proporcionar una sintaxis más concisa para implementar interfaces funcionales.

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

Texto

Descripción generada automáticamente

R=Uno Dos Tres Otro número

Uno se imprime porque x es igual a 1 y coincide con case 1.

Dos, Tres y Otro número se imprimen porque no hay instrucciones break para detener la ejecución después de case 1.

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

R=

* javax.swing
* java.awt

Tanto javax.swing como java.awt ofrecen una amplia gama de componentes para la creación de interfaces gráficas de usuario en Java. javax.swing es más poderoso y flexible, mientras que java.awt es más básico pero puede ser útil en ciertas situaciones o para aplicaciones más simples. En general, javax.swing se considera la opción preferida para el desarrollo de GUI en Java debido a su riqueza de características y flexibilidad.

## 54.- ¿Cuál de las siguientes líneas deben ir en el espacio en blanco para que el código compile?

public class News < \_\_\_\_\_> { }

R= Dentro de los corchetes angulares < > en la declaración de una clase genérica en Java, se coloca el parámetro genérico, que es un tipo o una lista de tipos. Este parámetro genérico puede ser cualquier nombre de tipo válido en Java, y se utiliza para representar un tipo que será definido más adelante cuando se utilice la clase. Es posible utilizar el propio nombre de la clase como un parámetro genérico dentro de los corchetes angulares en la declaración de una clase genérica en Java, también Object y N.

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

R=En Java, un Stream es una secuencia de elementos que permite realizar operaciones agregadas de forma funcional y fluida sobre datos. Los Streams fueron introducidos en Java 8 como parte del paquete java.util.stream, y proporcionan una manera moderna y eficiente de procesar colecciones de datos (como listas, conjuntos y mapas) y otras fuentes de datos (como arreglos).