**EXAMEN ASO A1**

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

|  |  |
| --- | --- |
| a) git reset | deshace cambios locales moviendo la referencia HEAD a un commit anterior. |
| b) git revert | deshace los cambios introducidos por un commit específico creando un nuevo commit que aplica los cambios inversos. |
| c) git amend | modifica el commit más reciente. |
| d) git checkout | se utiliza principalmente para cambiar entre ramas o restaurar archivos del directorio de trabajo a un estado anterior. |

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

|  |  |
| --- | --- |
| 1. Una clase abstracta puede contener implementaciones de métodos, mientras que una interfaz no puede. | *Características clases Abstractas*   * Una clase abstracta debe definirse como abstracta si cuenta al menos con un método abstracto, (Es aquel que está declarado, pero no implementado) sin embargo puede haber clases abstractas sin ningún método abstracto. * También puede incluir atributos, métodos y constructores estándares. * Una clase que herede una clase abstracta está obligada a sobrescribir los métodos abstractos heredados, el objetivo es forzar a que todas las subclases tengan el mismo formato de método. * "Es un" (herencia simple)   *Características las Interfaces*   * Es un conjunto de métodos abstractos, su objetivo es definir el formato de ciertos métodos. * Pueden incluir constantes, que serán públicas y estáticas. * Herencia múltiple * Puede tener métodos default, pero si una clase implementa dos interfaces con el mismo método default esta obligado a sobrescribirlo, si no, error de compilación. * Las interfaces sí pueden contener implementaciones de métodos a través de métodos por defecto y métodos estáticos. * "Puede hacer" (comportamiento/contrato) |
| 1. Una clase abstracta puede contener variables de instancia, mientras que una interfaz no puede. |
| 1. Una interfaz puede contener implementaciones de métodos, mientras que una clase abstracta no puede. |
| 1. Una interfaz solo puede heredar de una clase, mientras que una clase abstracta puede heredar… (no se ve) |

1. De los siguientes ¿Qué tipos de declaraciones se deben usar para contar la cantidad de monedas en cadenas de varias monedas? (no se ve bien el final de la pregunta)

|  |  |
| --- | --- |
| 1. Conditional | Un condicional se utiliza para ejecutar un bloque de código solo si se cumple una determinada condición (una expresión booleana que evalúa a verdadero o falso). |
| 1. Assertion | Implica una verificación de una suposición del programador en tiempo de desarrollo/prueba |
| 1. Assignment | Una asignación se utiliza para dar un valor a una variable o constante. Almacena un dato en una ubicación de memoria referenciada por el nombre de la variable. |
| 1. Iteration | Una iteración (o bucle) se utiliza para repetir un bloque de código múltiples veces. |

1. ¿Qué es un archivo JAR en java?

|  |  |
| --- | --- |
| 1. Un archivo que contiene un archivo de configuración Maven | La función principal de un JAR es agrupar múltiples archivos (.class compilados, archivos de sonido, imágenes, archivos de propiedades, etc.) |
| 1. Un archivo que contiene un archivo de configuración Git. |
| 1. Un archivo que contiene una clase Java compilada |
| 1. Un archivo que contiene un archivo de configuración de Spring |

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

|  |  |
| --- | --- |
| 1. Cuando un método tiene múltiples definiciones con el mismo nombre y tipo de parámetros. | Una clase puede contener varios métodos con el mismo nombre, pero deben diferenciarse en el número o tipo de parámetros   * El tipo de devolución no afecta |
| 1. Cuando un método tiene múltiples definiciones con diferentes nombres y cantidades de parámetros. |
| 1. Cuando un método tiene múltiples definiciones con diferentes tipos de cantidades de parámetros. |
| 1. Cuando un método tiene múltiples definiciones con diferentes nombres y tipos de parámetros. |

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

|  |  |
| --- | --- |
| 1. ArrayList es más rápido que un LinkedList para agregar y eliminar elementos. | ArrayList   * ArrayList tienen un índice * Es práctico para buscar un elemento por medio de su índice * En arrayList es más complejo eliminar un elemento, porque todos los elementos se recorren sus índices y ocupa mucha memoria * Almacena los elementos en un array contiguo. Cada elemento ocupa solo el espacio de almacenamiento del dato   LinkedList   * LikedList es una lista entrelazada y puede manejar pilas LIFO y colas FIFO. * Una de sus desventajas es que es pesado para buscar un elemento porque tiene que pasar uno por uno de sus elementos * Una de la ventaja de Likedlist es a la hora de eliminar un elemento, ya que solo le elimina el enlace siguiente y enlace anterior. * Consume más memoria porque cada elemento que almacena, además del dato guarda punteros al nodo siguiente y al anterior. |
| 1. ArrayList es más eficiente en el uso de memoria que LinkedList. |
| 1. LinkedList es una clase abstracta mientras que ArrayList es una clase concreta. |
| 1. LinkedList es más rápido que ArrayList para acceder a elementos aleatorios. |

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

|  |
| --- |
| 1. Cuando no hay bloques catch en una declaración try. |
| 1. Nunca. |
| 1. Cuando hay dos o más bloques catch en una sentencia try. |
| 1. Cuando hay exactamente un bloque catch en una sentencia try. |

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

|  |
| --- |
| 1. Comprobar la eficiencia del hardware. |
| 1. Medir la velocidad de la aplicación. |
| 1. Ahorrar tiempo en el desarrollo. |
| 1. Asegurar la calidad del software |

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

Texto

Descripción generada automáticamente

|  |
| --- |
| 1. cad1 diferente a cad2 🡪 *cad2 al ser un objeto su referencia apunta a un objeto* cad1 es igual a cad3 🡪 *cad1 y cad2 son primitivos por lo tanto su valor esta dentro del Pool* |
| 1. ca1 es igual a cad2 ca1 es igual a cad3 |
| 1. No compila |
| 1. cad1 diferente a cad2 cad1 diferente a cad3 |

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

🡪 Hay un error en la línea 7,

Todas las clases incluye de forma implícita en sus constructores como primera línea la instrucción de super(); que es la que llama al constructor por defecto de la superclase.

Mammal al declarar un constructor con parámetros y no crear el constructor default, genera un error de compilación.

Texto, Carta

Descripción generada automáticamente

|  |
| --- |
| 1. Mammal. |
| 1. MammalPlatypus. |
| 1. El código no se compila en la línea 11. |
| 1. (no se ve, pero debería ser que no compila en la línea 7 ya que las otras son incorrectas). |

1. ¿Cómo se manejan las excepciones en java?

|  |
| --- |
| 1. Con la instrucción try-catch. |
| 1. Con la instrucción if-else. |
| 1. Con la instrucción for. |
| 1. Con la instrucción while. |

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

|  |
| --- |
| 1. Verdadero |
| 1. Falso |

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

🡪 Hay un error en la línea 7,

Todas las clases incluye de forma implícita en sus constructores como primera línea la instrucción de super(); que es la que llama al constructor por defecto de la superclase.

Mammal al declarar un constructor con parámetros y no crear el constructor default, genera un error de compilación.

Texto

Descripción generada automáticamente

|  |
| --- |
| 1. Error de compilación en la línea 7, x no se inicializa. |
| 1. -1 |
| 1. -2 |
| 1. 0 |

1. ¿Qué es un operador de short circuit?

|  |
| --- |
| 1. Es un patrón de arquitectura de microservicios que nos permite evitar el consumo de servicios que están… (no se ve, pero es incorrecta) |
| 1. Sirve para realizar más eficientes las operaciones condicionales evitando realizar operaciones si están (tampoco se ve completa, pero esta es la correcta) 🡪 Es un comportamiento de los operadores lógicos && y || |
| 1. Sirve para lanzar una excepción personalizada. |
| 1. Operador que nos sirve para crear una nueva clase anónima. |

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

|  |
| --- |
| 1. El patrón de diseño DAO es un patrón que se utiliza para abstraer la capa de negocios en una aplicación (no se ve) clases abstractas y métodos estáticos. |
| 1. El patrón de diseño DAO es un patrón que se utiliza para extraer la capa de presentación en una aplicación (no se ve) clases concretas. |
| 1. El patrón de diseño DAO es un patrón que se utiliza para abstraer la capa de acceso a datos en una aplicación (no se ve) y clases concretas. |
| 1. El patrón de diseño DAO es un patrón que se utiliza para abstraer la capa de infraestructura en una aplicación (no se ve) excepciones y bloques try-catch. |

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

|  |
| --- |
| 1. Un endpoint es la URL que se utiliza para acceder a una API REST. |
| 1. Un endpoint es un método que se utiliza para procesar datos en una API REST. |
| 1. Un endpoint es un controlador que se utiliza para administrar una API REST. |
| 1. Un endpoint es un objeto que se utiliza para almacenar datos en una API REST. |

¿Cuál de las siguientes opciones son verdaderas? (elija todas las correctas)

|  |
| --- |
| 1. Java es un lenguaje orientado a objetos. |
| 1. El código Java compilado en Windows puede ejecutarse en Linux. |
| 1. Java permite la sobrecarga de operadores |
| 1. Java es un lenguaje de programación funcional. |
| 1. Java es un lenguaje procedimental. |
| 1. Java tiene punteros a ubicaciones específicas en la memoria. |

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

|  |
| --- |
| 1. Maven es un lenguaje de programación. Se utiliza en el desarrollo de aplicaciones Java para escribir… (no se ve) |
| 1. Maven es un servidor de base de datos. Se utiliza en el desarrollo de aplicaciones java para alojar… (no se ve) |
| 1. Maven es un sistema de control de versiones. Se utiliza en el desarrollo de aplicaciones java para (no se ve) |
| 1. Maven es una herramienta de gestión de dependencias. Se utiliza en el desarrollo de aplicaciones… (no se ve) en el proyecto. |

1. ¿Cuál de lo siguiente es cierto? (elija todas las correctas)

|  |
| --- |
| 1. javac compila un archivo .java en un archivo .bytecode. |
| 1. Java toma el nombre del archivo .bytecode como parámetro. |
| 1. javac compila un archivo .java en un archivo .class |
| 1. Java toma el nombre de la clase como parámetro. |
| 1. Java toma el nombre del archivo .class como parámetro. |
| 1. javac compila un archivo .class como archivo java. |

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

|  |
| --- |
| 1. Git es un lenguaje de programación. Algunos comandos básicos de Git incluyen “print” e “if-else”. |
| 1. Git es una herramienta para el análisis de código. Algunos comandos básicos de Git incluyen “analyze”… (no se ve). |
| 1. Git es un sistema de control de versiones. Algunos comandos básicos de Git incluyen “commit” y “push”. |
| 1. Git es una herramienta para realizar pruebas de software. Algunos comandos básicos de Git incluyen… no se ve, pero no es correcta. |

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

|  |  |
| --- | --- |
| 1. int variableA = 10; float variableB = 10.5f; int variable = variableA + variableB; | El resultado de la suma (variableA + variableB) es un float. Al intentar asignar el resultado a una variable int, esto es un error de compilación porque no puedes asignar directamente un valor float a un int sin un casting explícito. |
| 1. byte variableA = 10; double variableB = 10.5f; double variableC = variableA + variable; |  |
| 1. byte variableA = 10; float variableB = 10.5f; float variable = variableA + variable; |  |

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

|  |
| --- |
| 1. La computadora se incendió. |
| 1. No sabe cómo codificar un método. |
| 1. No se encuentra un elemento al buscar en una lista. |
| 1. Se pasa un parámetro inesperado a un método. |
| 1. Quiere recorrer una lista. |

1. ¿Qué es un bean en Spring?

|  |
| --- |
| 1. Una herramienta de inyección de dependencias que se utiliza para inyectar en una clase… (no se ve) |
| 1. Una instancia de una clase que se administra por el contenedor de Spring. |
| 1. Un archivo de configuración XML que se utiliza para definir la estructura de una tabla de base de datos. |
| 1. Una clase que se utiliza para configurar la conexión a una base de datos. |

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

|  |
| --- |
| 1. Un método que toma un parámetro con tipo java.lang.Object tomará cualquier referencia. |
| 1. Una referencia a un objecto se puede convertir a una subclase del objecto sin una conversión explícita... (no se ve) |
| 1. Todas las excepciones de conversión se pueden detectar en tiempo de compilación. |
| 1. Al definir un método de instancia pública en la superclase, garantiza que el método específico se… (no se ve) |
| 1. Si un método toma una superclase de tres objetos, cualquiera de esas clases puede pasarse… (no se ve). |

1. ¿Qué son las pruebas de integración?

|  |  |
| --- | --- |
| 1. Pruebas que comprueben el rendimiento de la aplicación. | Pruebas de rendimiento. |
| 1. Pruebas que comprueben el funcionamiento de la interfaz de usuario. | Pruebas de interfaz de usuario |
| 1. Pruebas que comprueban el funcionamiento de varias unidades de código juntas. | pruebas de integración |
| 1. Pruebas que comprueban el funcionamiento de una sola unidad de código. | Pruebas unitarias. |

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

Texto

Descripción generada automáticamente

🡪 **sct** Este es el resultado recordando la regla de la llamada implícita super() de los constructores en herencia.

|  |
| --- |
| 1. t,s,c |
| 1. Error en tiempo de ejecución. |
| 1. No se ven las siguientes, pero recuerdo que la correcta no estaba, al menos no en el orden. |

1. ¿Son patrones de diseño de software estructural?

|  |  |
| --- | --- |
| 1. Adapter, Proxy, Prototype y Bridge. | Adapter: Estructural  Proxy: Estructural  Prototype: Creacional  Bridge: Estructural  Composite: Estructural  Agile: No es un patrón de diseño de software; es una metodología de desarrollo ágil.  Builder: Creacional  Singleton: Creacional  Abstract Factory: Creacional |
| 1. Adapter, Bridge, Proxy y Composite. |
| 1. Agile, Builder, Singleton y Prototype. |
| 1. Builder, Singleton y Prototype y Abstract Factory. |

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

Texto

Descripción generada automáticamente

|  |
| --- |
| 1. 3.0 🡪 El código realiza el promedio de los elementos de la lista |
| 1. 2.5 |
| 1. No se ven las demás, pero la correcta es 3.0 |

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

|  |
| --- |
| 1. git push |
| 1. it pull |
| 1. git commit |
| 1. git add |

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

|  |
| --- |
| 1. git branch |
| 1. git merge |
| 1. git commit |
| 1. git push |

1. ¿Qué es Git?

|  |
| --- |
| 1. Una herramienta de automatización de compilación que se utiliza para compilar y construir un proyecto (no se ve lo demás). |
| 1. Una herramienta de gestión de software de dependencias que se utiliza para descargar bibliotecas y paquetes en… no se ve |
| 1. Una herramienta de generación de informes que se utiliza para generar informes sobre el rendimiento… no se ve |
| 1. Una herramienta de control de versiones que se utiliza para almacenar y administrar el código… no se ve. |

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

Texto

Descripción generada automáticamente

|  |
| --- |
| 1. Línea 15 no compila porque other.ISBN es un atributo con modificador de acceso private. |
| 1. Línea 14 no compila porque no está declarada o manejada ClassCastException |
| 1. La clase compila satisfactoriamente. |

1. ¿Cuál es la primer línea en fallar al compilar?

🡪 Al heredar la clase y sobrescribir el método, el modificador de acceso no debe de ser más restrictivo

Texto

Descripción generada automáticamente

|  |
| --- |
| 1. r5 |
| 1. r4 |
| 1. r3 |
| 1. Ninguna de las anteriores. |

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

|  |
| --- |
| 1. ArrayIndexOutOfBoundException |
| 1. NumberFormatException |
| 1. ExceptionInInitializerError |
| 1. Java.io.IOException |
| 1. NullPointerException |

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

|  |
| --- |
| 1. git Branch |
| 1. git merge |
| 1. git push |
| 1. git pull |

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

|  |
| --- |
| 1. REST es un protocolo de comunicación. Su relación con las API web es que se utiliza para definir los elementos… no se ve |
| 1. REST es un estilo de arquitectura para aplicaciones web. Su relación con las API web es que se utiliza… no se ve |
| 1. REST es un servicio en la nube. Su relación con las API web es que se utiliza para alojar las aplicaciones… no se ve. |
| 1. REST es un lenguaje de programación. Su relación con las API web es que se utiliza para crear… no se ve. |

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

|  |
| --- |
| 1. git checkout |
| 1. git clone |
| 1. git push |
| 1. git pull |

**EXAMEN ASO A2**

1. ¿Qué es un microservicio?

|  |
| --- |
| 1. Son componentes que se pueden desplegar de forma independiente, de función múltiple, es decir… no necesariamente están relacionados. |
| 1. Es un componente que se pueden desplegar de forma independiente y suelen ser de función única, es decir, que están estrechamente relacionados. --------- |
| 1. Es el conjunto de endpoints contenidos en múltiples desarrollos que se despliegan en conjunto y que están estrechamente relacionados. |
| 1. Ninguna de las anteriores |

1. Dado el siguiente código:

🡪 En la condición del <= ciclo for números.length hace que suma intente acceder a un índice que no existe por lo tanto el error sería en tiempo de ejecución lanzando una excepción

Texto

Descripción generada automáticamente

¿Este código compila sin errores?

|  |
| --- |
| 1. Si, compila sin errores. ------------ |
| 1. No, hay un error en el ciclo for. |
| 1. No, hay un error en la inicialización de la variable “suma”. |
| 1. No, hay un error en la declaración del arreglo. |

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

|  |
| --- |
| 1. getClass() |
| 1. printStackTrace() |
| 1. toString() |
| 1. getMessage() ------- |

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

|  |
| --- |
| 1. Puede declarar solo excepciones no comprobadas (unchecked). |
| 1. Las excepciones en tiempo de ejecución son lo mismo que las excepciones no comprobadas. |
| 1. Las excepciones en tiempo de ejecución son lo mismo que las excepciones comprobadas. |
| 1. Solo puede declarar excepciones comprobadas (checked) |
| 1. Solo puede manejar subclases de Exception. |

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

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

🡪 String es un objeto inmutable, por lo tanto, no se modifica, se crea un nuevo objeto del cual no tiene la asignación de referencia

|  |
| --- |
| 1. NullPointerException |
| 1. hello |
| 1. HELLO |
| 1. Hello |

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

|  |
| --- |
| 1. git commit |
| 1. git branch |
| 1. git merge |
| 1. git push |

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

|  |
| --- |
| 1. import.java.net.\*; |
| 1. import.java.awt.\*; |
| 1. import.java.io. \*; |
| 1. import.java.util.\*; |

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

|  |
| --- |
| 1. YAML |
| 1. XML |
| 1. JSON |
| 1. Todos los anteriores |

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

|  |
| --- |
| 1. El operador doble dos puntos se utiliza para crear una nueva instancia de una clase en Java 8.   Esto es parcialmente cierto porque se puede usar ClassName::new para crear una referencia a un constructor, y al invocar esa referencia, se crea una nueva instancia pero no crea por si sola la instancia. |
| 1. EL operador de doble dos puntos no se utiliza en java 8. |
| 1. El operador de doble dos puntos se utiliza para acceder a métodos estáticos en Java 8. |
| 1. El operador de doble dos puntos se utiliza para acceder a métodos no estáticos en Java 8. |

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

|  |
| --- |
| 1. try |
| 1. throw 🡪 *Propagación* |
| 1. finally |
| 1. catch |

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

|  |  |
| --- | --- |
| 1. mvn site | 🡪 Se utiliza para generar la documentación del proyecto (un sitio web del proyecto). No limpia el directorio target por defecto. |
| 1. mvn build | Este no es un comando estándar de Maven. |
| 1. mvn answer | Este no es un comando estándar de Maven. |
| 1. mvn clean ------- | 🡪Este es el comando específico de Maven diseñado para **limpiar el proyecto**, lo que por defecto implica la eliminación del directorio target |

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

|  |  |
| --- | --- |
| 1. git log ----------- | 🡪 Muestra un registro de los commits realizados en el repositorio |
| 1. git status | 🡪 Muestra el estado del directorio de trabajo y del área de preparación |
| 1. git commit | 🡪 Se utiliza para guardar los cambios confirmados en el historial del repositorio creando un nuevo commit. |
| 1. git diff | 🡪 Muestra las diferencias entre dos puntos en el historial (entre commits, ramas, etc.) |

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

|  |
| --- |
| 1. Una expresión lambda es una forma de escribir una clase anónima en Java 8. |
| 1. Una expresión lambda es una forma de escribir una función anónima en Java 8. |
| 1. Una expresión lambda es un método que se llama automáticamente cuando se crea un objeto. |
| 1. Una expresión lambda es un método que se llama de forma explícita desde el código. |

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

|  |
| --- |
| 1. Una interfaz solo puede heredar de una clase, mientras una clase abstracta puede heredar de múltiples interfaces |
| 1. Una clase abstracta puede contener implementaciones de métodos, mientras que una interfaz no puede. |
| 1. Una clase abstracta puede contener variables de instancia, mientras que una interfaz no puede. |
| 1. Una interfaz puede contener implementaciones de métodos, mientras que una clase abstracta no puede. |

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

Texto

Descripción generada automáticamente

🡪 Debido a la ausencia de sentencias break, el código imprimirá todas las cadenas desde el punto donde se encontró la primera coincidencia (hasta el final del bloque switch.

|  |
| --- |
| 1. Dos |
| 1. Uno Dos Tres Otro número |
| 1. Uno |
| 1. Otro número |

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

|  |
| --- |
| 1. REST es un protocolo de comunicación. Su relación con las API web es que las utiliza para definir los endpoints de una API //*FALSO* |
| 1. REST es in lenguaje de programación. Su relación con las API web es que se utiliza para crear aplicaciones web. //FALSO |
| 1. REST es un servicio en la nube. Su relación con las API web que se utiliza para alojar las aplicaciones web. //FALSO |
| 1. 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. |

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

|  |  |
| --- | --- |
| 1. java.net | 🡪 Para operaciones de red. |
| 1. java.io | 🡪 Para operaciones de entrada/salida (archivos, streams). |
| 1. javax.swing | 🡪 Es un kit de herramientas GUI más rico y flexible que AWT. Proporciona un  conjunto más amplio de componentes (tablas, árboles, barras de progreso, etc.) |
| 1. java.util | 🡪 Contiene clases de utilidad y colecciones. |
| 1. java.awt | 🡪 Fue el primer kit de herramientas para GUI en Java. Contiene clases  fundamentales para crear ventanas, botones, etiquetas, eventos |

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

|  |
| --- |
| 1. Solo N |
| 1. Solo ? |
| 1. Ninguna de las anteriores |
| 1. News,y Object |
| 1. ? y N |
| 1. N, News y Object ------------- |

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

|  |
| --- |
| 1. Un objeto que representa una conexión de entrada o salida de datos. |
| 1. Un objeto que representa una secuencia de elementos y se utiliza para procesar colecciones de forma declarativa. |
| 1. Un objeto que se utiliza para leer y escribir archivos de texto. |
| 1. Un objeto que se utiliza para crear y manipular bases de datos. |

1. Son patrones de diseño de microservicios

|  |  |
| --- | --- |
| 1. Circuit Breaker, Adaptative Lifo, MQ Strategy | 🡪 Circuit Breaker  🡪 Retry  🡪 Bulkhead |
| 1. System, Process y Client |
| 1. Retry, Circuit Breaker, Adaptative Lifo y Bulkhead ------------ |
| 1. Ninguna de las anteriores |

1. ¿Qué afirmaciones son verdaderas tanto para las clases abstractas como para las interfaces? (Elije todas las correctas)

|  |
| --- |
| 1. Ambos pueden contener métodos estáticos. --- |
| 1. Ambos se pueden ampliar con la clave extend. |
| 1. Ambos pueden contener métodos predeterminados. |
| 1. Ambos heredan de java.lang.Object. |
| 1. Ninguno de los dos puede ser instanciado directamente. --- |
| 1. Ambos pueden contener variables finales estáticas públicas. -- |
| 1. Supone que todos los métodos dentro de ellos son abstractos. |

1. ¿Cuál no es un objetivo de Maven?

|  |
| --- |
| 1. Clean |
| 1. Package |
| 1. Debug --- |
| 1. Install |

1. ¿Si deseas obtener una copia de un repositorio Git existente en un servidor qué comando se utiliza?

|  |
| --- |
| 1. git commit |
| 1. git log |
| 1. git clone |
| 1. git add |

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

|  |
| --- |
| 1. git push |
| 1. git branch |
| 1. git pull |
| 1. git merge |

1. ¿Qué es un repositorio remoto en Git?

|  |
| --- |
| 1. Una herramienta que se utiliza para compartir y fusionar cambios entre diferentes ramas de un repositorio. |
| 1. Una copia local de un repositorio que se utiliza para hacer cambios en el código fuente. |
| 1. Un servidor Git que almacena una copia central del repositorio. |
| 1. Un archivo que contiene una instantánea del código fuente en un momento determinado. |

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

|  |
| --- |
| 1. git pull |
| 1. git push |
| 1. git clone |
| 1. git checkout |

1. Dada la siguiente clase

Texto

Descripción generada automáticamente

¿Cuál de las siguientes instrucciones puede colocarse en la línea 9 para que la clase Helper compile?

|  |
| --- |
| 1. Helper.printException(new Exception(“B”)); |
| 1. Helper. printException(new FileNotFoundException(“A”)); |
| 1. Helper.<Throwable>printException(new Exception(“C”));   🡪 no cumple con el límite extends Exception |
| 1. Helper.<NullPointerException>printException(new NullPointerException (“D”)); |
| 1. Helper. printException(new Throwable(“E”)); 🡪 no extiende Exception |

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

Texto

Descripción generada automáticamente

|  |
| --- |
| 1. 51tuna |
| 1. 5tuna |
| 1. 5 |
| 1. 41 |
| 1. 5 tuna |
| 1. 4 1 |
| 1. El código no compila 🡪 No se puede agregar un valor int a String |

1. ¿A qué hace referencia el principio de Liskov?

|  |  |
| --- | --- |
| 1. Nos indica que una clase no debe tener solo una funcionalidad sino varias para reducir el uso de objetos. | Describe el Principio de Responsabilidad Única (Single Responsibility Principle - SRP). |
| 1. Este principio nos indica que dentro del programa una clase puede ser sustituida por cualquier clase que se extienda de ella sin alterar el comportamiento del programa. | Describe el Principio de Sustitución de Liskov |
| 1. Nos indica que cualquier clase se puede extender para agregar funcionalidad, pero no se puede modificar. | Describe el Principio Abierto/Cerrado (Open/Closed Principle - OCP). |
| 1. Este principio nos indica que dentro del programa una clase puede ser sustituida por su clase padre sin alterar el comportamiento del programa. | Describe lo contrario del Principio de Liskov. |

**EXAMEN ASO B**

1. Cuál es el valor de x e y al final el programa?

Pantalla de computadora con letras

Descripción generada automáticamente

|  |
| --- |
| 1. X=9 y=10 |
| 1. X=10 y=9 |
| 1. X=10 y=10 |
| 1. X=9 y=9 |

1. ¿Dado el siguiente enum y clase cuál es la opción que puede ir en el espacio en blanco para que el código compile?

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

🡪 La línea en blanco está dentro de la etiqueta default, solo deben de ser sentencias ejecutables

|  |
| --- |
| 1. Ninguno de los anteriores |
| 1. case SUMMER -> |
| 1. case Season.Winter: |
| 1. case FALL: |
| 1. case Winter, Spring: |
| 1. case SUMMER | WINTER: |

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

|  |
| --- |
| 1. Hacer que las pruebas dependan de otras pruebas |
| 1. probar solo una pequeña parte de una función |
| 1. ejecutar pruebas con poca frecuencia |
| 1. asegurarse de que las pruebas sean claras y concisas |

1. Cuál es la diferencia entre las anotaciones @restController , @component , @service y @repository?

|  |
| --- |
| 1. No existe diferencia funcional entre ellas sino semántica las cuatro son anotaciones de Spring que crean un bean y lo agregan al contexto de Spring |
| 1. @Controller @component son anotaciones que crean bean y exponen la serialización de las clases mientras que @service y @repository requieren de una iniciación manual |
| 1. @controler es una anotación que nos ayuda a construir un API rest mientras que @service @component y @repository solo marcan las clases que se deben de inicializar |
| 1. @service y @repository son anotaciones que crean bean y exponen la serialización de las clases mientras que @Controller @component requieren de una inicialización manual. |

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

|  |
| --- |
| 1. Import java.util.\*; |
| 1. Import java.util; |
| 1. #include java.util; |
| 1. #include java.util.\*; |

1. ¿Resultado de compilar y ejecutar el siguiente programa?

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

|  |
| --- |
| 1. False |
| 1. no se produce salida |
| 1. true |
| 1. error de compilación |

1. cómo se manejan las excepciones en Java?

|  |
| --- |
| 1. Las excepciones el manejan con bloques switch case en Java. La excepción trae with Resources es una forma de lanzar una excepción en un método. |
| 1. Las excepciones se manejan con bloques while en Java. La excepción trae with Resources es una forma de manejar excepciones de compilación |
| 1. las excepciones se manejan con bloques if else en Java. La excepción trae with resource es una forma de manejar las excepciones en tiempo de ejecución. |
| 1. Las excepciones se manejan con bloques try catch finally en Java. La excepción trae with Resources es una forma de cerrar automáticamente los recursos abiertos en un bloque try. |

1. ¿Cuáles son los cuatro pilares de la programación orientada a objetos?

|  |
| --- |
| 1. Polimorfismo, abstracción, herencia y encapsulamiento |
| 1. polimorfismo coerción herencia y encapsulamiento |
| 1. polimorfismo herencia encapsulamiento y sincronía |
| 1. encapsulamiento coerción polimorfismo y abstracción |

1. ¿Qué clase del paquete java.io permite leer y escribir archivos en ubicaciones específicas dentro de un archivo?

|  |  |
| --- | --- |
| 1. File | Representa un archivo o directorio, no permite leer/escribir contenido en posiciones específicas. |
| 1. filename filter | Es una interfaz para filtrar nombres de archivos. |
| 1. file descriptor | Representa un descriptor de archivo de bajo nivel |
| 1. RandomAccessFile | Permite leer y escribir archivos en ubicaciones específicas dentro de un archivo |

1. Todas las siguientes definiciones de clases my School classroom y my City School ¿qué números de línea en el método main generan un error de compilación?(Elija todas las opciones correctas)

Texto, Carta

Descripción generada automáticamente

|  |
| --- |
| 1. Ninguna, el código compila bien |
| 1. línea 6 🡪 constructor default |
| 1. línea 9 🡪 atributo con modificador protected |
| 1. línea 7 🡪 atributo private |
| 1. línea 8 |
| 1. línea 5 🡪 atributo estático default |

1. ¿Qué es un microservicio?

|  |
| --- |
| 1. Es un componente que se puede desplegar de forma independiente, y que suele ser de función única es decir englobando en point que están estrechamente relacionados |
| 1. es el conjunto de inputs contenidos en múltiples desarrollos que se despliegan en conjunto y que están estrechamente relacionados |
| 1. es un componente que se puede desplegar de forma independiente, de función múltiple es decir englobando en point que no necesariamente están relacionados |
| 1. ninguna de las anteriores |

1. ¿Qué hace el siguiente programa?

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

|  |
| --- |
| 1. calcula si una palabra es un palíndromo |
| 1. cuentas las letras que hay en una palabra |
| 1. calcula si hay letras repetidas en una palabra |
| 1. el programa no compila |

1. ¿Qué es el siguiente código fuente?

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

|  |
| --- |
| 1. Muestra los números del 1 al 10 |
| 1. muestra un 10 |
| 1. se queda en un bucle infinito |
| 1. muestra los números del 0 al 9 |

**EXAMEN EDUARDO**

1. ¿Qué es un “code smell”?

|  |
| --- |
| 1. Un componente de la biblioteca estándar de Java |
| 1. Un error en tiempo de ejecución que se produce en Java |
| 1. 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 |
| 1. Una práctica de programación recomendada en Java |

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

|  |
| --- |
| 1. Code, Register, Update, Debug |
| 1. Create, Read, Update, Delete |
| 1. Call, Recive, Use, Debug |
| 1. Customize, Request, Use, Debug |

1. ¿Para qué nos sirve utilizar un profile dentro del archive pom.xml?

|  |
| --- |
| 1. Etiqueta por la cual podemos definir la versión de nuestras dependencias |
| 1. Es la etiqueta por la cual podemos definir las características que tendrá nuestro proyecto al ser compiladas |
| 1. Etiqueta por la cual definimos los parámetros de conexión a un repositorio |
| 1. No existe esta etiqueta en maven |

1. Texto

   El contenido generado por IA puede ser incorrecto.Texto

   El contenido generado por IA puede ser incorrecto.Dadas las siguientes clases Vehicle y Car

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

|  |
| --- |
| 1. milage = 15285; |
| 1. Ninguna de las anteriores |
| 1. make = “Honda”; |
| 1. year = 2009; |
| 1. model = “Pilot”; |

1. Enumere cuatro interfaces de la API de colecciones:

|  |
| --- |
| 1. List, Map, Set, Queue |
| 1. ArrayList, Map, Set, Queue |
| 1. List, HashMap, HashSet, PriorityQueue |
| 1. List, Map, HashSet, PriorityQueue |

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

Texto

El contenido generado por IA puede ser incorrecto.

🡪 Coin.overload(primerIntento): primerIntento es de tipo Side. El método más específico aplicable es overload(Object). Imprime "Object ".

Coin.overload(segundoIntento): segundoIntento es de tipo Side. El método más específico aplicable es overload(Object). Imprime "Object ".

Coin.overload((Object)segundoIntento): El argumento es de tipo Object. El método más específico aplicable es overload(Object). Imprime "Object ".

Coin.overload((Side)primerIntento): El argumento es de tipo Side. El método más específico aplicable es overload(Object). Imprime "Object ".

|  |
| --- |
| 1. Head Object Tail Side |
| 1. No compila |
| 1. Side |
| 1. Object Tail Side |
| 1. Head Head Tail Tail |
| 1. Side Head Tail Side |

1. Cuál es la salida al ejecutar el siguiente código

Texto

El contenido generado por IA puede ser incorrecto.

|  |
| --- |
| 1. roar roar!!! |
| 1. roar!!! roar |
| 1. Se lanza una excepción |
| 1. roar!!! roar!!! |
| 1. roar roar |
| 1. El código no compila |

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

|  |
| --- |
| 1. Una subclase concreta no se puede marcar como final |
| 1. Una subclase concreta debe implementar todos los métodos definidos en una interfaz heredada |
| 1. Una subclase concreta debe implementar todos los métodos abstractos heredados |
| 1. Una subclase concreta puede declararse como abstracta |
| 1. Los métodos abstractos no pueden ser anulados por una subclase concreta |

1. ¿Cuál es la salida del siguiente código?

|  |
| --- |
| 1. Error compilación linea 12 |
| 1. Error compilación línea 16 |
| 1. Caught string |
| 1. Error compilación línea 2 |
| 1. Caught object |

Texto

El contenido generado por IA puede ser incorrecto.

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

Texto

El contenido generado por IA puede ser incorrecto.

|  |
| --- |
| 1. 3 |
| 1. 9 |
| 1. 14 |
| 1. 6 |

1. ¿Qué declaración representa una declaración valida que permitirá la inclusión de la clase del paquete java.util?

|  |
| --- |
| 1. #include java.util.\*; |
| 1. #include java.util; |
| 1. Import java.util.\*; |
| 1. Import java.util; |

1. ¿Que es la cobertura de código?

|  |  |
| --- | --- |
| 1. La cantidad de veces que se ejecuta una línea de código |  |
| 1. La cantidad de errores detectados por una prueba | Esto es sobre la efectividad de las pruebas para encontrar errores. |
| 1. La cantidad de tiempo que tarda una prueba en ejecutarse | Esto es rendimiento de la prueba. |
| 1. La cantidad de código que se ejecuta durante una prueba | Se expresa típicamente como un porcentaje |
| 1. Una herramienta de automatización de compilación que se utiliza para compilar y construir un proyecto | Esto describe herramientas como Maven o Gradle. |

1. Which one do you like. ¿Cuál es el formato correcto para hacer un commit en Git?

|  |
| --- |
| 1. Descripción breve de cambio y nombre del autor |
| 1. Tipo de cambio, descripción breve, cuerpo opcional y notas de pie de pagina |
| 1. Solo se necesita una breve descripción del cambio |
| 1. Nombre de la rama, descripción detallada del cambio y fecha |

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

|  |
| --- |
| 1. 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 puede implementar en Java 8 utilizando una variable estática y un constructor privado |
| 1. El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de infraestructura en una aplicación. Se puede implementar en Java 8 utilizando excepciones y bloques try-catch |
| 1. El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de presentación en una aplicación. |
| 1. Se puede implementar en Java 8 utilizando interfaces y clases concretas |
| 1. El patrón de diseño Singleton es un patrón que se utiliza para abstraer la capa de negocios en una aplicación. Se puede implementar en Java 8 utilizando clases abstractas y métodos estáticos |

1. En los verbos REST, ¿cuál es la diferencia en el uso de PATCH y PUT?

|  |
| --- |
| 1. Son exactamente iguales, no hay diferencia en su uso |
| 1. PATCH requiere se le envié la entidad completa mientras que PUT solo los atributos a modificar |
| 1. PUT requiere se le envié la calidad completa mientras que PATCH solo los atributos a modificar |
| 1. PATCH es un verbo deprecado sustituido por PUT |

1. ¿Cuál es la ventaja principal de usar API Rest sobre otros tipos de servicios web?

|  |
| --- |
| 1. Mayor seguridad |
| 1. Mayor facilidad de implementación |
| 1. Mayor velocidad de transferencia de datos |
| 1. Mayor compatibilidad con diferentes plataformas |

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

Texto

El contenido generado por IA puede ser incorrecto.

|  |
| --- |
| 1. No compila |
| 1. [25,7,25,67]   [67,7,25]  [7,25,67]  [67,7,25]  [7,25,67] |
| 1. [25,7,67]   [67,7,25]  [7,25,67]  [67,7,25]  [7,25,67] |
| 1. [67,7,25]   [67,7,25]  [67,7,25]  [67,7,25]  [67,7,25] |
| 1. [25,7,25,67]   [7,25,67]  [67,7,25]  [7,25,67]  [67,7,25] |

Texto

El contenido generado por IA puede ser incorrecto.

21 ¿Qué utilidad de línea de comandos basada en MS Windows te permite ejecutar el interprete de Java sin abrir la ventana de la consola?

|  |
| --- |
| 1. jconsole |
| 1. javaw |
| 1. interpw |
| 1. java -wo |