Java Backend Básico

Reactivos

Pregunta 1

¿Por qué se considera a Java como un lenguaje multiplataforma?

Opciones:

a. Porque un programa escrito en Java se puede ejecutar en plataformas que tengan instalado Java Runtime Environment.
b. Porque solo funciona en Windows.
c. Porque utiliza una sintaxis derivada de C y C++.
d. Porque es propiedad de Oracle.

Pregunta 2

¿Qué diferencia existe entre JRE y Java SE?

Opciones:

a. Ambos son lo mismo.
b. JRE es necesario para ejecutar aplicaciones en Java, mientras que Java SE es un kit de desarrollo.
c. JRE es un lenguaje de programación y Java SE es un entorno de ejecución.
d. Java SE es una versión gratuita de Java.

Pregunta 3

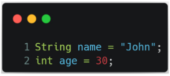
¿Qué caracteriza a los tipos de datos primitivos en Java?

Opciones:

a. Son creados por el programador.
b. Comienzan con una letra mayúscula.
c. Siempre tienen un valor y están predefinidos en Java.
d. Pueden ser nulos.

Pregunta 4

Dado el siguiente código en Java, ¿qué tipo de dato es *name* y qué tipo de dato es *age*?



```java
String name = "John";
int age = 30;
```

Opciones:

a. Name es primitivo y age es no primitivo.
b. Ambos son primitivos.
c. Ambos son no primitivos.
d. Name es no primitivo y age es primitivo.

Pregunta 5

Si estás trabajando en una empresa que requiere soporte y actualizaciones constantes para su software en Java, ¿qué versión de Java sería más adecuada?

Opciones:

a. Open JDK
b. Java Runtime Environment (JRE)
c. Java EE
d. Java SE

Pregunta 6

¿Qué operador de asignación en Java se utiliza para sumar un valor a una variable y asignarle el resultado?

Opciones:

a. =
b. -=
c. +=

d. /=

Pregunta 7

¿Qué operador aritmético en Java se utiliza para obtener el residuo de una división?

Opciones:

a. +
b. %
c. -
d. *

Pregunta 8

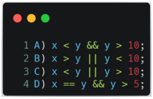
Si tienes dos condiciones y deseas que ambas se cumplan para que una afirmación sea verdadera, ¿qué operador lógico usarías en Java?

Opciones:

a. &&
b. ||
c. !
d. ==

Pregunta 9

Si tienes dos variables, x = 5 y y = 10, y deseas verificar si x es menor que y y y es mayor que 10, ¿cuál sería la expresión correcta?



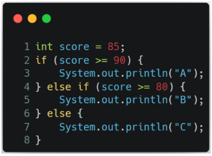
```
A) x < y && y > 10;
B) x > y || y < 10;
C) x < y || y > 10;
D) x == y && y > 5;
```

Opciones:

a. B)
b. A)
c. C)
d. D)

Pregunta 10

Dado el siguiente código en Java, ¿Qué se imprimirá en la consola?



```java
int score = 85;
if (score >= 90) {
    System.out.println("A");
} else if (score >= 80) {
    System.out.println("B");
} else {
    System.out.println("C");
}
```

Opciones:

a. A
b. C
c. B
d. Ninguno

Pregunta 11

¿Qué permite la programación orientada a objetos en comparación con la programación estructurada?

Opciones:

a. Limita la reutilización del código.
b. Prohíbe la ampliación del código.
c. Impide la creación de sistemas complejos.
d. Trata el código como objetos del mundo real.

Pregunta 12

¿Cuál de los siguientes NO es uno de los pilares fundamentales de la programación orientada a objetos?

Opciones:

a. Segmentación.
b. Abstracción.
c. Encapsulamiento.
d. Polimorfismo.

Pregunta 13

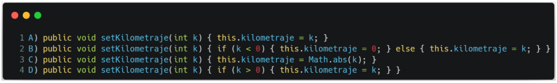
¿Qué caracteriza a un constructor en Java?

Opciones:

a. Retorna un valor.
b. Tiene un nombre diferente al de la clase.
c. Se llama automáticamente al crear un objeto de una clase.
d. No puede existir más de un constructor en una clase.

Pregunta 14

Si deseas que el atributo kilometraje de la clase Auto no pueda tener valores negativos, ¿cuál sería la forma correcta de definir el setter?



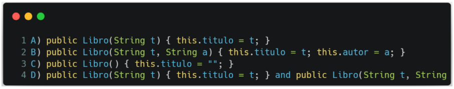
```java
A) public void setKilometraje(int k) { this.kilometraje = k; }
B) public void setKilometraje(int k) { if (k < 0) { this.kilometraje = 0; } else { this.kilometraje = k; } }
C) public void setKilometraje(int k) { this.kilometraje = Math.abs(k); }
D) public void setKilometraje(int k) { if (k > 0) { this.kilometraje = k; } }
```

Opciones:

a. A)
b. B)
c. C)
d. D)

Pregunta 15

Si tienes una clase Libro y deseas que cada libro tenga un título y un autor al ser instanciado, pero también quieres que sea posible crear un libro sin especificar el autor, ¿cuál sería la forma correcta de definir los constructores?



```java
A) public Libro(String t) { this.titulo = t; }
B) public Libro(String t, String a) { this.titulo = t; this.autor = a; }
C) public Libro() { this.titulo = ""; }
D) public Libro(String t) { this.titulo = t; } and public Libro(String t, String
```

Opciones:

a. D)
b. C)
c. B)
d. A)

Pregunta 16

¿Qué modificador de acceso en Java permite que una clase sea accesible solo por clases en el mismo paquete?

Opciones:

a. Public
b. Private
c. Protected
d. Default

Pregunta 17

Si un método en Java tiene el modificador *static*, ¿qué significa esto?

Opciones:

a. El método pertenece a la clase en lugar de a un objeto específico.
b. El método debe ser sobrescrito en todas las subclases.
c. El método puede acceder directamente a variables de instancia.
d. El método no puede ser llamado sin instanciar un objeto.

Pregunta 18

¿Qué representa la herencia en la programación orientada a objetos en Java?

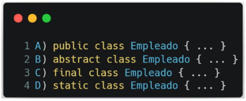
Opciones:

a. La capacidad de una clase de copiar métodos de otra clase.
b. La capacidad de una clase de heredar atributos y métodos de otra clase.
c. La restricción de una clase para acceder a métodos de otra clase.

d. La capacidad de una clase de modificar métodos de otra clase.

Pregunta 19

Si estás diseñando una clase Empleado, y quieres asegurarte de que ninguna otra clase pueda heredar de ella, ¿qué modificador deberías usar?



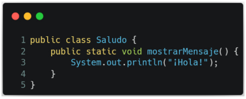
```
A) public class Empleado { ... }
B) abstract class Empleado { ... }
C) final class Empleado { ... }
D) static class Empleado { ... }
```

Opciones:

a. A
b. B
c. C
d. D

Pregunta 20

Dado el siguiente código, ¿cuál es la forma correcta de acceder al método “mostrar Mensaje” sin crear una instancia de la clase “Saludo”?



```
public class Saludo {
    public static void mostrarMensaje() {
        System.out.println("¡Hola!");
    }
}
```

Opciones:

a. new Saludo().mostrarMensaje();
b. Saludo->mostrarMensaje();
c. Saludo::mostrarMensaje();
d. Saludo.mostrarMensaje();

Pregunta 21

¿Cuál es el propósito principal de una API?

Opciones:

a. Mejorar la velocidad de una aplicación.
b. Cambiar el diseño de una aplicación.
c. Traducir un lenguaje de programación a otro.
d. Permitir la interacción con un sistema para obtener datos o ejecutar funcionalidades.

Pregunta 22

¿Qué método HTTP se utiliza para solicitar una representación de un recurso y recuperar datos?

Opciones:

a. POST
b. GET
c. DELETE
d. PUT

Pregunta 23

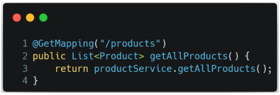
Para que una API se considere RESTful, ¿cuál de los siguientes criterios NO es necesario?

Opciones:

a. Que la API esté escrita en Java.
b. Una arquitectura cliente-servidor.
c. Que la comunicación entre el cliente y el servidor sea sin estado.
d. Datos que se almacenen en caché.

Pregunta 24

Dado el siguiente código en Java, ¿qué hace este fragmento de código?



```
@GetMapping("/products")
public List<Product> getAllProducts() {
    return productService.getAllProducts();
}
```

Opciones:

a. Actualiza la lista de productos.
b. Elimina todos los productos.
c. Obtiene la lista completa de productos.
d. Crea un nuevo producto.

Pregunta 25

Si deseas enviar datos para crear un nuevo producto en tu API RESTful, ¿qué método HTTP y formato preferido usarías basándote en las prácticas comunes?

Opciones:

a. GET y HTML
b. POST y JSON
c. PUT y XLT
d. DELETE y Python

Pregunta 26

¿Qué lenguajes de programación son compatibles con Spring Boot para desarrollar aplicaciones?

Opciones:

a. Java, Kotlin y Groovy.
b. Python y Ruby.
c. C++ y PHP.
d. JavaScript y TypeScript.

Pregunta 27

¿Cuál es la función principal de un controlador en un proyecto Spring Boot?

Opciones:

a. Almacenar datos en una base de datos.
b. Configurar propiedades de la aplicación.
c. Definir todos los endpoints que tenga la aplicación.
d. Instalar dependencias del proyecto.

Pregunta 28

¿Cuál es el principal objetivo de utilizar marcos de trabajo como Spring Boot en el desarrollo de aplicaciones?

Opciones:

a. Aumentar el costo de desarrollo.
b. Limitar la reutilización de componentes.
c. Evitar la creación de aplicaciones web.
d. Reducir el tiempo y costo necesario para desarrollar una aplicación.

Pregunta 29

Estás utilizando Postman para probar tu API desarrollada con Spring Boot. Si deseas actualizar la información de un registro específico de la entidad "Usuario" con el ID 5, ¿cuál sería el método y URL adecuados?

Opciones:

a. GET - Localhost:8080/usuario/5
b. POST - Localhost:8080/usuario/
c. DELETE - Localhost:8080/usuario/5
d. UPDATE - Localhost:8080/usuario/5

Pregunta 30

Estás trabajando en un proyecto con Spring Boot y necesitas agregar una nueva funcionalidad. ¿Dónde deberías colocar las clases que contienen la lógica y funcionalidad de esta nueva característica?

Opciones:

a. En el paquete "controller".
b. En el paquete "model".
c. En el paquete "service".
d. En el paquete "repository".

Pregunta 31

¿Qué es CRUD en el contexto de aplicaciones y bases de datos?

Opciones:

a. Un tipo de base de datos especializada.

b. Un conjunto de operaciones básicas: Create, Read, Update, Delete.
c. Un protocolo de seguridad para bases de datos.
d. Un lenguaje de programación para bases de datos.

Pregunta 32

En Java, ¿Qué es Hibernate?

Opciones:

a. Un ORM liberado para el lenguaje de programación Java.
b. Un lenguaje de programación basado en Java.
c. Un protocolo de seguridad para aplicaciones Java.
d. Un tipo de base de datos optimizada para Java.

Pregunta 33

¿Qué representa la anotación @Entity en el contexto de Java y bases de datos?

Opciones:

a. Indica que una clase es una interfaz.
b. Especifica que una clase es abstracta.
c. Define que una clase es un controlador.
d. Señala que una clase es una entidad en la base de datos.

Pregunta 34

Estás trabajando con Hibernate y necesitas especificar una relación uno-a-uno entre dos entidades. ¿Qué anotación usarías?

Opciones:

a. @OneToMany
b. @ManyToOne
c. @OneToOne
d. @AllToOne

Pregunta 35

Si estás utilizando Hibernate y quieres indicar que un atributo es una columna en la base de datos, ¿qué anotación usarías?

Opciones:

a. @DatabaseColumn
b. @Column
c. @Attribute
d. @Field

Pregunta 36

¿Qué es Spring Security?

Opciones:

a. Un marco de autenticación y control de acceso personalizable.
b. Una base de datos para aplicaciones Spring.
c. Un lenguaje de programación basado en Java.
d. Una herramienta de diseño de interfaces para aplicaciones Spring.

Pregunta 37

¿Qué es JWT?

Opciones:

a. Un tipo de base de datos.
b. Un estándar basado en JSON para autenticación.
c. Un algoritmo de encriptación.
d. Una herramienta de diseño gráfico.

Pregunta 38

¿Cuál es el propósito de los diferentes entornos como DEVELOPMENT, QA, PREPRODUCCIÓN y PRODUCCIÓN?

Opciones:

a. Diferentes versiones de una aplicación para diferentes sistemas operativos.
b. Diferentes niveles de acceso en una aplicación.
c. Espacios independientes donde la aplicación puede adquirir un comportamiento distinto.
d. Diferentes interfaces de usuario para una aplicación.

Pregunta 39

Estás en el proceso de proteger las contraseñas en tu aplicación. ¿Qué librería te ayudaría a proteger las contraseñas mediante Bcrypt?

Opciones:

a. JWT
b. Spring Security
c. JSON Web Tokens
d. Argon2

Pregunta 40

Estás en el proceso de configurar la seguridad de tu aplicación en Spring. ¿Cuál es el estándar para proteger las aplicaciones desarrolladas en Spring?

Opciones:

a. Spring Security
b. JWT
c. Argon2
d. JSON