

# Cuestionario Back End

\* Required

## Información general

1. Nombre y apellidos: \*

Cristian Leonardo Benitez

2. ¿Cuál es el motivo por el que quieres participar en este proyecto? \*

Me presenta nuevos desafios, con lo ultimo en tecnologias y ademas buenas practicas

3. ¿Qué versión de Java utilizas? ¿Cuál es el motivo? \*

Actualmente java 11 pero tambien java 8 por proyectos de migracion a java 11

#### 4. ¿Qué ventajas ves en la utilización de expresiones lambda? \*

Las expresiones lambda son una herramienta valiosa para los lenguajes de programación pues ofrecen una forma más concisa y eficaz de escribir funciones. Estas expresiones permiten la creación de funciones anónimas, que se pueden usar para tareas que no necesitan definirse en una función separada. Esto puede ahorrar tiempo y esfuerzo, lo que genera un código más legible. Las expresiones lambda también brindan una variedad de funciones útiles, como la conversión implícita de tipos y la capacidad de pasar una cantidad ilimitada de argumentos. Esto las vuelve versátiles y aplicables a una

#### 5. ¿Por qué es útil la API de Streams de Java? \*

El API de Java es una poderosa herramienta para procesar colecciones de datos. Permite realizar operaciones rápida y fácilmente en colecciones de datos, como transformación, filtrado y clasificación. La API está diseñada para ser altamente eficiente, lo que la hace ideal para aplicaciones que necesitan procesar gran cantidad de datos en poco tiempo. Permite escribir código más conciso y legible, y realizar operaciones de datos complejas en una sola línea de código. La API de Streams también permite incorporar

#### 6. ¿Qué ventajas principales ves en utilizar Spring Boot? \*

Spring Boot brinda muchas ventajas cuando se trata de crear e implementar aplicaciones.

Los principales beneficios incluyen:

1. Instalación y configuración sencillas: Spring Boot requiere muy poca configuración y esfuerzo de instalación, lo que facilita su uso y permite concentrarse en la aplicación en sí.
2. Ciclos de desarrollo más rápidos: Spring Boot proporciona muchas funciones potentes y reduce significativamente el tiempo de desarrollo. Esto permite crear y ejecutar rápidamente las aplicaciones.
3. Aplicaciones cloud-ready: Spring Boot facilita la creación de aplicaciones cloud-ready y ready-to-deploy para implementarse en plataformas como AWS y Azure.
4. Reutilización: las aplicaciones Spring Boot se pueden reutilizar y extender para crear otras aplicaciones. Esto permite desarrollar rápidamente aplicaciones existentes y crear otras nuevas.
5. Seguridad: Spring Boot proporciona un entorno seguro, lo que facilita garantizar que

## 7. ¿Cómo testearías un controlador de una API? \*

Es un proceso relativamente sencillo. Se puede usar el framework JUnit estándar para definir los casos de prueba y los métodos de prueba, que se pueden escribir para probar los métodos en sus controladores.

Primero, deberá importar las bibliotecas de Java necesarias para crear sus pruebas. Estos incluyen el framework JUnit, las clases de controlador relevantes, cualquier clase de Java relevante y cualquier otra dependencia de su controlador.

Una vez que haya importado todas las bibliotecas necesarias, puede comenzar a crear sus casos de prueba. Cada caso de prueba debe incluir afirmaciones que evalúen el comportamiento esperado del controlador, como si produce la salida esperada dadas ciertas entradas. También puede agregar aserciones que verifiquen cualquier excepción que pueda generarse desde el controlador.

Una vez que haya creado sus casos de prueba, deberá ejecutarlos para verificar que el controlador tenga el comportamiento esperado. El framework JUnit proporciona un

## 8. ¿Qué tipo de test harías para un servicio y por qué? \*

Para probar la funcionalidad de un service de Spring, hay que realizar unit test, integration test y test del sistema. Las pruebas unitarias se utilizan para verificar la funcionalidad de componentes individuales del código, como métodos y clases. También aseguran que el código sea fácilmente mantenible y extensible en el futuro. Las pruebas de integración verifican la integración de los componentes individuales entre sí y

## 9. ¿Cómo harías para mockear una clase en caso de que necesitaras que ciertos métodos se comportaran como una instancia real?

El mockeo de una clase JAVA se puede realizar mediante varios de framework, como Mockito, JMock, etc. Dependiendo de la complejidad de los requisitos del mockeo, algunos de estos frameworks pueden ofrecer soluciones más efectivas que otros.

El proceso general para simular una clase JAVA implica crear un objeto mockeado para la clase. Este objeto es una representación de la clase y proporciona control sobre los métodos que se van a mockear. Antes de crear el objeto mockeado, se debe decidir qué métodos mockear y cómo configurarlos.

Una vez que se crea el objeto mockeado, se usa para crear un código auxiliar. Un stub es un método en el que el código se escribe para proporcionar un determinado resultado o comportamiento. Cuando se invoca el código real, el método stub ejecuta el código, mockeando así del método.

Finalmente, el objeto mockeado se verifica y prueba. Esto se hace para asegurarse de

## 10. ¿Cuándo utilizarías sistemas de mensajería? Ejemplo: RabbitMQ

Los sistemas de mensajería como RabbitMQ se usan comúnmente para la comunicación asíncrona entre aplicaciones o microservicios. Permiten el desacoplamiento de servicios, lo que ayuda a que los sistemas sean más tolerantes a fallas, resistentes y escalables. RabbitMQ es un popular agente de mensajes de código abierto que se puede utilizar para una amplia gama de aplicaciones, desde la ejecución de canalizaciones de datos en tiempo real hasta la potenciación de backends móviles. Se usa ampliamente en sistemas distribuidos y ayuda a reducir la complejidad de la comunicación entre servicios.

## 11. ¿Qué ventajas tiene utilizar cachés distribuidas?

Los cachés distribuidos ofrecen muchas ventajas en comparación con las técnicas de almacenamiento en caché tradicionales. En primer lugar, permiten un acceso más rápido y fiable a los datos, ya que pueden distribuirse entre varios servidores o centros de datos. Esto significa que si un servidor falla, aún se puede acceder a los datos desde otras ubicaciones. Además, los cachés distribuidos brindan una mejor escalabilidad y flexibilidad, ya que pueden almacenar una mayor cantidad de datos y se pueden usar en diferentes configuraciones. También ofrecen una mejor tolerancia a fallas, ya que la falla de un solo servidor no afecta a todo el sistema. Finalmente, los cachés distribuidos son

## 12. ¿Quién crees que debería participar en las Pull Requests y por qué?

Los pull request son una excelente manera para que los desarrolladores colaboren en un proyecto, ya que les permite sugerir cambios en el código sin tener que realizar cambios directamente. Todos los involucrados en el proyecto deben tener la oportunidad de participar en las solicitudes de extracción, ya que les permite contribuir al desarrollo general del proyecto. También les permite obtener una mejor comprensión de cómo funciona el proyecto y comprender mejor cómo sus propias contribuciones encajan en el todo. Además, permite a los desarrolladores hacer preguntas y brindar comentarios, lo que puede ayudar a que el proyecto avance en una mejor dirección. En última instancia, todos deberían tener las mismas oportunidades de participar en las solicitudes de

## 13. Valora tu experiencia en los siguientes puntos \*

	Nada	Baja	Normal	Alta	Muy alta
Java	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Spring Boot	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Spring Cloud	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Microservicio s	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Maven	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Git	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
JUnit	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mockito	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tests Spring	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
RestAssured	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cucumber	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Docker	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kubernetes	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kafka	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Submit

Never give out your password. [Report abuse](#)

This content is created by the owner of the form. The data you submit will be sent to the form owner. Microsoft is not responsible for the privacy or security practices of its customers, including those of this form owner. Never give out your password.

Powered by Microsoft Forms |

The owner of this form has not provided a privacy statement as to how they will use your response data. Do not provide personal or sensitive information.

| [Terms of use](#)