

Part 2: Theoretical Questions

Java Spring Framework

a. Explain the concept of Dependency Injection (DI) and its importance in Spring.

Inyección de Dependencias (DI): Se trata de un patrón de diseño que promueve el desacoplamiento de los componentes al suministrar o inyectar las dependencias desde el exterior. Esta práctica mejora la modularidad, testabilidad y mantenimiento del código al evitar la creación de dependencias internas en los componentes.

b. What is the difference between Spring MVC and Spring Boot? When would you choose one over the other?

Spring MVC: Es un marco de trabajo para la construcción de aplicaciones web que implementa el patrón de diseño MVC. Los controladores manejan las solicitudes, los modelos representan los datos y las vistas renderizan la interfaz de usuario.

Spring Boot: Es un marco de trabajo construido sobre Spring que simplifica la configuración y puesta en marcha de aplicaciones Spring, agrupando las soluciones en paquetes.

No son mutuamente excluyentes; la mejor práctica es usarlos juntos. Utilizar Spring MVC para desarrollar aplicaciones web escalables y Spring Boot para la configuración de la aplicación.

Databases: MongoDB and PostgreSQL

a. Describe the main differences between MongoDB and PostgreSQL in terms of data model, querying, and scalability.

MongoDB es una base de datos NoSQL que ofrece flexibilidad y escalabilidad para aplicaciones que manejan grandes volúmenes de datos no estructurados, diseñada para escalar horizontalmente.

PostgreSQL es una base de datos relacional con un modelo de datos estructurado que admite transacciones y relaciones entre datos. Ofrece un alto grado de integridad y consistencia de datos.

b. When would you choose MongoDB over PostgreSQL, and vice versa, for a project?

La elección depende de los requisitos específicos de la aplicación. Explotar las características particulares de cada uno:

MongoDB para aplicaciones que requieren estructuras dinámicas y escalables.

PostgreSQL para aplicaciones que requieren transacciones, donde es crucial la integridad de los datos y relaciones complejas.

AWS Infrastructure

a. Describe the difference between EC2 and ECS in AWS.

EC2 es un servicio de AWS que permite a los usuarios gestionar instancias EC2 para ejecutar aplicaciones en la nube con control total sobre la configuración y el mantenimiento de los servicios implementados en las instancias.

ECS es un servicio de AWS diseñado para ejecutar, escalar y administrar contenedores Docker en la nube. Permite desplegar y gestionar aplicaciones utilizando Docker, proporcionando capacidades de orquestación de contenedores, gestión de clústeres y escalabilidad automática.

b. How would you implement high availability and fault tolerance in an AWS environment?

AWS ofrece diferentes servicios para implementar alta disponibilidad:

Distribución geográfica: Proporciona redundancia y protección contra desastres naturales.

Elastic Load Balancing (ELB): Distribuye el tráfico entre instancias EC2 o contenedores ECS.

Auto Scaling: Ajusta automáticamente el número de instancias EC2 o contenedores ECS según la demanda.

Replicación de datos: Almacena copias de seguridad y datos replicados en múltiples regiones de AWS.

Monitoreo y alertas: Implementa sistemas de monitoreo para supervisar la salud y el rendimiento de los recursos en AWS.

2 Microservices and Kubernetes

a. What are microservices, and what are the benefits of using a microservices architecture?

Los microservicios son una arquitectura de diseño de software que estructura una aplicación como una colección de servicios pequeños, independientes y autónomos. Cada servicio de microservicios está diseñado para cumplir con una única responsabilidad de negocio y se puede desarrollar, desplegar, escalar y actualizar de forma independiente. Los beneficios incluyen:

Desacoplamiento y modularidad.

Escalabilidad y flexibilidad.

Despliegue continuo.

Resistencia a fallos.

Tecnología y herramientas especializadas.

Reutilización y mantenibilidad.