

1. ¿Qué ventajas tiene hacer este programa en un modelo de arquitectura de n capas?

Ventajas del modelo en n capas:

- **Separación de responsabilidades:** Organizar el proyecto en capas facilita que cada capa tenga una responsabilidad clara. La capa de presentación se encarga de la interfaz de usuario, la capa de lógica de negocio se ocupa de las reglas y procesos, y la capa de datos maneja la persistencia.
- **Escalabilidad y mantenibilidad:** Separar la lógica en capas hace que sea más fácil escalar el sistema y realizar cambios sin afectar otras partes.
- **Reutilización de código:** La separación de capas permite que varias funcionalidades puedan compartir y reutilizar el código de cada capa.
- **Facilidad de pruebas:** Cada capa se puede probar por separado, facilitando el desarrollo de pruebas unitarias y de integración.

2. ¿Qué desventajas tiene hacer este programa en un modelo de microservicios?

Desventajas de usar microservicios:

- **Complejidad de administración:** Un modelo de microservicios implica manejar múltiples servicios independientes, lo cual puede ser complicado y requiere experiencia en orquestación y monitoreo.
- **Sobrecarga de comunicación:** En una arquitectura de microservicios, los servicios deben comunicarse entre sí a través de APIs, lo cual puede introducir latencia y requerir protocolos de comunicación adicionales.
- **Costos adicionales:** La infraestructura necesaria para gestionar microservicios puede ser más costosa, y podría ser innecesaria para aplicaciones pequeñas.
- **Desafíos de consistencia:** Mantener la consistencia de los datos en múltiples servicios puede ser un desafío, ya que cada servicio puede tener su propio esquema de base de datos.

3. ¿Si tuviera que implementar un sistema de autenticación y autorización como lo haría?

Para implementar autenticación y autorización en esta aplicación:

- **Autenticación:** Utilizaría un sistema basado en **tokens JWT (JSON Web Token)**, generando un token de autenticación al inicio de sesión que el cliente debe incluir en cada solicitud posterior.
- **Autorización:** Definiría roles (por ejemplo, administrador, profesor, estudiante) y permisos basados en roles. Spring Security sería ideal para aplicar esta autorización, permitiendo configurar quién puede acceder a cada endpoint en función de su rol.
- **Seguridad adicional:** Implementaría HTTPS para asegurar la comunicación entre cliente y servidor y establecería reglas de caducidad para los tokens.

4. ¿En este proyecto qué tan importante es la Confidencialidad y la Integridad? ¿Por qué?

- **Confidencialidad:** Es crucial, ya que se manejan datos sensibles como los nombres de estudiantes y sus asignaturas. Sin una adecuada confidencialidad, estos datos podrían ser accesibles a personas no autorizadas.
- **Integridad:** Es igualmente importante, ya que asegura que los datos en la base de datos (como los cursos, asignaturas y estudiantes) no sean modificados o corrompidos de forma accidental o malintencionada.

5. ¿Cómo calcularía la disponibilidad de la implementación que usted realizó?

Para calcular la disponibilidad del sistema:

- **Monitoreo:** Usaría herramientas de monitoreo como Prometheus o Grafana para observar métricas de rendimiento y tiempo de actividad.
- **Pruebas de carga:** Realizaría pruebas de estrés para ver cómo responde el sistema en situaciones de alta carga, y establecería planes de recuperación en caso de caída.

Cálculo de disponibilidad: Aplicaría la fórmula de disponibilidad:

$$\text{Disponibilidad} = \left(\frac{\text{Tiempo total} - \text{Tiempo de inactividad}}{\text{Tiempo total}} \right) \times 100$$

Esto permite obtener un porcentaje de disponibilidad basado en el tiempo de actividad y el tiempo total.

6. Si tuviera 3 desarrolladores para desarrollar esta aplicación, ¿Cómo les distribuiría las tareas?

- **Desarrollador 1:** Responsabilidades en el **backend** – configuración de base de datos, implementación de servicios y controladores, así como la lógica de negocio.
- **Desarrollador 2: Frontend en Angular** – creación de componentes para la interfaz de usuario, integración con el backend, y diseño de las pantallas necesarias para la aplicación.
- **Desarrollador 3: Pruebas y Seguridad** – desarrollo de pruebas unitarias y de integración, configurando también la autenticación y autorización en el sistema.

7. Si hay que generar un reporte, ¿Cómo lo haría?

- Usaría **JasperReports o Apache POI** para generar reportes en formato PDF o Excel, que permita exportar información sobre cursos, asignaturas y estudiantes.
- **Interfaz de usuario:** Incluiría una opción en el frontend que permita al usuario seleccionar los datos que desea incluir en el reporte y, al generarlo, descargue el archivo correspondiente.
- **Backend:** Configuraría un endpoint que genere el reporte y lo devuelva como respuesta.