**Preguntas**:

1. **¿Qué entiendes por tecnología JEE y cuáles son algunos de sus componentes más importantes?**

Java Platform Enterprise Edition (JEE) se refiere al conjunto de tecnologías y especificaciones que brinda Oracle para la creación y administración de aplicaciones web empresariales dinámicas que sean robustas, escalables, distribuidas y seguras. Dentro de los componentes tenemos todos los extendidos desde JSE, los que facilitan la creación de servicios, aplicaciones, sistemas, etc.

JEE incluye bibliotecas y APIs para tareas como:

* Comunicación en red (RMI, JAX-RS, JAX-WS)
  + **JAX-RS**: servicios REST.
* Acceso a bases de datos (JPA, JDBC)
  + **JPA** (Jakarta Persistence): mapeo objeto–relacional para trabajar con BD desde entidades (por ej., Hibernate como implementación).
* Desarrollo de interfaces web (Servlets, JSP, JSF)
  + **Servlets**: clases Java que atienden peticiones HTTP (controladores). Base de todo web en Java.
  + **JSP + JSTL + EL**: páginas que generan HTML en el servidor; JSTL/EL ayudan a escribir menos Java en la vista.
* Manejo de seguridad, transacciones y concurrencia
* Gestión de componentes distribuidos (EJB, CDI)

1. **¿Cuál es la diferencia entre una Java Server Page (JSP) y un Servlet?**

El JSP sería la vista, donde se escribe lo necesario para la UI, donde se produce el html que interpreta el código de java del Servlet.

El Servlet es la lógica donde se procesan las solicitudes y respuestas http, lee los parámetros que se pasan/solicitan en las vistas y llaman servicios dependiendo de las interacciones, mostrando ciertas vistas o otras.

1. **Explica con tus propias palabras qué es un servidor de aplicaciones y menciona al menos dos ejemplos conocidos.**

Se refiere al software que ejecuta las aplicaciones y proporciona lo necesario para una infraestructura robustas, escalables, distribuidas y seguras. Según entiendo los servidores pueden dividir los proyectos en capas como el MVC lo que permite ordenar o modularizar, favoreciendo el orden/organización y mantenimiento del proyecto.

* Apache Tomcat: contenedor de Servlets/JSP muy popular (no es JEE completo, pero es el más usado para web clásica y también lo usa Spring Boot por dentro).
* WildFly (JBoss) o GlassFish/Payara: servidores de aplicaciones Jakarta EE completos (incluyen JPA, CDI, JAX‑RS, etc.).

1. **¿Qué diferencia hay entre el desarrollo front-end y el back-end? Proporciona al menos un ejemplo para cada uno.**

* Front‑end: todo lo que el usuario ve y toca en el navegador. Tecnologías: HTML, CSS, JavaScript, frameworks como Bootstrap, Tailwind, React, Vue.
  + Ejemplo: una página de catálogo con tarjetas de productos, estilos de Bootstrap y un pequeño JS para filtrar por nombre.
* Back‑end: la lógica del servidor: controladores, reglas de negocio, acceso a BD, APIs. Tecnologías: Java (Servlets, Spring MVC/Spring Boot), SQL, JPA/Hibernate.
  + Ejemplo: un endpoint /api/productos en Spring Boot que devuelve JSON desde MySQL usando JPA.

1. **Imagina que creas un proyecto web dinámico. Enumera y explica los pasos necesarios desde su creación hasta su ejecución local en el navegador usando Spring Tool Suite.**
2. Pre‑requisitos: Tener JDK instalado (Configura en STS: Window → Preferences → Java → Installed JREs).
   * Opción: descargar Apache Tomcat (v9 o v10) y tomar nota de su carpeta.
3. Agregar Tomcat a STS: Window → Preferences → Servers → Runtime Enviroments → botón Add → Apache → Tomcat vX
   * Opción a) selecciona la carpeta de Tomcat descargado anteriormente
   * Opción b) descargar e instalar directamente → aceptar los terminos.
4. Crear el proyecto: File → New → Dynamic Web Project.
5. Asignar nombre y Target Runtime = Tomcat (versión que queramos) → versión de module, configuración de apache tomcat
6. Estructura mínima: carpeta WebContent/ o src/main/webapp/ con index.jsp; si creas un Servlet: File → New → Servlet.
7. Ejecutar: clic derecho al proyecto → Run As → Run on Server → elige Tomcat → Finish. STS publicará la app y abrirá el navegador en [http://localhost:8080/<contexto](http://localhost:8080/%3ccontexto)> donde contexto sería el nombre del Servlet.
8. **¿Por qué es importante configurar correctamente el servidor dentro del entorno de desarrollo antes de ejecutar un proyecto?**

Es de suma importancia puesto que de no tener las configuraciones correctas presentaremos diversos errores. Hay que verificar compatibilidad, las librerías utilizadas, las variables de entorno, las configuraciones respecto de la publicación automática, etc.