

RESPUESTAS A LAS PREGUNTAS DEL PUNTO 4 TALLER 2

1. ¿Qué es un Enterprise bean?

R/: Es un componente del lado servidor que encapsula la lógica de negocio de una aplicación.

2. ¿Qué es un session bean?

R/: Es aquel que encapsula la lógica de negocio que puede ser invocado programáticamente por un cliente sobre local, remoto o sobre vista cliente de servicio web.

3. ¿Cuáles son las ventajas de un bean?

R/: El principal beneficio de un bean es que simplifica el desarrollo de aplicaciones a gran escala y distribuidos porque:

- El contenedor EJB (Enterprise JavaBeans) proporciona servicios a nivel de sistema al bean empresarial. Así, el desarrollador del bean puede concentrarse en resolver los problemas de negocio.
- El contenedor EJB, en lugar del desarrollador del bean, es responsable de los servicios a nivel del sistema, tales como administración de transacciones y autenticación de seguridad.
- Los beans, en lugar de los clientes, contienen la lógica de negocio de la aplicación, así, el desarrollador del cliente puede enfocarse en la presentación del cliente. El desarrollador del cliente no tiene que codificar las rutinas que implementan reglas de negocio o acceda a las bases de datos. Como resultado, los clientes son más ligeros, un beneficio que es particularmente importante para clientes que corran en dispositivos pequeños.
- Los beans empresariales son componentes portables, el ensamblador de la aplicación puede construir nuevas aplicaciones de beans existentes.

4. ¿Cuáles son los tipos de session beans y en que consiste cada uno?

R/: Tipos de sesión beans:

Session bean con estado: Consiste en que las variables de la instancia representan el estado de una única sesión cliente/bean. Debido a que la capa cliente interactúa con su bean, este estado ocasionalmente se llama, estado conversacional. El estado del bean se retiene durante tal sesión. Si el cliente elimina su bean, la sesión termina y su estado desaparece.

Session bean sin estado: No mantiene un estado conversacional con el cliente. Cuando un cliente invoca los métodos de un bean sin estado, las variables de instancia del bean pueden contener un estado específico para el cliente, pero sólo lo retiene durante su invocación. Cuando el método se termina, el estado de cliente específico no debe ser retenido. Esto permite a los

clientes cambiar el estado de variables de instancia dentro de un conjunto de beans de sesión sin estado y posponerlo a la próxima invocación de un bean de sesión sin estado del conjunto. Así, el estado de un bean de sesión sin estado se aplica a través de todos los clientes.

Session bean singleton: Ofrece similar funcionalidad respecto a un bean de sesión sin estado, con la diferencia de que hay un bean de sesión singleton por aplicación, opuesto a un conjunto de beans de sesión sin estado, en el que cualquier puede responder a un requerimiento del cliente.

5. ¿Cuándo usar un session bean?

Stateful: Se usa cuando:

- Cuando el estado del bean representa la interacción entre el bean y un cliente específico.
- Cuando el bean necesita mantener la información de los clientes a través de invocaciones a métodos.
- Cuando el bean sea el intermedio entre el cliente y los otros componentes de la aplicación presentándole una vista simplificada.
- Cuando en detrás de escenas, el bean administra el flujo de trabajo de diversos beans empresariales.

Stateless: Se usa cuando:

- Cuando el estado del bean no tenga datos para un cliente en específico.
- Cuando en una invocación a un simple método, el bean realiza una tarea genérica para todos los clientes.
- Cuando el bean implemente un servicio web.

Singleton: Se usa cuando:

- Cuando el estado necesite ser compartido a través de la aplicación.
- Cuando un bean empresarial simple necesite ser accedido por múltiple hilos de manera concurrente.
- Cuando la aplicación necesite un bean empresarial para realizar tareas al inicio o al cierre de la aplicación.

6. ¿Qué es JPQL?

R/: Es un lenguaje de consulta que es usado para crear consultas para entidades a almacenar en una base de datos. Es desarrollado basado en la sintaxis de lenguaje SQL, pero no afecta directamente a la base de datos.

7. ¿Cuál es la diferencia entre JPQL y SQL?

R/: La principal diferencia entre JPQL y SQL es que con SQL trabaja directamente para tablas de bases de datos relacionales, registros y campos, mientras que con JPQL trabaja con clases e instancias.