

**LECTURA 3: ENTERPRISE BEANS. CUADRO RESUMEN**

¿Qué es un Bean empresarial?	Componente del lado servidor que encapsula la lógica de negocio de una aplicación.	Beneficio	Simplifica el desarrollo de aplicaciones a gran escala y distribuidos.
		¿Cuándo se usa?	Cuando la aplicación pueda ser escalable. Cuando las transacciones pueden asegurar la integridad de datos. Cuando la aplicación vaya a tener una variedad de clientes.
		Tipos	Bean de sesión Bean de mensaje conducido
¿Qué es un Bean de sesión?	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.	Tipos	Con estado. Sin estado. Singleton.
		¿Cuándo se usa un bean de sesión con estado?	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.
		¿Cuándo se usa un bean de sesión sin estado?	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.
		¿Cuándo se usa un bean de sesión Singleton?	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. Cuando el bean implemente un servicio web.

¿Qué es un Bean de mensaje conducido?	Es un bean empresarial que permite a las aplicaciones Java procesar mensajes asincrónamente.	Diferencia con respecto a los beans de sesión	Los clientes no acceden a mensajes a través de interfaces como se hace mediante los beans de sesión. Además, un bean de mensaje conducido sólo tiene una clase bean.
		Características	Ejecutan sobre la recepción de mensaje de cliente simple. Se invocan asincrónamente. Son relativamente efímeros. No representan directamente datos compartidos en una base de datos, pero pueden tener acceso y editar estos datos. Pueden ser de transacción a consciencia. Son beans sin estado.
Acceso a los beans empresariales	Los clientes pueden acceder a los beans empresariales ya sea mediante vistas sin interfaz o mediante interfaces de negocio. Accediendo mediante lo primero puede invocar cualquier método público de la clase de implementación del bean empresarial o de cualquier superclase de la clase de implementación. Accediendo mediante lo segundo se invocan los métodos de negocio de la interfaz de negocio estándar de Java.		
Contenidos de un Bean empresarial.	Clase bean empresarial. Interfaces de negocio. Clases ayudantes tales como excepciones y clases de utilidad.		
Convenciones nombradas para un Bean empresarial.	<b>Item</b>	<b>Sintaxis</b>	<b>Ejemplo</b>
	Nombre de bean empresarial	nameBean	AccountBean
	Clase de bean empresarial	nameBean	AccountBean
	Interfaz de negocio	Name	Account

Los ciclos de vida de un Bean empresarial.	Dependen del tipo de bean empresarial.	Ciclo de vida de un bean con estado	<p>           ① Create            ② Dependency injection, if any            ③ PostConstruct callback, if any            ④ Init method, or ejbCreate&lt;METHOD&gt;, if any         </p> <pre> graph LR     DNE[Does Not Exist] -- "① Create" --&gt; Ready[Ready]     Ready -- "① Remove" --&gt; DNE     Ready -- "PrePassivate callback, if any" --&gt; Passive[Passive]     Passive -- "PostActivate callback, if any" --&gt; Ready           </pre> <p>           ① Remove            ② PreDestroy callback, if any         </p>
		Ciclo de vida de un bean sin estado y Singleton	<p>           ① Dependency injection, if any            ② PostConstruct callback, if any         </p> <pre> graph LR     DNE[Does Not Exist] -- "① Dependency injection, if any ② PostConstruct callback, if any" --&gt; Ready[Ready]     Ready -- "PreDestroy callback, if any" --&gt; DNE           </pre>
		Ciclo de vida de un bean de mensaje conducido	<pre> graph LR     DNE[Does Not Exist] -- "① Dependency injection, if any ② PostConstruct callback, if any" --&gt; Ready[Ready]     Ready -- "onMessage" --&gt; Ready     Ready -- "PreDestroy callback, if any" --&gt; DNE           </pre>