

0^a
Emisión

DIPLOMADO Desarrollo de Sistemas con Tecnología Java

Módulo 8 Enterprise Java Beans

Ing. Jorge Alberto Montalvo Olvera



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Dirección General de Cómputo y de Tecnologías de información y Comunicación

Dirección de Docencia en TIC



**Educación
Continua
1971 - 2021**

TEMARIO

- Introducción a la Tecnología EJB
- Desarrollo de Java Beans
 - Stateless
 - Stateful

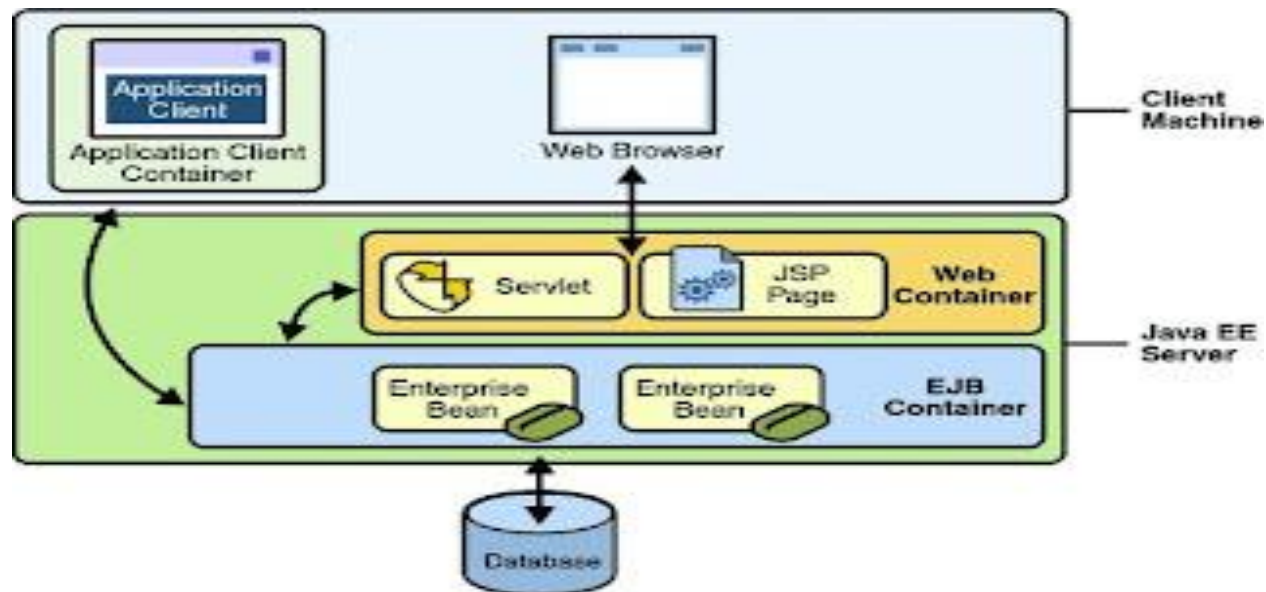
Tecnología EJB

- La tecnología de los Enterprise Java Bean forma parte de la arquitectura EE, un EJB es una porción de código que contiene métodos y campos que implementan módulos de lógica de negocio.
- El servidor que soporta la tecnología EE está integrado por contenedores:
 - Contenedor de aplicaciones cliente
 - Contenedor web
 - Contenedor EJB



Tecnología EJB

- Debemos entender por contenedor el espacio lógico donde se encuentra el software que atenderá las peticiones bajo un protocolo particular, a diferencia del servidor que es el segmento lógico donde se instala y configura el contenedor.



Tecnología EJB

- El contenedor web
 - El contenedor web es la interfaz entre los componentes web y el servidor web. Un componente web puede ser un servlet o una página JavaServer Faces Facelets. El contenedor administra el ciclo de vida del componente, envía solicitudes a los componentes de la aplicación y proporciona interfaces para datos de contexto, como información sobre la solicitud actual.

Tecnología EJB

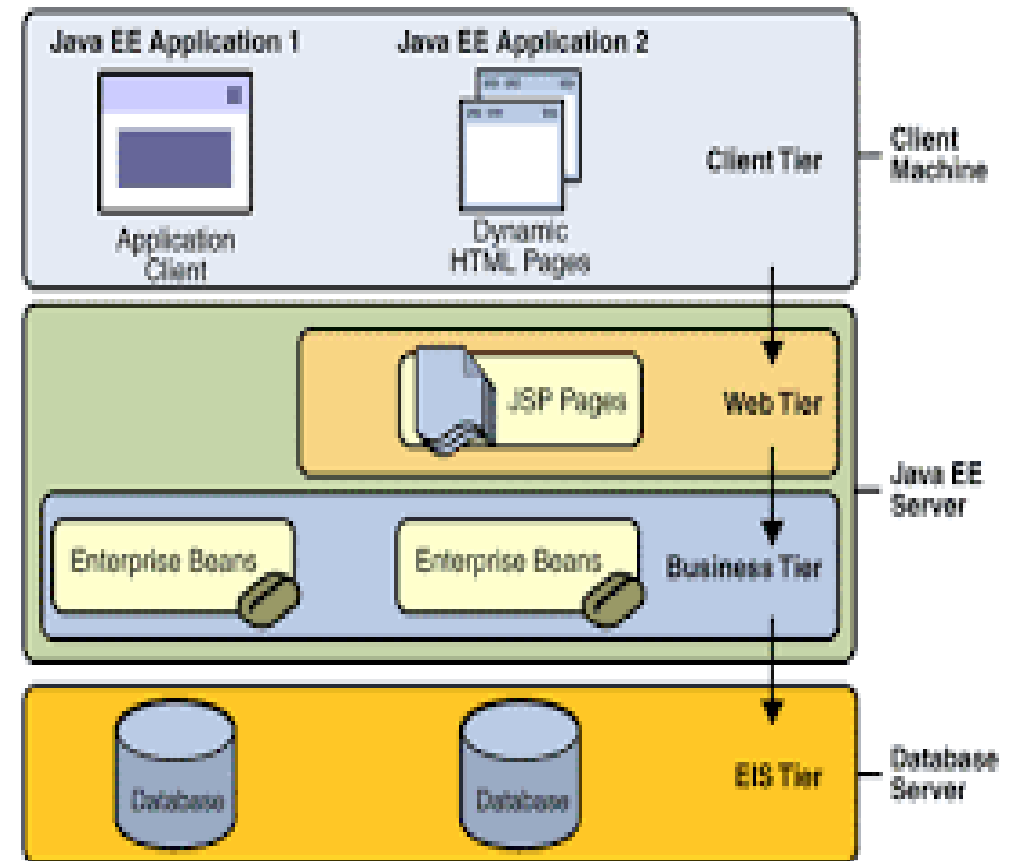
- El contenedor EJB
 - El contenedor EJB es la interfaz entre los beans empresariales, que proporcionan la lógica comercial en una aplicación Jakarta EE y el servidor Jakarta EE. El contenedor EJB se ejecuta en el servidor Jakarta EE y administra la ejecución de los beans empresariales de una aplicación.

Tecnología EJB

- El contenedor del cliente de la aplicación
 - El contenedor del cliente de la aplicación es la interfaz entre los clientes de la aplicación Jakarta EE (aplicaciones Java SE especiales que utilizan componentes del servidor Jakarta EE) y el servidor Jakarta EE. El contenedor del cliente de la aplicación se ejecuta en la máquina del cliente y es la puerta de enlace entre la aplicación del cliente y los componentes del servidor Jakarta EE que utiliza el cliente.

Tecnología EJB

- Los Enterprise Java Beans se encuentran contenidos en la capa de negocio de una arquitectura multicapas.
- El propósito de localizar los objetos de negocio en una capa independiente se debe a que estos podrán ser accedidos desde la capa Web o incluso desde la capa de aplicaciones cliente de escritorio.



Tecnología EJB

- Beneficios de la tecnología EJB:
 - El contenedor EJB provee los servicios de nivel de sistema a los Enterprise Beans, por lo que el desarrollador solo debe concentrarse en resolver los problemas de negocio.
 - Permiten que el desarrollador de la aplicación cliente se concentre en los temas de Vista y control de las vistas, ya que el EJB es quien contiene toda la lógica del negocio que se consume en la aplicación.
 - Los EJB's son componentes portables, las aplicaciones pueden ejecutarse en cualquier servidor EE que pueda proveer las API's estándar descritas en la arquitectura EE.

Tecnología EJB

- ¿Por qué usar EJB?
 - Escalabilidad. Para soportar un número creciente de usuarios necesitamos distribuir los componentes de la aplicación a través de múltiples equipos.
 - Transaccionalidad. Para poder asegurar la integridad de los datos. El contenedor es capaz de manejar la concurrencia de acceso a los objetos compartidos en el servidor.
 - Varios clientes. Se requieren de pocas líneas de código para que clientes remotos puedan localizar los EJB's. Estos clientes pueden ser muchos y variados.

Tecnología EJB

- Características de la arquitectura EE:
 - Ambiente Distribuido
 - Portabilidad
 - Interoperabilidad
 - Escalabilidad
 - Simplicidad
 - Alta disponibilidad y desempeño

Tecnología EJB

- Características de la arquitectura EJB:
 - Transacciones
 - Seguridad
 - Concurrencia
 - Persistencia
 - Manejo de errores
 - Gestión de recursos

Tecnología EJB

- Contenedor Enterprise JavaBeans EJB.
 - Administra la ejecución de Enterprise Java Beans para aplicaciones EE. Los Enterprise Beans y este contenedor corren en el servidor EE.
 - Algunos de los servicios que ofrece un contenedor web son:
 - Java Naming and Directory interface (JNDI)
 - Manejo de Transacciones (JTA)
 - Java mail
 - WebServices
 - Java Persistence

Session Beans

- Session Bean representan una sesión interactiva entre un cliente y el Enterprise Bean.
- Pueden tener un estado o no, pero en caso de tenerlo éste se almacenará en el propio Enterprise Bean no siendo compartido con otro cliente.
- Existen dos modalidades para el manejo del estado de los EJB de sesión:
 - Stateful
 - Stateless

Session Beans

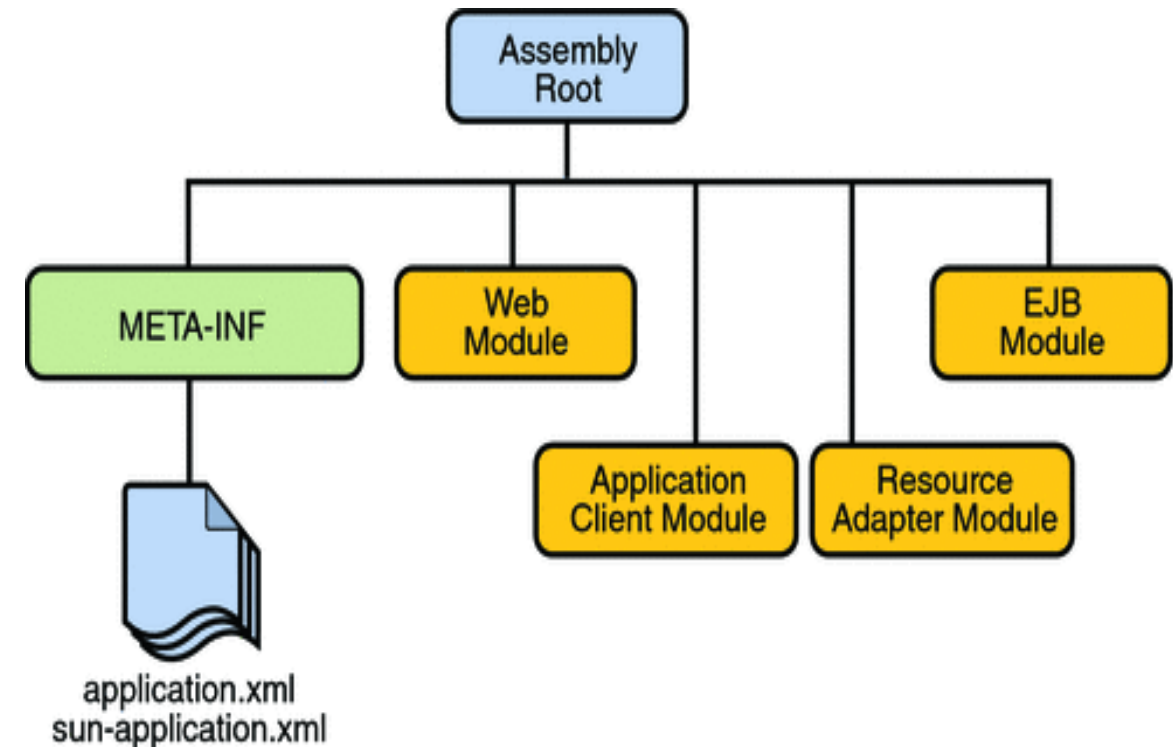
- Sin estado (Stateless).
 - En este modo, no se modifica el estado del Enterprise Bean con las llamadas de los clientes. Los métodos que lo componen tienen como principal objetivo obtener y poner datos a disposición de los clientes.
 - @Stateless

Session Beans

- Con estado (@Stateful).
- Sus variables almacenan datos de la interacción con el cliente, este estado se va modificando en función del uso que hace el cliente de los métodos del Enterprise Bean
- Estos EJB se deben usar si se cumplen con lo siguiente:
 - Se requiere representar la interacción entre el EJB y el cliente que lo consume.
 - El EJB necesita almacenar información del cliente durante la interacción.
 - Se requiere que el EJB funcione como una vista simplificada del cliente para el resto de componentes de la aplicación.

Estructura EAR

- Un archivo de almacenamiento de Java Enterprise se almacena en el formato EAR y se utiliza como un archivo ejecutable de un programa desarrollado en Java, en particular la Enterprise Edition.
- Este tipo de ficheros son desplegados en servidores de aplicaciones que soporten el stack completo de EE.



Estructura EAR

- Aparte de poder contener varias aplicaciones web también tiene la capacidad de contener EJB que son empaquetados en archivos JAR para su utilización en las distintas aplicaciones web.
- Los EAR disponen del fichero application.xml (directorio META-INF) que define la estructura de módulos incluidos en este tipo de empaquetamiento .

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <display-name>testear</display-name>
  <module>
    <web>
      <web-uri>testwebapp.war</web-uri>
      <context-root>testwebapp</context-root>
    </web>
  </module>
</application>
```

EJB

- Para la construcción del Session Bean es importante la definición de los métodos, es decir los métodos que estarán expuestos para los clientes que requieran consumir una función de negocio.
- Se debe definir qué tipo de cliente es el que realizará el llamado a los métodos del EJB, si se trata de un cliente local o remoto.

EJB

- Entiéndase por cliente local a aquellos que se ejecutan bajo la misma JVM en la que se ejecuta el servidor de aplicaciones y por cliente remoto a los que se ejecutan en una JVM distinta.
- Esto quiere decir que los Session Bean están integrados por una interface local o remota o ambas, una clase que implementa la(s) interface(s) y las clases soporte para interactuar con el negocio.

EJB

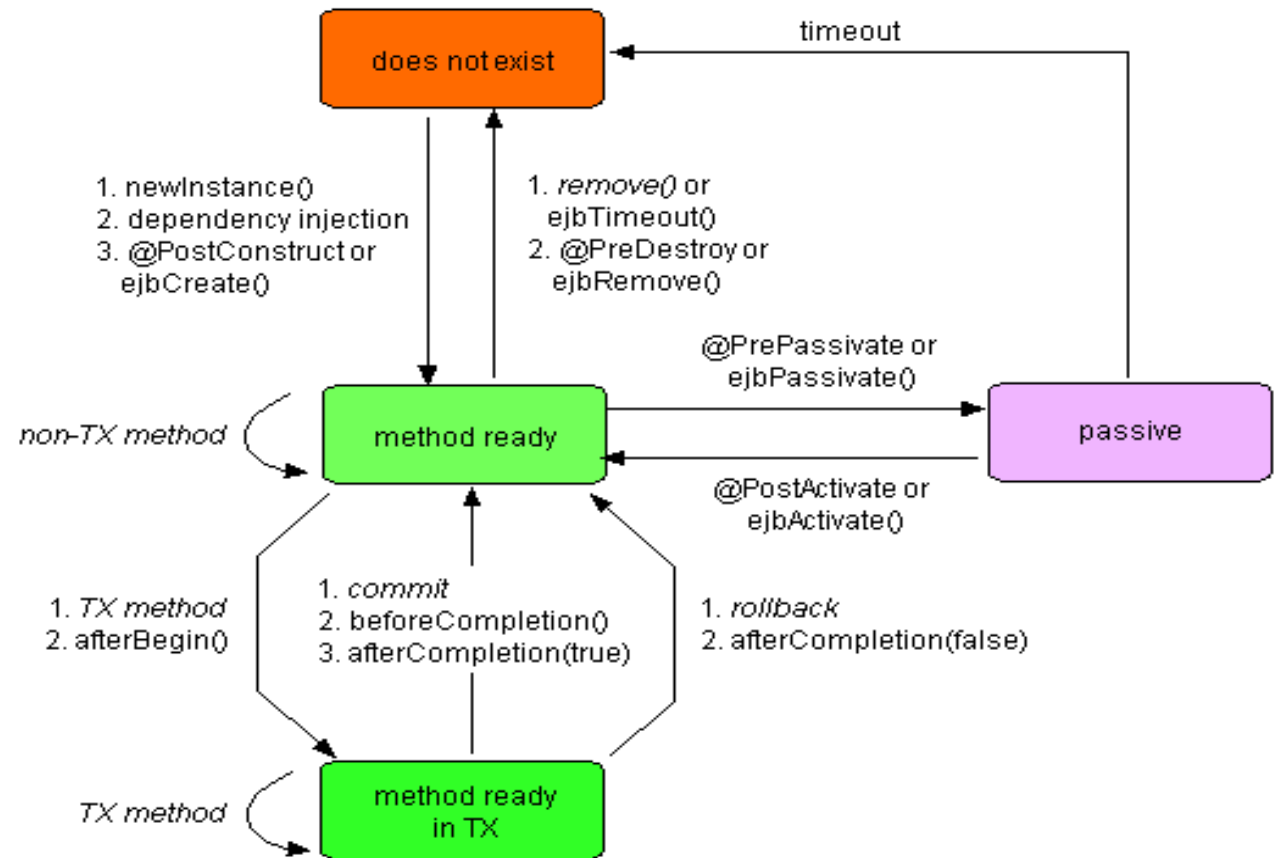
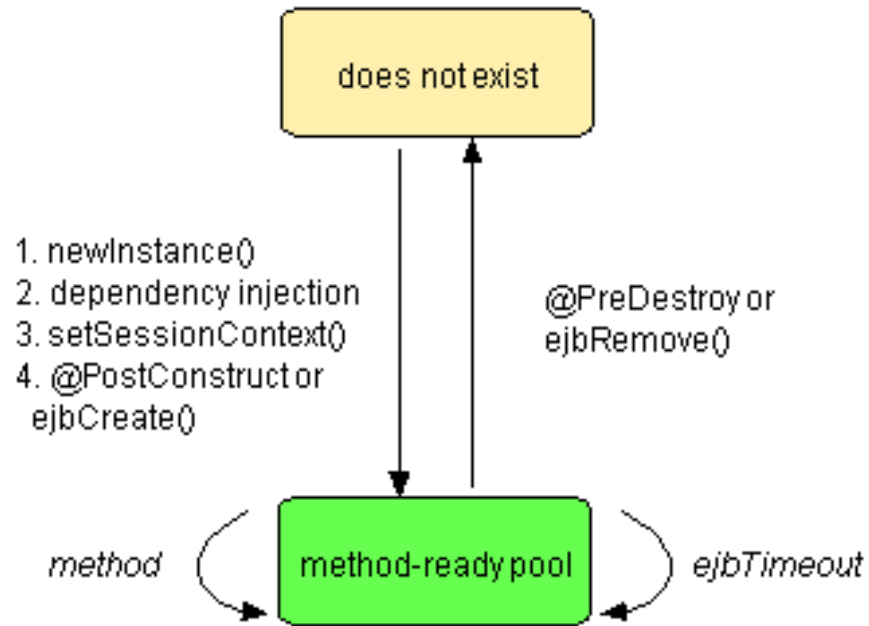
- Interface Remota. Esta interface debe de ser anotado como `@Remote (jakarta.ejb.Remote)` y solo debe de contener la definición de los métodos que serán expuesto a los clientes ubicados fuera del entorno de ejecución del servidor de aplicaciones.

EJB

- Interface Local. Aquí debemos de colocar los métodos que estarán disponibles para los clientes locales, es decir aquellos que se ejecutan en el mismo entorno que el servidor de aplicaciones donde se localiza el contenedor de EJB. La forma en que debemos de anotar esta interface es `@Local (jakarta.ejb.Local)`.

EJB

- Ciclo de vida de un EJB



EJB

- PostConstruct
 - Ocurre antes de la primera invocación de método en la instancia de bean. Esto es en un punto después del cual el contenedor ha realizado cualquier inyección de dependencia.

EJB

- PreDestroy
 - La notificación de PreDestroy indica que el contenedor está eliminando la instancia. En los métodos del ciclo de vida de PreDestroy, la instancia normalmente libera los recursos que ha estado reteniendo.

EJB

- La especificación de la plataforma Jakarta EE define un espacio de nombres JNDI global estandarizado y una serie de espacios de nombres relacionados que se asignan a los distintos ámbitos de una aplicación Jakarta EE. Las aplicaciones pueden utilizar estos espacios de nombres para recuperar de forma portátil referencias a componentes y recursos.
 - `java:global[/<app-name>]/<module-name>/<bean-name>[!<fully-qualified-interface-name>]`

EJB

- **<app-name>**
 - solo se aplica si el bean de sesión está dentro de un archivo .ear. El valor predeterminado es el nombre base del archivo .ear sin extensión de nombre de archivo
- **<nombre-módulo>**
 - es el nombre del módulo en el que está el bean de sesión. En un archivo ejb-jar independiente o un archivo .war, <nombre-módulo> por defecto es el nombre base del módulo con cualquier extensión de nombre de archivo eliminada.
 - En un archivo ear, <nombre-módulo> por defecto es la ruta del módulo con cualquier extensión de nombre de archivo eliminada, pero con cualquier nombre de directorio incluido.
- **<nombre-bean>**
 - es el nombre-ejb del bean empresarial. Para los ejbs definidos a través de anotaciones, el valor predeterminado es el nombre no calificado de la clase de bean de sesión

Contacto

Jorge Alberto Montalvo Olvera
Ingeniero en Computación

jorge.Montalvo@gm3s.com.mx