Java EE

Aplicaciones Web / Sistemas Web

Juan Pavón Mestras
Dep. Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense Madrid

Java EE

- **Java Platform, Enterprise Edition (Java EE)**

- Aplicaciones web construidas a base de **componentes**:
  - Clientes de aplicación y applets (en el cliente)
  - Java Servlet, JavaServer Faces (JSF), and JavaServer Pages (JSP) (en el servidor)
  - Enterprise JavaBeans (EJB) (o *enterprise beans*) (en el servidor)

- Los componentes se despliegan y ejecutan en **contenedores** especializados
  - Ejemplos de contenedores:
    - Contenedor de applets en un navegador Web
    - Contenedor Web Tomcat
    - Contenedor de EJBs

Java EE – Contenedores de componentes

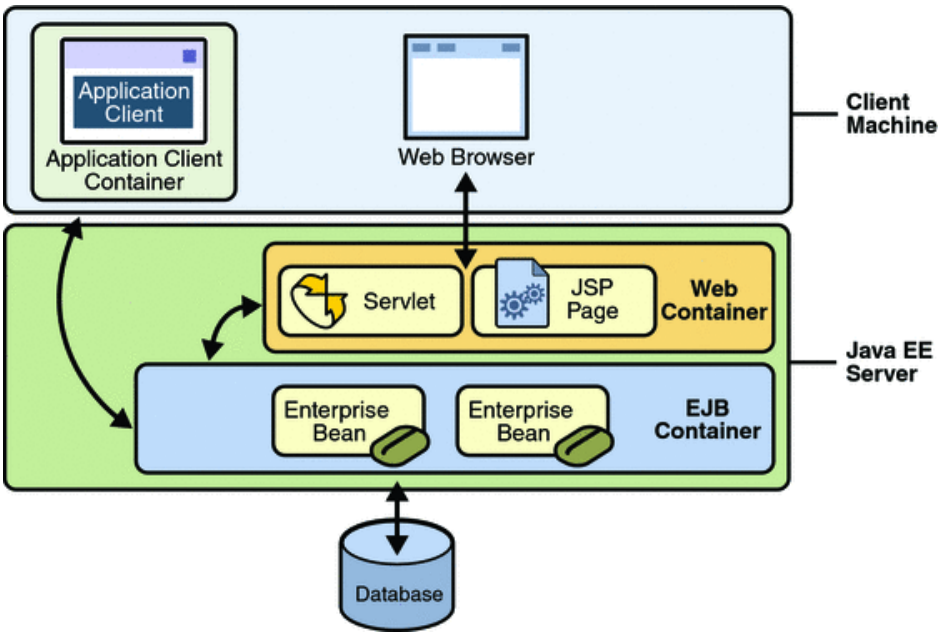


Figura de *The Java EE 6 Tutorial* (2013). Java EE Containers
http://docs.oracle.com/javaee/5/tutorial/doc/bnabo.html

Componentes vs. objetos

- Un componente se caracteriza por:
  - Ser una **unidad de despliegue** independiente
    - Encapsula sus características constituyentes respecto a su entorno
      - Las terceras partes no pueden acceder a los detalles de construcción del componente
    - No se implanta de manera parcial
  - Ser una **unidad de composición**
    - Con componentes posiblemente desarrollados por otros
    - Debe ser suficientemente autocontenido
    - Especificaciones claras de lo que requiere y de lo que proporciona
      - Interacciona con su entorno a través de interfaces bien definidas
  - No tener estado persistente
    - Un componente no se distingue de otras copias del mismo
      - Excepto atributos no funcionales como el número de serie
    - Por tanto, en un proceso se puede decir si hay o no un componente, pero no varias instancias del mismo

Componentes vs. objetos

- Un objeto se caracteriza por:
  - Ser una **unidad de instanciación**; tienen una **identidad única**
    - No se instancia de manera parcial
    - La identidad es única y no cambia durante la vida del objeto
  - Tener un **estado**
    - Se crea con un estado inicial que evoluciona durante la ejecución
  - Encapsular su estado y comportamiento
    - Que está definido bien por una clase o por un objeto prototipo

Interfaces

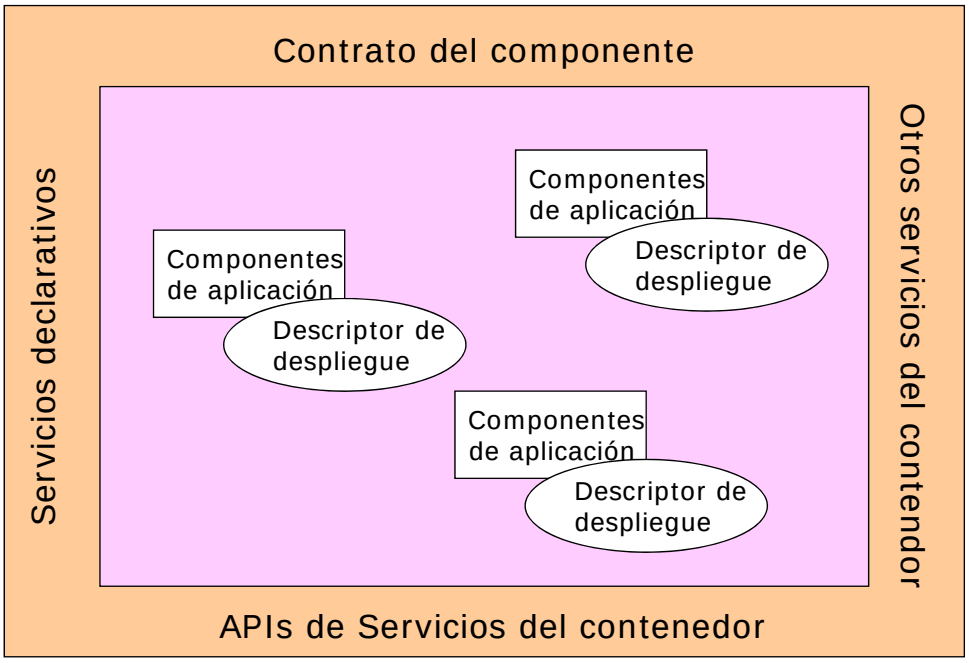
- Puntos de acceso a los componentes
  - Permite a los clientes acceder a los servicios proporcionados por un componente
- Un componente puede tener varias interfaces
  - Una por cada punto de acceso: uso, administración, configuración, ...
  - Pero no conviene tener varias interfaces similares o redundantes
- La especificación de las interfaces es un **contrato**
  - El respeto de este contrato por cliente y componente asegura el éxito de la interacción

Contenedores

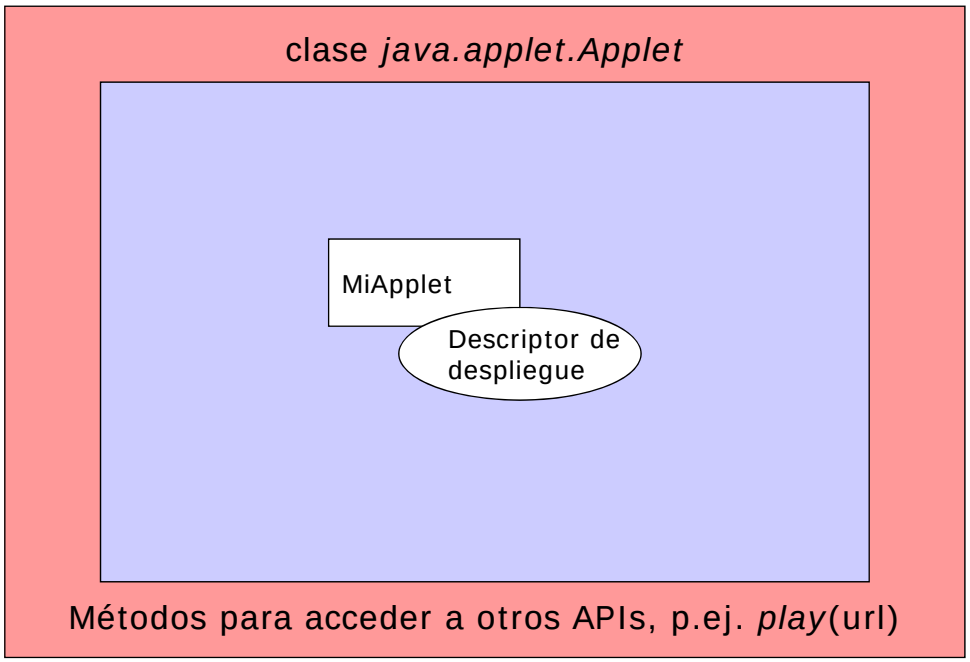
- Un contenedor es un proceso donde se ejecutan los componentes
  - Gestiona los componentes de la aplicación
    - **Ciclo de vida**
  - Proporciona acceso a **servicios** de la plataforma
    - Seguridad, transacciones, persistencia, conectividad, etc.

- El desarrollador tiene que especificar
  - Los componentes de la aplicación
    - Servlets
    - JSPs (JavaServer Pages)
    - JSFs (JavaServer Faces)
    - EJBs (Enterprise Java Beans)
  - Los descriptores de despliegue (*deployment*)
    - Ficheros XML que describen los componentes de aplicación

Arquitectura de un contenedor

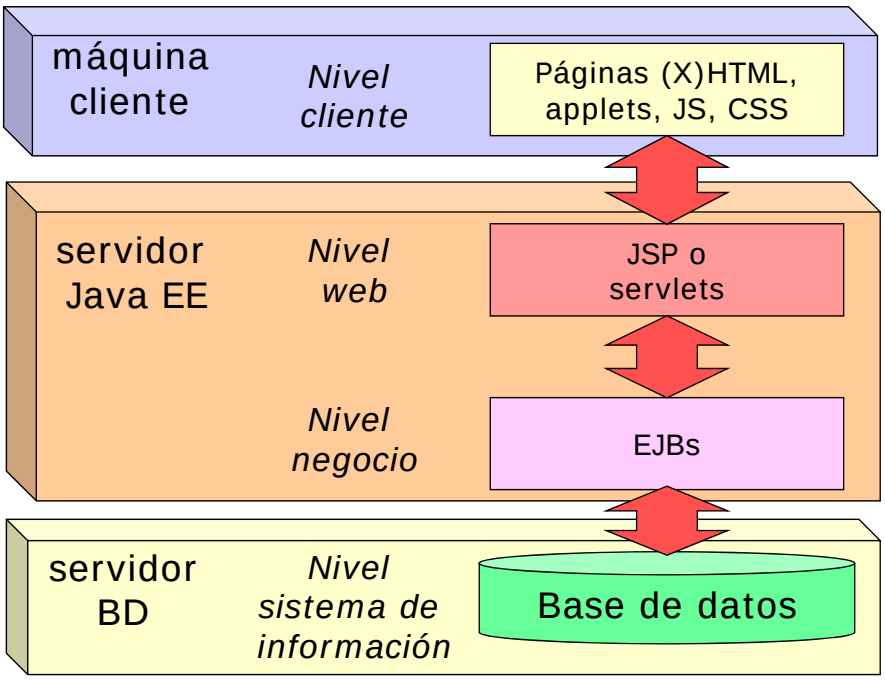


Contenedor de applets



Java EE – Arquitectura multi-nivel (multi-tier)

- Este modelo propicia aplicaciones web en 4 niveles:

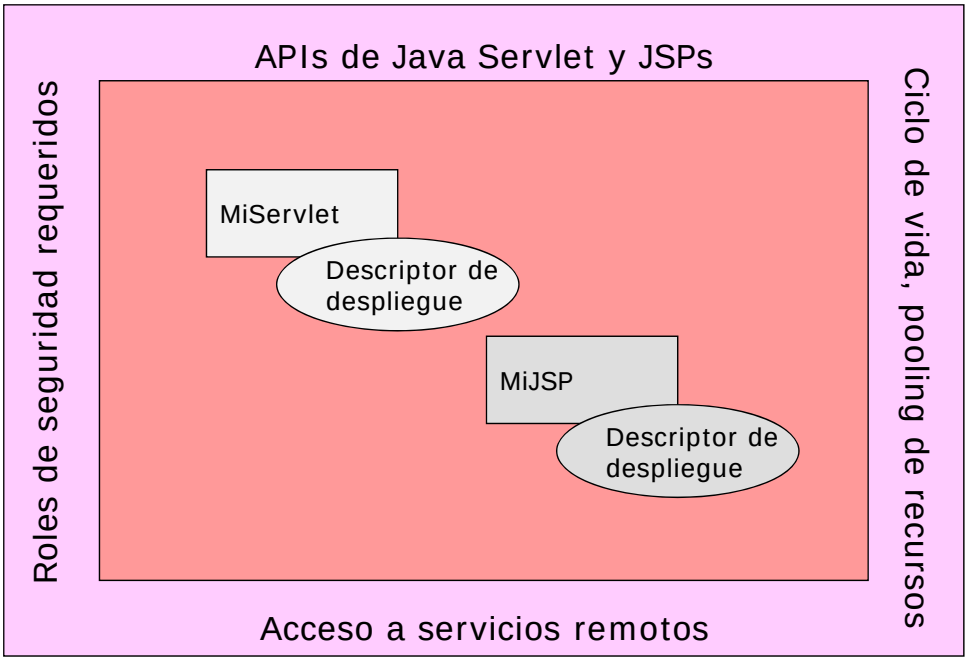


Componentes web en Java EE

- **Java Servlets**
  - Clases escritas en Java que procesan peticiones HTTP y construyen respuestas
- **JavaServer Pages (JSP)**
  - Documentos basados en texto que contienen dos tipos de texto
    - una plantilla de datos estática que puede expresarse en un formato como (X)HTML o XML
    - elementos JSP que determinan cómo la página construye el contenido dinámico
- **JavaServer Faces (JSF)**
  - Componentes de interfaz de usuario para aplicaciones web

- Los **clientes** en Java EE son
  - Clientes web: navegadores web, páginas web, applets
  - Aplicaciones de cliente

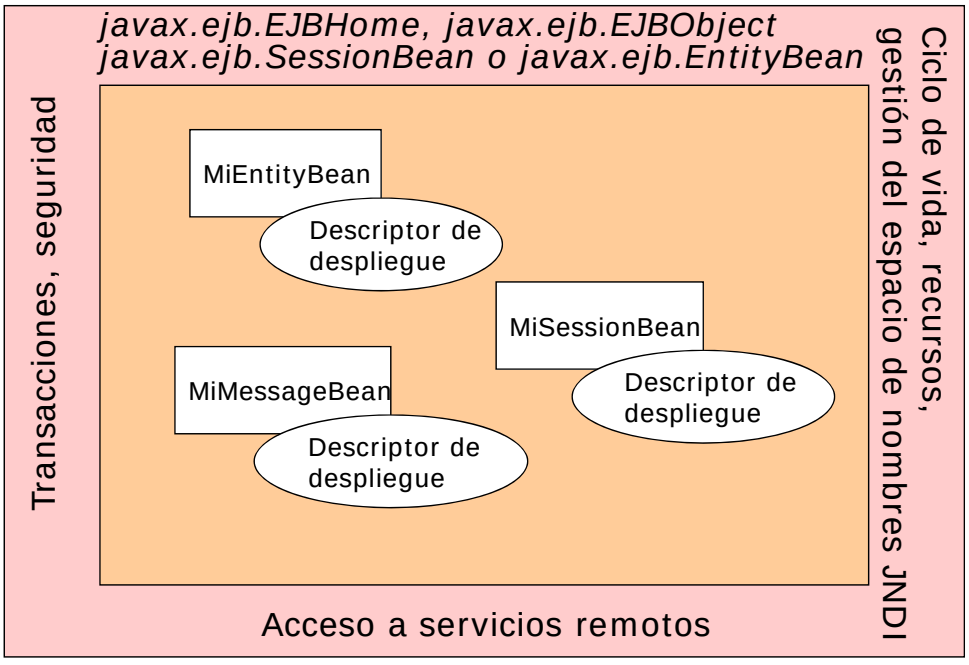
Contenedor Web



Componentes de negocio en Java EE

- Lógica que resuelve las necesidades de un determinado dominio de aplicación
- **Enterprise beans (EJBs)**
  - Pueden procesar datos recibidos del lado cliente y enviarlos al nivel de sistema de información para su almacenamiento
  - Pueden recuperar datos del sistema de información, procesarlos y enviarlos al cliente
- Tipos de EJBs
  - Bean de sesión
    - Una conversación con un cliente
  - Bean dirigido por mensajes
    - Permite que un componente de negocio pueda recibir mensajes asíncronamente, normalmente con el Java Message Service (JMS)

Contenedor de EJBs



Servicios Java EE

- Cada contenedor Java EE proporciona servicios a los componentes
  - Java Naming Direct Interface (JNDI)
  - Java Persistence API (JPA)
  - Java Database Connectivity API (JDBC)
  - Java Transaction API (JTA)
  - Java Message Service (JMS)
  - JavaMail
  - Java Beans Active Framework (JAF)
  - Java EE Connector Arquitecture
  - Java Authentication and Authorization Service (JAAS)
  - Java API for XML Procesing (JAXP)
  - SOAP with Attachments API for Java (SAAJ)
  - Servicios Web (JAX-WS)
  - Java API for RESTful Web Services (JAX-RS)

Ensamblado y despliegue de componentes Java EE

- Los componentes se instalan en contenedores desde los que pueden utilizar los servicios de la plataforma
- El proceso de ensamblado de los componentes requiere especificar el soporte del servidor J2EE
  - Seguridad: usuarios autorizados
  - Modelo de gestión de transacciones: relaciones entre métodos que constituyen una transacción (tratados como una unidad)
  - Java Naming and Directory Interface (JNDI): acceso a servicios de nombres y directorio
  - Conectividad remota: permite que los clientes invoquen métodos en los EJBs como si estuvieran en la misma máquina virtual

Composición de módulos en aplicaciones

- Una aplicación Java EE se puede entregar como ficheros:
  - Java Archive (JAR)
  - Web Archive (WAR) file
  - Enterprise Archive (EAR)
    - Módulos Java EE
    - Descriptor de despliegue (documento XML con extensión .xml)

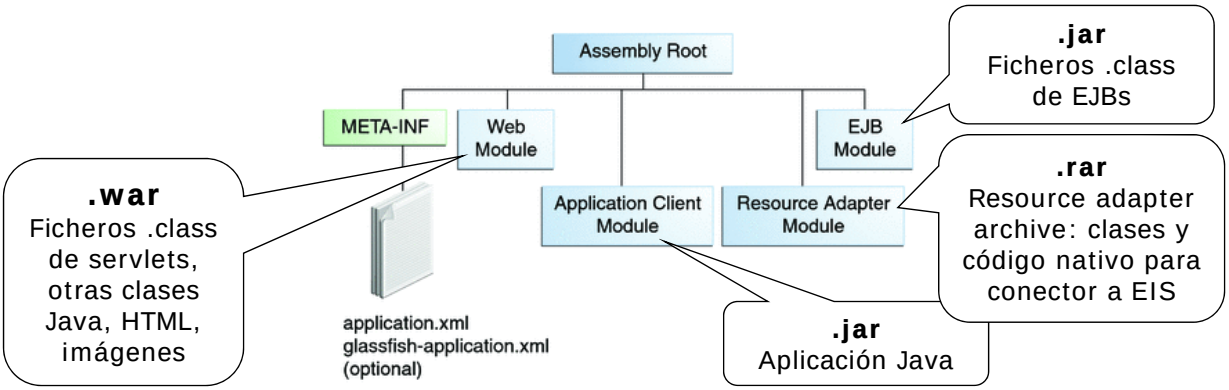
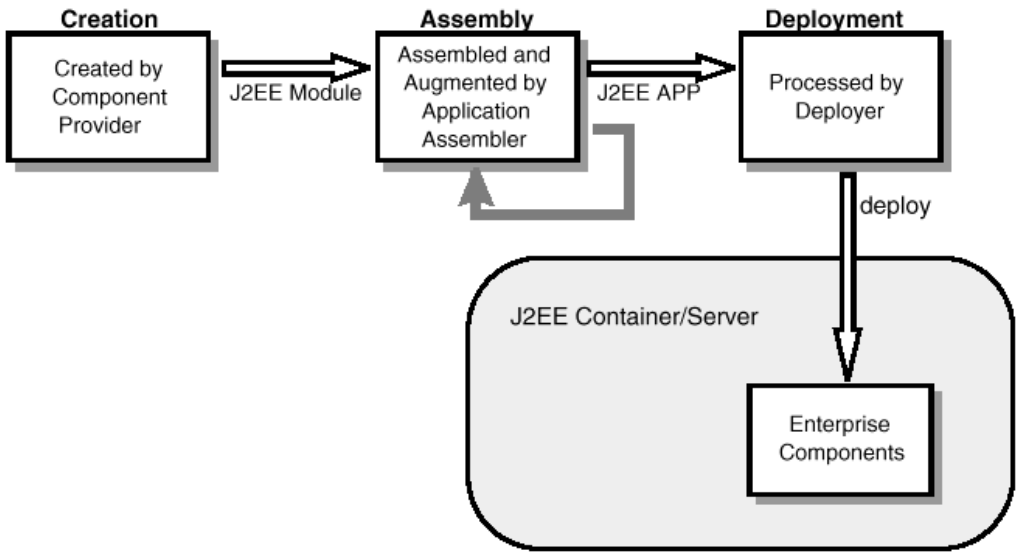


Figura 1-6 EAR File Structure, de *The Java EE 6 Tutorial* (2013).
http://docs.oracle.com/javaee/5/tutorial/doc/bnabo.html

Ciclo de vida de una aplicación Java EE



Fuente: Sun Microsystems, Inc., J2EE Connector Architecture Specification

Frameworks Java EE

- Apache Struts
- Spring
- JBoss Seam
- GWT (Google Web Toolkit)
  - Aplicaciones en Java basadas en Ajax
- Stripes, Tapestry, Wicket, Maverick, etc.

- Ver http://www.slideshare.net/ikercanarias/frameworks-j2ee

Bibliografía

- Eric Jendrock et al. The Java EE 6 Tutorial (2013). http://docs.oracle.com/javaee/6/tutorial/doc/

- Documentación oficial
  - Java EE Specifications http://www.oracle.com/technetwork/java/javaee/tech/index.html
  - API specification for version 6 of Java EE http://docs.oracle.com/javaee/6/api/
  - API specification for GlassFish Server, including Java EE 6 platform packages and nonplatform packages that are specific to the GlassFish Server product http://glassfish.java.net/nonav/docs/v3/api/

- J2EE es más eficiente que PHP:

http://www.mgrecol.com/2012/08/09/php-vs-j2ee-a-practical-approach/