

Sistemas Informáticos II

Práctica 1 – Segunda parte

EJB

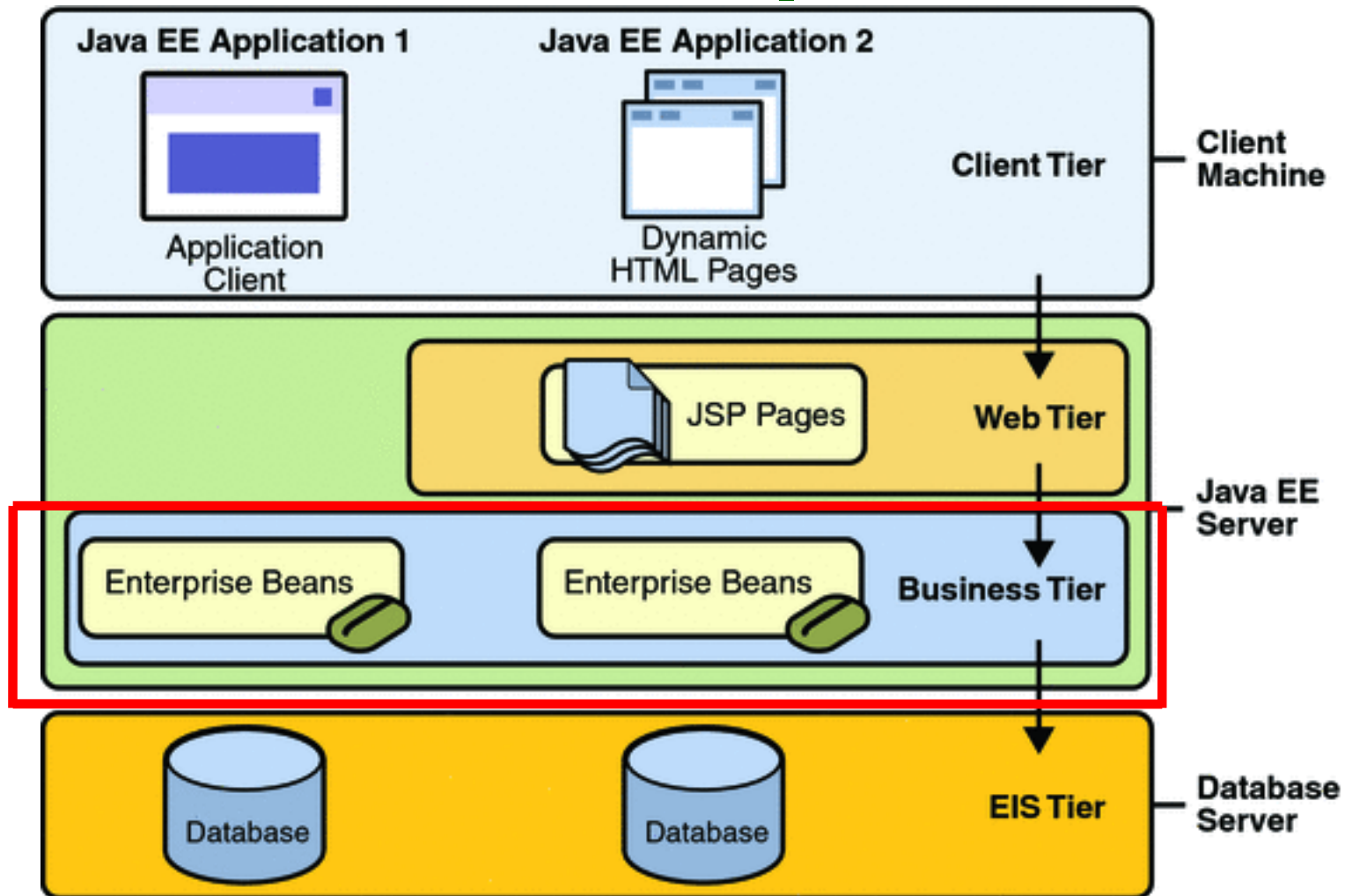
Prácticas 2016/17

Jaime López

Objetivos y entorno

- Objetivos
 - Conocer y experimentar con la tecnología de **Enterprise Java Beans (EJBs)**
 - Mensajería **JMS**, gestores de colas
 - **Message-Driven Beans**
- Material entregado
 - **P1-ejb-base.tgz**: Proyecto de ejemplo con la estructura para EJB Session Stateless (con interfaz local)
 - **P1-ejb-transaccional-base.tgz**: Modificaciones para comprobar la transaccionalidad de un EJB
 - **P1-jms-base.tgz**: Proyecto de ejemplo con cliente JMS y servidor MDB para envío de mensajes
- Entorno
 - Glassfish v4.0
 - Línea de comandos
 - Opcionalmente: NetBeans para depurar código

Aplicaciones web distribuidas multicapa



Capa Negocio

- Primera parte: **EJB** sesión sin estado con **interfaz local**

P1-ejb

- **Objetivo:** convertir la clase VisaDAOWS realizada en la P1a en un EJB stateless.
- **Interfaz local:** cliente y servidor residirán en el mismo servidor.
- **Interfaz remota:** cliente y servidor residirán en distintos servidores.

- Segunda parte: Transaccionalidad **EJB gestionada por el contenedor**

P1-ejb-transaccional

- **Objetivo:** comprobar la correcta gestión de transaccionalidad realizada por parte del contenedor del EJB.
- Modificar *realizaPago* para comprobar su funcionamiento transaccional → añadir campo de saldo en la tabla de tarjetas de la base de datos.

- Tercera parte: **MDBs** y colas de mensajes JMS

P1-jms

- **Objetivo:** Familiarizarse con el uso de mensajería JMS, gestores de colas, y Message-Driven Beans (MDB).
- Incluir en la aplicación la posibilidad de que un agente externo realice la cancelación de un pago mediante el envío de un mensaje.

EJB vs JavaBeans

- **JavaBean:** objetos POJO (Plain Old Java Object)
 - Modelo más simple y adecuado para componentes sencillos.
 - No se pueden usar en entornos objetos distribuidos ya que no soportan RMI
 - No soportan transaccionalidad
- **EJB:** presentan mayor complejidad, pero a cambio
 - Soportan invocación remota: mayor escalabilidad
 - Proporcionan transaccionalidad abstrayendo al programador de su gestión

EJB vs WS

- **EJB:** Emplea RMI-IIOP
 - Transferencias más ligeras (protocolo binario)
 - Protocolo IIOP actualmente limita la interconectividad a través de firewalls en comparación con HTTP
 - Limitado a Java
 - **Permite interfaz de invocación local más eficiente**
 - Gestión automática de la transaccionalidad
- **WS:** SOAP emplea XML sobre HTTP
 - Transferencias más pesadas (XML vs binario)
 - Mayor interconectividad
 - Soportado por más lenguajes de programación
 - No permite invocación local – la permite, pero es igual de ineficiente
 - Gestión manual de la transaccionalidad

Transaccionalidad

- **Gestionada por la aplicación (*Bean Managed*):**
 - La aplicación puede decidir abortar la transacción
 - La “marcha atrás” se hará a mano, deshaciendo todas las operaciones (ROLLBACK sobre la base de datos, modificaciones manuales sobre el modelo)
- **Gestionada por el contenedor (*Container Managed*):**
 - La aplicación lanza **EJBException**
 - Todos los cambios son deshechos automáticamente

EJB vs MDBs

- **EJB:** Su ejecución viene desencadenada por la invocación de un método (local o remoto) por otra aplicación
- **MDB:** Su ejecución viene desencadenada por la llegada de un mensaje (método `onMessage()`)

JMS

- **Destinos de conexión:**
 - Contenedores de mensajes a los que se conecta la aplicación para enviar / obtener mensajes
 - Dos tipos:
 - **Message Queues (MQs):** Un lector, varios escritores
 - **Message Topics (MTs):** Varios lectores, varios escritores
- **Factorías de conexión:** Permiten la instanciación de destinos de conexión de manera ordenada

Entrega

- La **entrega de los resultados de esta práctica se registrará por las normas expuestas durante la presentación de la asignatura.**
- Nomenclatura del fichero a entregar
SI2P1B_<grupo>_<pareja>.zip
(ejemplo: SI2P1B_2311_1.zip)
- Contenido del fichero:
 - Informe técnico siguiendo la plantilla publicada en la página del laboratorio con las respuestas a todas las preguntas
 - **P1-ejb** con las modificaciones que hayan sido necesarias para el EJB.
 - **P1-ejb-servidor-remoto** con modificaciones para servidor remoto.
 - **P1-ejb-cliente-remoto** con modificaciones para cliente remoto.
 - **P1-ejb-transaccional** con las modificaciones que hayan sido necesarias para comprobar la transaccionalidad de los EJB
 - **P1-jms** con las modificaciones que hayan sido necesarias para el MDB.
- Entrega: semana del 13-17 de Marzo de 2016, antes del comienzo de la clase