Construa o diagrama de componentes e implantação para ilustrar a arquitetura do sistema descrito a seguir.

Considere um sistema de livraria virtual cuja arquitetura está descrita a seguir:

- A aplicação possui 3 componentes, no padrão arquitetural MVC: Páginas Web, Objetos de Negócio, Objetos Persistentes.
- Uso de 4 servidores distribuídos em 4 máquinas distintas: Container Web (IBM WAS V8), Container EJB (IBM WAS V8), BD (Oracle 11g), nomes (Oracle Names Server).
- O servidor de nomes implementa o protocolo LDAP 3 para recuperar e gravar o login (autenticação) e informações de um usuário.
- O servidor Web suporta a especificação Servlet 3.0 e JSP 2.0 para Páginas Web, o servidor EJB suporta a especificação EJB 3.0 para Objetos de Negócio.
- A API Hibernate 4.0 é utilizada para implementar os objetos persistentes, ou seja, realizar o mapeamento de objetos para o esquema relacional do BD.
- O esquema do BD segue a especificação SQL ANSI.
- O link de comunicação entre os containers Web e EJB é de 10 Mbps.
- O link de comunicação entre o container EJB e os servidores de BD e Nomes de é 100 Mbps.
- O link de comunicação entre clientes da aplicação e o container Web é de 2Mbps.
- O protocolo de comunicação entre as máquinas é TCP/IP

> **Obs:** No diagrama de deployment não é necessário exibir as interfaces, apenas as dependências. Entretanto, elas podem ser representadas, se se desejar.

Alguns conceitos que podem ajudar (Fonte: wikipédia)

Um **Container**, em programação orientada a objetos, é um objeto que contém outros objetos.Estes objetos podem ser incluídos ou removidos dinamicamente, em tempo de execução, diferentemente do que ocorre com uma composição onde este relacionamento é fixado em tempo de compilação.

Em Java EE, o *container* contém os componentes construídos como Servlets (*container* para aplicações Web) ou EJBs (*container* para componentes de negócio). Um exemplo de *container* para Web é o Tomcat. Quando uma aplicação web faz uma solicitação para um Servlet, o servidor não entrega a solicitação diretamente ao Servlet, mas sim para o *container* que contém o Servlet.O container gerencia o ciclo de vida, dá suporte ao multithread, segurança, e suporte para páginas JSP.

Servlet é um componente que disponibiliza ao programador da linguagem Java uma interface para o servidor web (ou servidor de aplicação), através de uma API. As aplicações baseadas no Servlet geram conteúdo dinâmico (normalmente HTML) e interagem com os clientes, utilizando o modelo *request/response*. Os *servlets* normalmente utilizam o protocolo HTTP, apesar de não serem restritos a ele. Um Servlet necessita de um container Web para ser executado.

EJB ou **Enterprise JavaBeans** é um dos principais componentes da plataforma JEE (*Java Enterprise Edition*). É um componente do tipo servidor que roda no *container* para EJB do servidor de aplicação. Os principais objetivos da tecnologia EJB são fornecer rápido e simplificado desenvolvimento de aplicações Java baseadas em componentes, distribuídas, transacionais, seguras e portáveis.

A plataforma JEE provê algumas facilidades dedicadas à camada de lógica de negócio e para o acesso ao banco de dados. Através do EJB o desenvolvedor utiliza a infraestrutura do servidor de aplicação voltada para o desenvolvimento de aplicações de missão crítica (de alta importância para a empresa) e de aplicações empresarias em geral.

No caso do Oracle 11g, o esquema do BD contém a estrutura de todo o banco de dados.

O **Hibernate** é um *framework* de acesso a banco de dados escrito em Java. Ele é um software livre de código aberto distribuído com a licença LGPL. O objetivo do Hibernate é facilitar a construção de aplicações Java dependentes de bases de dados relacionais, particularmente,facilitar o desenvolvimento das consultas e atualizações dos dados. O uso de ferramentas de mapeamento objeto relacional, como o Hibernate, diminuem a complexidade resultante da convivência de modelos diferentes; o modelo orientado a objetos (da linguagem Java) e o relacional (da maioria dos SGBDs). O Hibernate é responsável apenas pelo mapeamento das tabelas do modelo relacional para classes da linguagem Java.

Passos para criar o diagrama de componentes:

1. Criar componentes Páginas Web, Objetos de Negócio e Objetos Persistentes. Criar as relações de dependência entre eles: Páginas Web dependem de Objetos de Negócio que, por sua vez, dependem de Objetos Persistentes
2. Criar os 4 componentes correspondentes aos servidores : Servidor Web IBM WAS V8, Servidor EJB IBM WAS V8, Servidor Oracle 11g e o Servidor Oracle Names Server.
3. Criar a interface implementada LDAP 3 no Servidor Oracle Names Server
4. Criar uma interface requerida nos Objetos de Negócio. Essa interface requer a interface LDAP 3 (os Objetos de Negócio são os responsáveis por fazer a autenticação via Oracle Names Server).
5. As Páginas Web usam as especificações Servlet 3.0 e JSP 2.0 (que são interfaces) providas pelo Servidor Web IBM WAS V8. Crie as interfaces implementadas e requeridas entre os dois componentes.
6. Os Objetos de Negócio usam as especificações EJB 3.0 (que é uma interface) providas pelo Servidor EJB IBM WAS V8. Crie as interfaces implementadas e requeridas entre os dois componentes (Objetos de Negócio sempre dependem das interfaces EJBs).
7. Criar os componentes Hibernate 4.0 e Esquema do banco de dados.
8. Criar a interface entre o Esquema do banco de dados e o Servidor Oracle 11G através do SQL ANSI(o Esquema requer a a interface SQL ANSI provida pelo Servidor Oracle 11g)
9. Criar a dependência entre os componentes Objetos Persistentes, Hibernate 4.0 e Esquema. Objetos Persistentes dependem do Hibernate 4.0 que, por sua ves, depende do Esquema.

Passos para criar o diagrama de deployment:

1. Como temos 4 servidores (componentes) distribuídos em 4 máquinas distintas, representar cada servidor em um nó do diagrama de deployment, cada nó representando uma máquina. No diagrama de componentes criado, identifique quais componentes vão estar cada máquina: componentes Oracle 11g e Esquema vão estar em uma máquina servidora denominada Servidor de BD P 4GHz, HD 1Tb, MP 32Gb; componentes Servidor EJB IBM WAS V8, Objetos

de Negócio, Objetos Persistentes e Hibernate 4.0 vão estar em uma máquina servidora denominada Servidor EJB P 4GHz, HD 500Gb, MP 32Gb; componente Oracle Name Server vai estar em uma máquina servidora denominada Servidor de Nomes P 4GHz, HD 500Gb, MP 32Gb; componentes Servidor Web IBM WAS V8 e Páginas Web vão estar em uma máquina servidora denominada Servidor Web P 3GHz, HD 500Gb, MP 16Gb.

2. Desenhe cada nó (máquina) contendo os componentes identificados em 1. Não desenhe as interfaces, como no diagrama de componentes, troque-as por dependências. Desenhe todas as dependências que estavam no diagrama de componentes.
3. Criar um nó Cliente(que representa todas as máquinas de clientes que acessam o sistema) que contém um componente chamado Cliente, que depende do componente Páginas Web (pois é através delas que o cliente vai acessar o sistema, via browser).
4. Representar os protocolos e os links de comunicação entre os nós: entre Cliente e Servidor Web P 3GHz, HD 500Gb, MP 16Gb crie um link TCP/IP 2Mbps; entre Servidor Web P 3GHz, HD 500Gb, MP 16Gb e Servidor EJB P 4GHz, HD 500Gb, MP 32Gb crie o link TCP/IP 10Mbps; entre Servidor EJB P 4GHz, HD 500Gb, MP 32Gb e os nós Servidor de Nomes P 4GHz, HD 500Gb, MP 32Gb e Servidor de BD P 4GHz, HD 1Tb, MP 32Gb crie um link TCP/IP 100Mbps.

Diagrama de Componentes

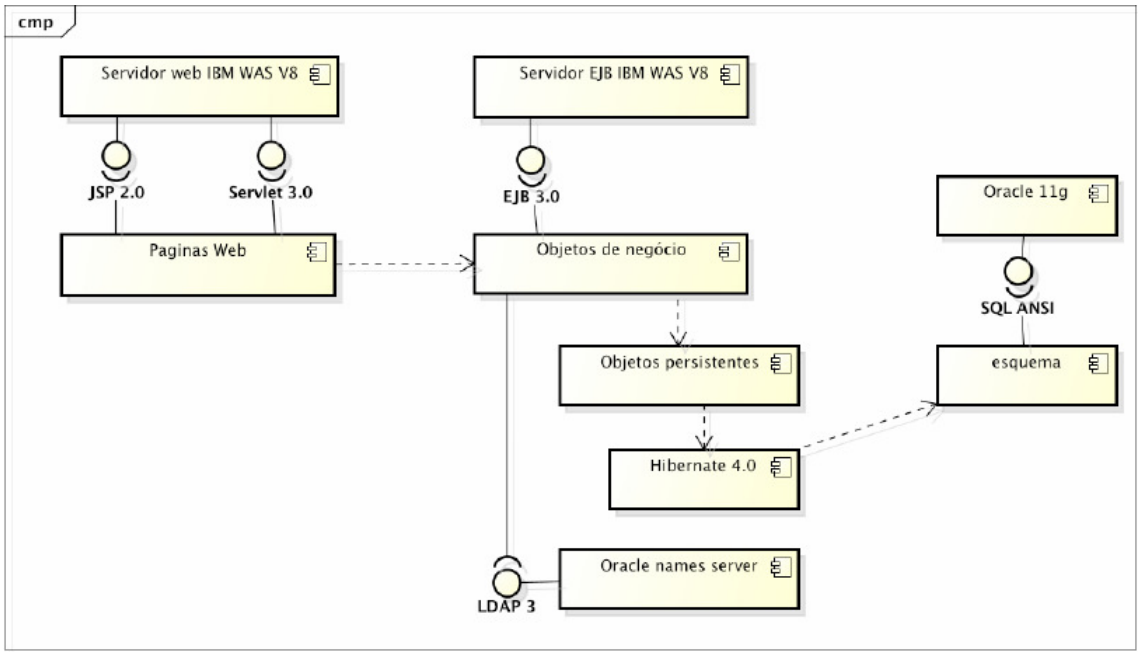
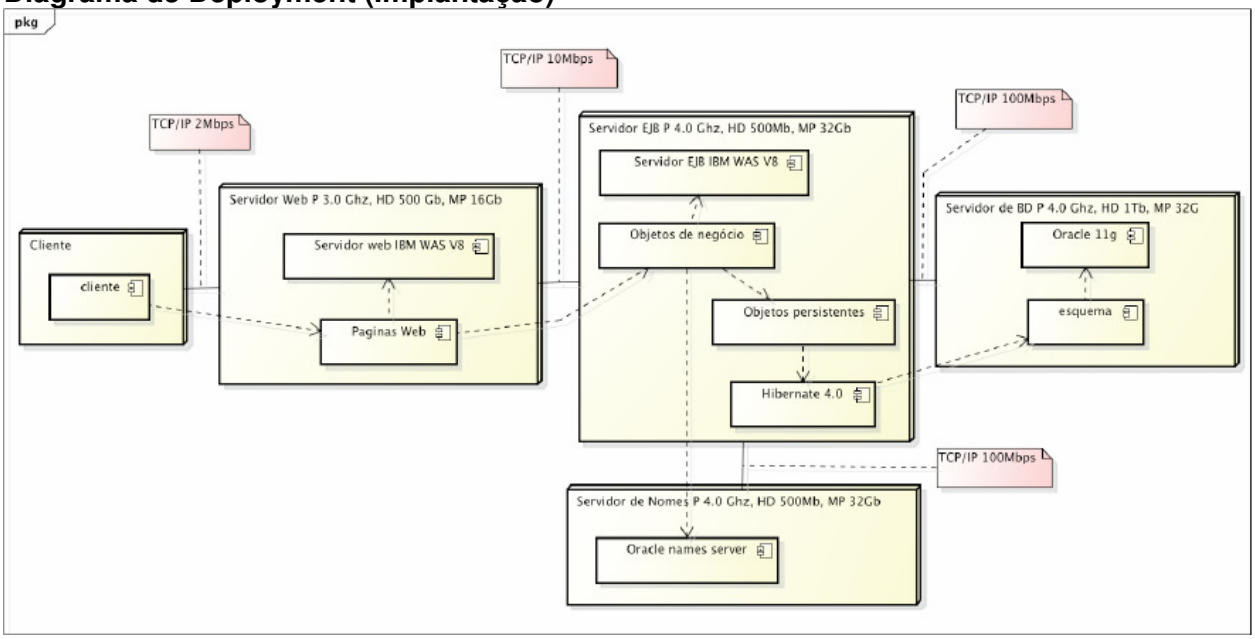


Diagrama de Deployment (Implantação)



Inspirado em exercício elaborado pela Prof. Maria Augusta Vieira Nelson do Depto. de Eng. de Software e Sistemas de Informação da PUC Minas.