Agile Parking

Projeto Arquitetural

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

# Objetivo

As arquiteturas utilizadas no sistema baseiam-se na arquitetura em camadas MVC.

Objetivos:

* Separação de componentes por responsabilidades em comum.
* Utilizado para estruturar e definir melhor as funcionalidades de um sistema.
* Separar o código de acesso aos dados do código de negócio e interface com o usuário (Modularidade)

# Metas Arquiteturais e Filosofia

* Implantação de um sistema Web Simples e de fácil acesso;
* Necessidade de manutenção mínima;
* Verificação de desempenho continua;
* Aplicações de verificação de usuários e funcionários no banco de dados;
* Sistema será adaptado para funcionar em diversos navegadores;

# Premissas e Dependências

* Atender a todas as funcionalidades do sistema;
* Qualificação na plataforma escolhida;
* Habilidade nas ferramentas escolhidas;
* Aproveitamento do tempo ocioso durante o projeto;

# Requisitos Críticos da Arquitetura

<http://pt.wikipedia.org/wiki/CRUD>

<http://java.sun.com/javaee/javaserverfaces/>

<http://pt.wikipedia.org/wiki/MVC>

# Decisões, Restrições e Justificativas

* Implementação de ferramentas auxiliares: Foram usadas ferramentas como primeface e hibernate para facilitar no designer e na comunicação do sistema web.
* Padrão da Arquitetura: Utilização de padrão MVC.
* Mudança no cronograma: Devido à dificuldade de acesso à internet.

# Mecanismos Arquiteturais

A arquitetura desse projeto é baseada no fundamento JSF com Primeface e injeção de dependências, logo para cada camada existe um jeito diferente de se fazer nessa seção ira descrever cada um para se utilizar esse framework.

**Construir um CRUD:**

Em termos de desenvolvimento, um CRUD estabelece os procedimentos básicos operacionais para um componente de um sistema. Por componente, utilizando orientação a objetos, entendemos como um objeto pertencente ao modelo (um Cliente, um Produto, por exemplo). Por procedimentos básicos, entendemos como as operações de inclusão (create), seleção (read), atualização (update) e exclusão (destroy) associadas a um desses objetos. Todo sistema possui seus CRUDs. Ao mostrar um exemplo de criação de CRUD, temos base para a criação de várias etapas de um sistema.

**Trabalhar com JSF:**

O Java ServerFaces é uma das mais atuais alternativas para se construir a camada view da aplicação web. Ela permite ao desenvolvedor utilizar tags específicas, sem nenhuma codificação nativa em Java (scriptles) no arquivo jsp. As páginas dinâmicas ficam ligadas diretamente aos Beans. A produtividade é favorecida e a manutenção facilitada, como veremos.



**Padrão JSF com MVC:**

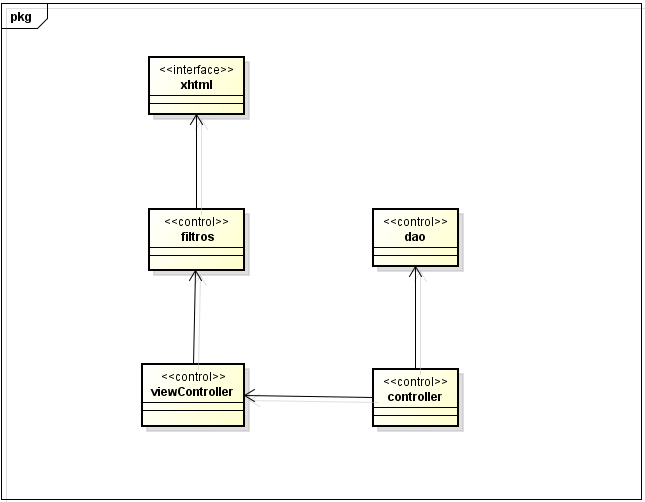
No JavaServer Faces, o controle fica por conta de um servlet chamado Faces Servlet, por arquivos de configuração (ex.: faces-config.xml), pelos Backing Beans e pelos validadores e conversores. O Faces Servlet se designa a receber requisições, encaminhá-las ao modelo e enviar as respectivas repostas. Os arquivos de configuração são responsáveis por definirem a navegação entre páginas e o mapeamento de ações. Já os validadores e conversores permitem um maior controle sobre os dados que serão enviados. Importante ressaltar que, mesmo sendo referenciados nas páginas, os conversores e validadores fazem parte do processamento de eventos, logo também fazem parte do controlador e não da visão.  
  
O modelo é constituído por classes de entidade e de negócio, que por sua vez, recebem dados da camada de visualização e executam regras pertinentes ao negócio.  
  
Já a visualização, remete ao que o usuário vê, ou seja: a interface - páginas JSP e kits renderizadores (HTML, WML, XML, etc.). Mas, como o JSF se utiliza de componentes, pode se dizer que esta é composta por uma hierarquia de componentes organizados.  
  
De modo simplificado, o MVC em aplicações web é distribuído da seguinte forma:  
  
**Visão**:  
• Componentes UI em páginas JSP/XHTML  
• Kits renderizadores (HTML, WML, XML, etc.)

**Controlador**:

• Faces Servlet (Front Controller)  
• Backing Bean (Page Controller ou Modelo)  
  
**Modelo**:  
• entidades e regras de negócio  
• Objetos gerais da aplicação (dados, etc.)



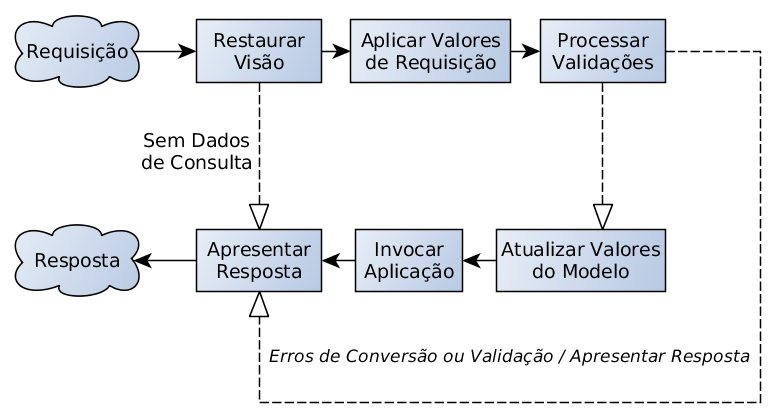
# Principais Abstrações



# Camadas do Framework da Arquitetura

O JSF, diferente de outros frameworks, possui um processamento de requisição dividido em seis fases:

1. Restauração da visão - o objetivo principal desta fase é construir a [árvores de componentes](http://pt.wikipedia.org/w/index.php?title=%C3%81rvores_de_componentes&action=edit&redlink=1).[7](http://pt.wikipedia.org/wiki/JavaServer_Faces" \l "cite_note-7) Ela utiliza o template e cria a árvore inicial através da análise da requisição. Após isto, ela salva o estado da árvore no objeto FacesContext. Nas requisições subsequentes, ela cria a árvore do estado e procede a execução do resto do ciclo de vida.
2. Aplicação dos valores de requisição - o JSF pega cada componente da árvore começando com a raiz e a cria ou recupera do objeto FacesContext. Cada componente na árvore gerencia seus próprios valores e toma-os dos parâmetros, cookies e cabeçalhos da requisição HTTP.
3. Validações de processo - o JSF realiza a conversão e validação sobre todos os componentes começando com o raiz. O valor submetido de cada componente é convertido em um objeto e validado chamando-se o validador registrado. O JSF salva o valor submetido. Se ocorrer um erro durante a conversão ou validação, o ciclo de vida escapa diretamente para a fase de "apresentação da resposta".
4. Atualização de valores de modelo - durante esta fase, o valor do componente é passado para o modelo através da atualização das propriedades dos backing beans.
5. Invocação da aplicação - a manipulação de evento para cada ação e ouvidor de ação é executada começando com o(s) ouvidor(es) de ação e então a chamada do método de ação.Apresentação da resposta



# Visões Arquiteturais

