# CMPROJECT

# 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

Este documento descreve a filosofia, as decisões e os elementos que serão usados no decorrer da implementação do projeto.

# Metas Arquiteturais e Filosofia

# Requisitos Críticos da Arquitetura

- Intellij IDEA - https://www.jetbrains.com/idea/

- Eclipse IDE - <http://goo.gl/ueRsJf>

- JSF 2.2 - <http://goo.gl/Tc2yf9>

- MySQL Server - <http://goo.gl/LPdhZ5>

- Apache Tomcat 8 - <http://goo.gl/td0jiA>

# Decisões, Restrições e Justificativas

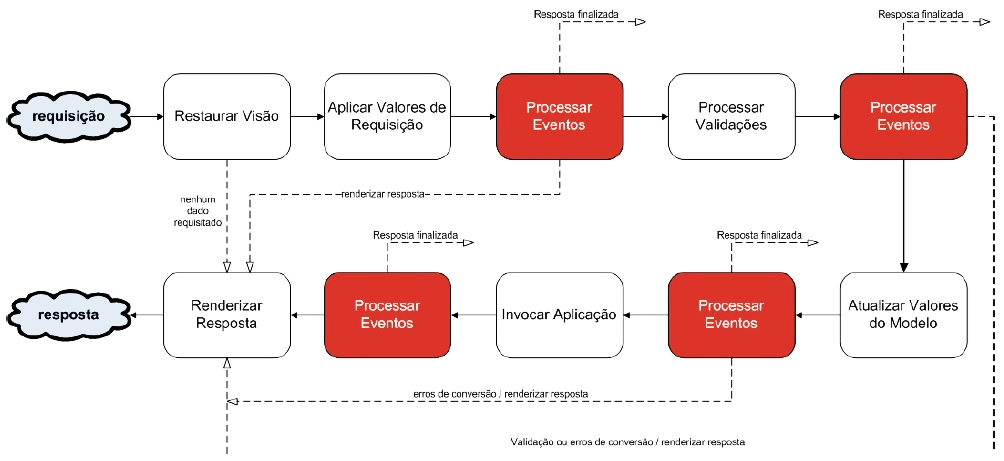
# Mecanismos Arquiteturais

Para esse projeto será utilizado o framework Java Server Faces (JSF) para o desenvolvimento da aplicação na web, que será baseado nas características do padrão de arquitetura MVC (Model-View-Controller), onde se tem uma divisão de tarefas por camadas, podendo assim ter um melhor reaproveitamento do código.

O JSF tem importantes serviços como:

* Arquitetura MVC: JSF adota os padrões MVC (Model-View-Controller), onde divide as funcionalidades em três camadas para ter um aproveitamento melhor.
* Validação e manipulação de erros: O JSF facilita a criação de regras de validação.
* Internacionalização: JSF fornece suporte à internacionalização.
* Componentes customizados: JSF dá liberdade para os desenvolvedores criarem componentes próprios ou usar componentes feitos por terceiros.
* Renderizadores alternativos: Por padrão o JSF produz como saida XHTML, mas também pode produzir outros tipos como WML (Wireless Markup Language) ou XUL (XML User Interface)

Ciclo de vida JSF é dividido em 6 fases:



Fase 1 (Restaurar a visão): Caso a página já tenha sido exibida, é recuperada toda a árvore de componentes configurados no seu estado anterior. Caso esteja sendo exibida pela primeira vez é construída uma nova árvore de componentes. Se a requisição não possuir dados solicitados o JSF pula para a fase de Renderizar Resposta.

Fase 2 (Aplicar Valores de Requisição): Nessa fase o JSF resgata todos os valores informados pelo usuário e os armazena em seus objetos.

Fase 3 (Processar Validações): A cadeia de entrada com o valor enviado é convertida para o tipo correto do objeto. Caso ocorra algum erro de validação uma mensagem de erro é adicionada no FacesContext, o componente é marcado como inválido e a implementação JSF invoca a fase de Renderizar Resposta, renderizando a visão ao usuário levando as mensagens de erro. Caso contrário o ciclo de vida continua normalmente.

Fase 4 (Atualizar Valores do Modelo): Atribui valores informados pelo usuário no formulário, para as respectivas propriedades associadas aos ManagedBeans. Pode haver erro na conversão, fazendo com que o JSF dispare um erro de tempo execução, caso ocorra o JSF adiciona esses erros no FacesContext e renderiza a página de visão ao usuário.

Fase 5 (Invocar Aplicação): O controlador do JSF chama o método associado ao submeter o formulário, disparando assim a camada de regras de negócio da aplicação. Todos os valores foram validados e carregados nas fases anteriores, por isso poderemos usá-los conforme necessitar. Geralmente é retornada uma string de resultado do método para JSF efetuar a naegação, se esse valor for null o JSF retorna a mesma página que chamou o método.

Fase 6 (Renderizar Resposta): Codifica a resposta e a envia de volta ao navegador.

# Principais Abstrações

# 

# - Pagina JSF: Controla os componentes visuais da aplicação.

# - Managed Been: Recebem as chamadas da camada de visualização e executam as chamadas necessárias a camada de negócio.

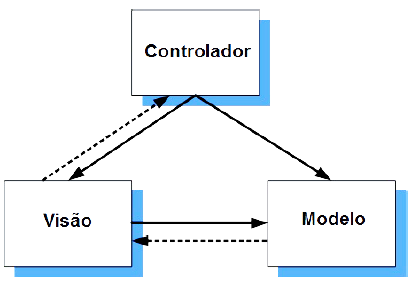
# - Service: Responsável por implementar as regras de negócio da aplicação.

# - DAO: Responsável por realizar as operações no banco de dados.

# O fluxo da aplicação tem início na camada de visualização sendo disparado por algum evento executado pelo usuário através de uma página JSF. O managed bean controlado pelo JSF recebe o evento, valida as entradas de dados e realiza a chamada a camada de serviços. A camada de serviço pode realizar uma ou mais chamadas à camada de persistência, devolvendo o modelo atualizado para a camada de serviço e consequentemente a camada de visualização.

# Camadas do Framework da Arquitetura

A arquitetura MVC (Model-View-Controller) separa claramente as responsabilidades e a comunicação feita entre os componentes da aplicação, as camadas se separam em Modelo, Visão e Controlador.



- **Visão** (View): É responsável pela interface com o usuário, controlando as entradas e saídas gráficas e textuais.

- **Modelo** (Model): Controla os comportamentos e os dados do domínio da aplicação, respondendo as solicitações e instruções para mudar seu estado, além de conter as regras de negócio.

- **Controlador** (Controller): Controla as solicitações do usuário repassando as mesmas para o modelo ou para a visualização, adequadamente. Esta divisão de tarefas tem como objetivo de aumentar a flexibilidade e a reutilização do código.

# Visões Arquiteturais

* **Lógica:**
* **Operacional:**
* **Casos de uso:**

