# 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

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

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

- Apache Tomcat 7 ou superior - <http://goo.gl/td0jiA>

- MySQL Connector Java.

# Decisões, Restrições e Justificativas

# Mecanismos Arquiteturais

Para esse projeto será utilizado o Java Server Page 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 em camadas, podendo assim ter um melhor reaproveitamento do código.

- JSP (Java Server Page) é a tecnologia Java para construção de páginas dinâmicas.

- JSTL (JSP Stantard Tag Library) é uma biblioteca de tags que nos ajudam na construção de página dinâmica sem utilizarmos a scriplet (código Java diretamente na página)

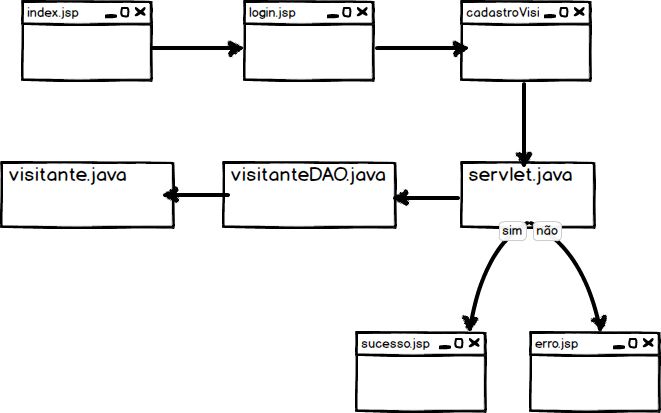
- Servlet são classes especiais java que recebem e respondem a requisições do protocolo HTTP.

- JDBC (Java Database Connectivity) é a forma padrão de acesso a Banco de dados em Java

- MySQL é um banco de dados relacional gratuito e de grande aceitação no mercado e com características muito boas para sistemas na Internet

- Padrão MVC – Model View Controller é um padrão de projeto que divide nossa aplicação em várias camadas

Ciclo de vida JSP:



# Principais Abstrações

# 

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

# - HttpServlet: Recebe os itens enviados pela página jsp.

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

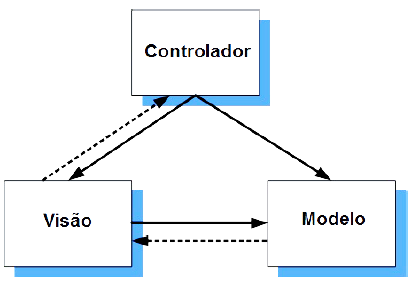
# - DAO: Responsável por realizar a conexão ao banco.

# - EntidadeDAO: responsável por realizar as persistências;

# - Entidade: classe onde será salvo o objeto antes de ser persistido ou após ser recuperado.

# 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:**

