

Projeto NOME PROJETO

Autor		Data		Versão	1.0

Sumário

Requisitos de Sistema.....	3
Requisitos Funcionais	3
RF001: XXXXXX.....	3
RF002: XXXXXX.....	3
Requisitos Não Funcionais;.....	3
RNF001: A solução será XXXXXXXXXX.....	3
RNF002: O sistema terá como XXXXXXXXXX.....	3
Casos de Uso.....	4
Atores	4
Diagrama de Casos de Usos: Gerenciar Mediadores.....	5
UC01 - XXXXXX.....	6
Arquitetura Básica para desenvolvimento	7
Componentes Java.....	7
Design Patterns (padrões de projeto de software)	8
MVC (Model-View-Controller	8
DAO – Data Access Obejct	8
Command	8

Escopo: O sistema deverá gerar

Requisitos de Sistema

Requisitos Funcionais

RF001: XXXXXX.

Possibilita que o usuário

RF002: XXXXXX

Possibilita que o

Requisitos Não Funcionais;

RNF001: A solução será XXXXXXXXXX

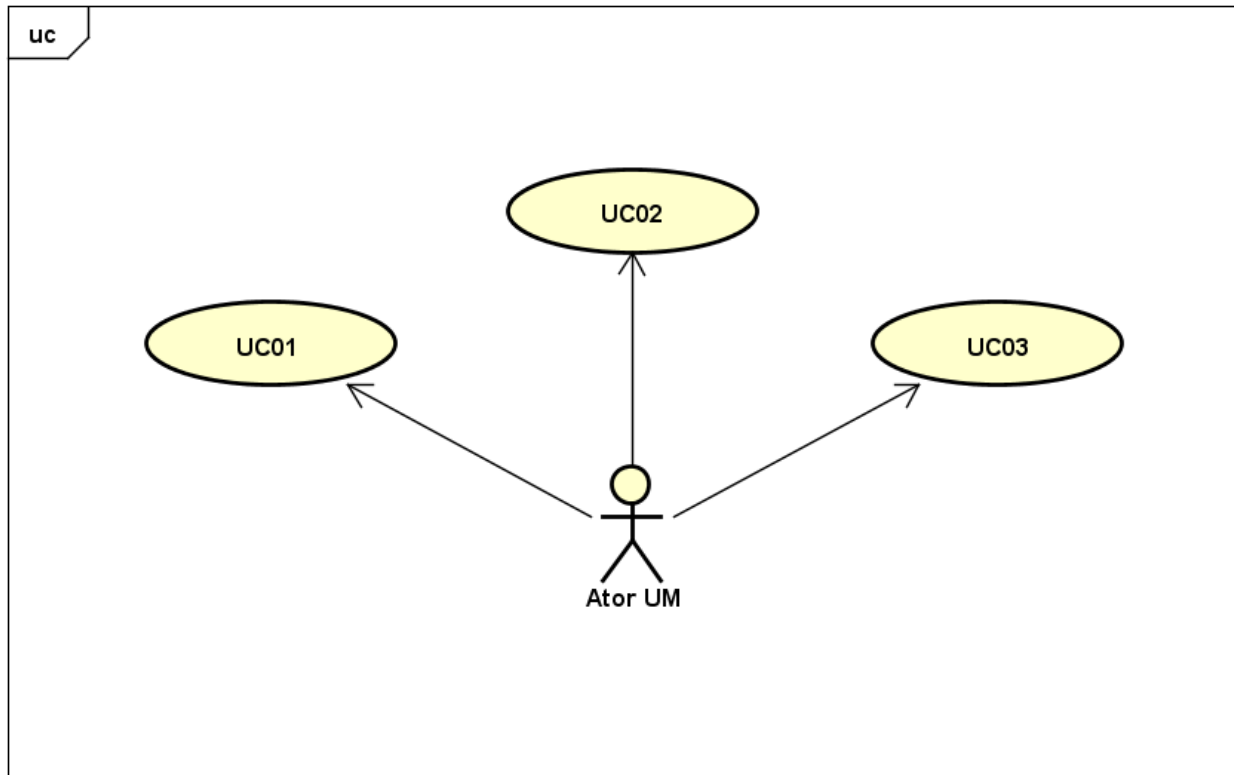
RNF002: O sistema terá como XXXXXXXXX.

Casos de Uso

Atores

Ator	
Ator Um	Responsável ...
Ator Dois	Responsável ...

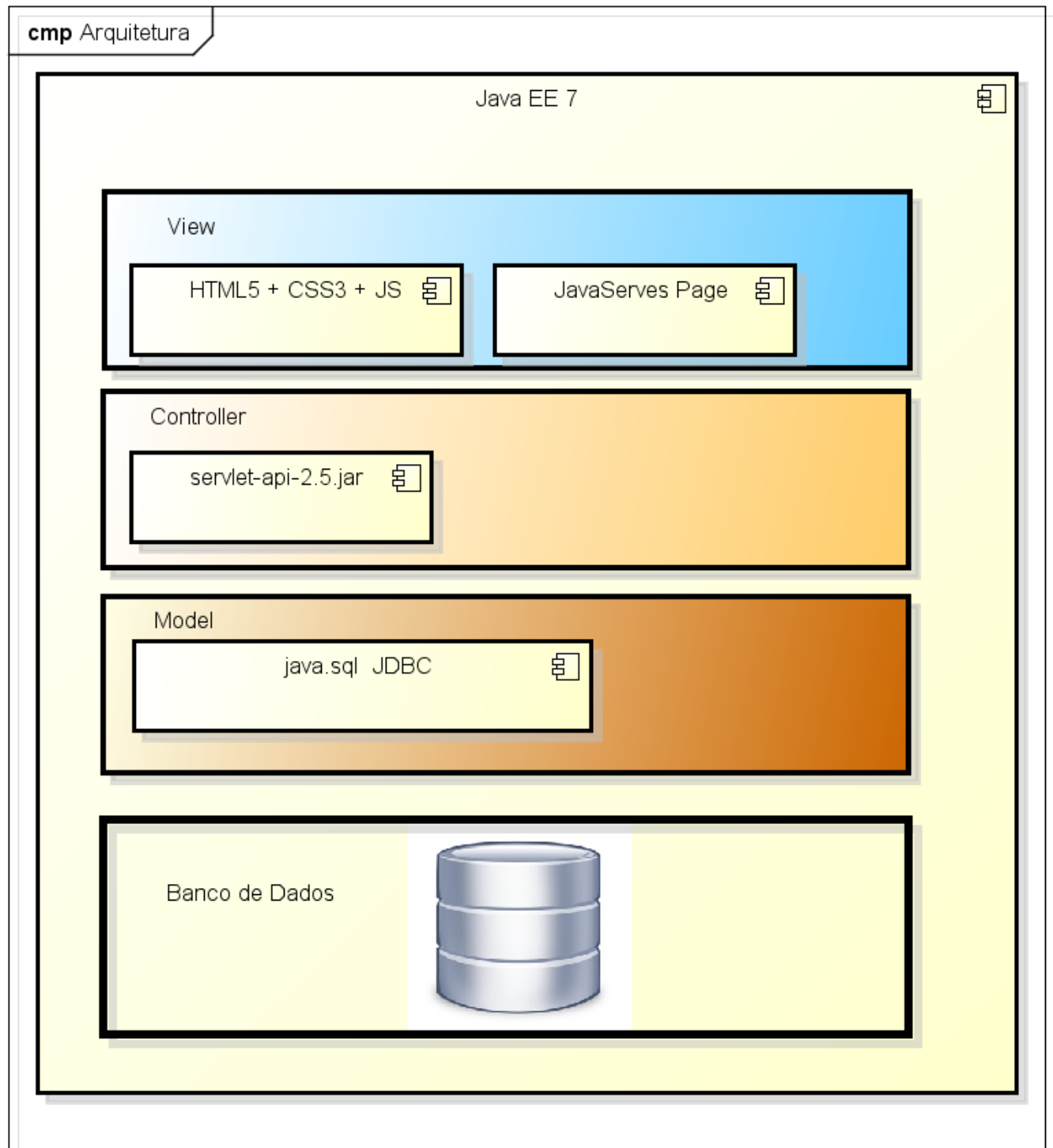
Diagrama de Casos de Usos: Gerenciar Mediadores



Caso de uso	UC01 - XXXXXX
Descrição	
Ator	Ator Um
Pré-Condições	
Pós-Condições	
Cenário Principal	<ol style="list-style-type: none"> 1. xxxx 2. xx <ol style="list-style-type: none"> 2.1. yyyyyy; * 2.2. yyyyyy; * 2.3. y 3. vvvvvv. 4. vvvvv. 5. Caso de uso finalizado. <p>* obrigatório</p>
Cenários Alternativos	<ol style="list-style-type: none"> 5a) xxxxxxx. 1. xxxxxxxs. 2. xxxxxx. 3. O fluxo é retomado no item 4 do cenário principal.
Cenário de Exceção	
Casos de uso Associados	
Regras de Negócio	RN001 – Tipo e tamanho dos campos do livro: <ol style="list-style-type: none"> 1. Matrícula (verificar GTIT): Texto 120 caracteres. * 2. Nome: Texto 120 caracteres. * 3. Data Cadastro: data dd/mm/aaaa. *
Notas	

Tela

Arquitetura Básica para desenvolvimento Componentes Java



Design Patterns (padrões de projeto de software)

O sistema será desenvolvido respeitando utilizando os seguintes Design Patterns:

MVC (Model-View-Controller)

Sendo assim divididas suas camadas de desenvolvimento:

- VIEW: Páginas JSP, JavaServer Pages + HTML5 + CSS3 + JavaScript.
- CONTROLLER: Classes Java Servlets.
- MODEL: Classes Java JDBC integrando com banco de dados MySQL.

DAO – Data Access Object

Esse Pattern propõe que tenhamos todas essas operações CRUD de forma transparente e desacoplado dos objetos negócio e coesos com o banco de dados.

Command

Permite encapsular um objeto, e fornecer a outro uma forma de executar o método do objeto encapsulado de forma dinâmica, com esse Pattern conseguimos criar um controle para nossos objetos de forma dinâmica.