



IME - Instituto de
Matemática e Estatística

MAC0416 / MAC5855 - Tópicos Especiais em Desenvolvimento para Web

Prof. Marco Aurélio Gerosa



Sistema Gestão de Biblioteca

Tabela de Conteúdos

Sistema Gestão de Biblioteca.....	1
1Especificação.....	3
1.1Identificação do problema.....	3
2Análise do Projeto.....	3
2.1Objetivos.....	3
2.1.1Objetivo Geral.....	3
2.1.2Objetivo Específico.....	3
2.2Requerimentos.....	3
2.2.1Requerimentos Funcionais BackOffice.....	3
2.2.2Requerimentos Funcionais OnLine.....	4
2.3Diagrama de Casos de Uso.....	4
2.4Diagrama de Classes.....	5
3Arquitetura.....	5
3.1Distribuição de Camadas.....	5
3.2Diagrama da arquitetura.....	5
3.3Especificação da arquitetura.....	6
3.3.1Camada do Cliente.....	6
3.3.2Camada Web.....	6
3.3.3Camada da Lógica do negócio.....	6
3.3.4Camada de Persistência.....	6
4Justificações da escolha das tecnologias.....	7
4.1Ajax, Json e JQuery.....	7
4.2JSF.....	7
4.3Struts.....	7
4.4Spring.....	7
4.5Hibernate.....	7
4.6JUnit e Selenium.....	8
4.7Web Services.....	8
4.8Rest e Mashups.....	8
4.9Ruby on Rails.....	8
5Funcionalidades referentes a Web 2.0.....	8
5.1Predições com Weka.....	8
5.2Recomendações.....	9
5.3Busca e indexação de texto.....	9
6Grau de participação dos integrantes.....	10
6.1Alvaro Mamani Aliaga.....	10
6.2Edith Sonco Mamani.....	10
6.3Erika Guetti Suca.....	10
7Aproveitamento da disciplina.....	10
7.1Alvaro Mamani Aliaga.....	10
7.2Edith Sonco Mamani.....	10
7.3Erika Guetti Suca.....	11
8Referências.....	11

1 Especificação

1.1 Identificação do problema

O cadastramento de obras e o controle de empréstimos são as atividades principais de uma biblioteca. A medida que o volume de obras em uma biblioteca aumenta e o número de usuários também, as atividades de controle deste acervo vão se tornando intensas, dificultando a administração da biblioteca. Os usuários também enfrentam problemas, pois têm dificuldade em localizar as obras desejadas, tendo que muitas vezes se deslocar até a biblioteca e descobrir que o título desejado não existe no cadastro, ou as obras deste título estão emprestadas para outro usuário. O tempo elevado para a execução dos procedimentos operacionais da biblioteca tem dificultado bastante a sua administração, requerendo um grande número de funcionários. Os tempos de atendimento dos usuários da biblioteca têm ficado cada vez maiores, o que tem causado insistentes protestos dos alunos e professores que utilizam a biblioteca [1].

2 Análise do Projeto

2.1 Objetivos

2.1.1 Objetivo Geral

- Estender o escopo da informação e a cultura aos leitores.

2.1.2 Objetivo Específico

- Desenvolver um sistema com capacidade de suportar grande quantidade de informação.
- Permitir uma gestão eficiente de usuários em todos os níveis.
- Permitir uma gestão eficiente de todos os tipos de material bibliográfico.

2.2 Requerimentos

2.2.1 Requerimentos Funcionais BackOffice

- | | |
|--------|--|
| RBO1. | Permitir cadastrar novos usuários da biblioteca de forma mais flexível, facilitando a alteração dos dados dos usuário. |
| RBO2. | Manter atualizado o cadastro de usuários da biblioteca. |
| RBO3. | Permitir cadastrar novos títulos e obras de forma eficiente. |
| RBO4. | Manter atualizado o cadastro das obras disponíveis destes títulos e o seu estado. |
| RBO5. | Manter o controle das multas dos usuários. |
| RBO6. | Controlar o empréstimo de obras. |
| RBO7. | Permitir reservas via internet ou locais para os usuários cadastrados. |
| RBO8. | Gerar relatório de material bibliográfico. |
| RBO10. | Gerar estatísticas de uso da biblioteca. |
| RBO11. | Permitir a autenticação dos usuários. |

RBO12. Permitir a configuração os parâmetros do sistema.

2.2.2 *Requerimentos Funcionais OnLine*

- ROL1. Permitir a autenticação dos usuários.
ROL2. Permitir cadastrar novos usuários da biblioteca de forma mais flexível, facilitando a alteração de dados do usuário.
ROL3. Permitir pesquisas remotas via internet usando computadores pessoais dos usuários.
ROL4. Fazer reserva de Material Bibliográfico.

2.3 Diagrama de Casos de Uso



Figura # 1. Diagrama de casos de uso do Sistema Gestão de Biblioteca

2.4 Diagrama de Classes

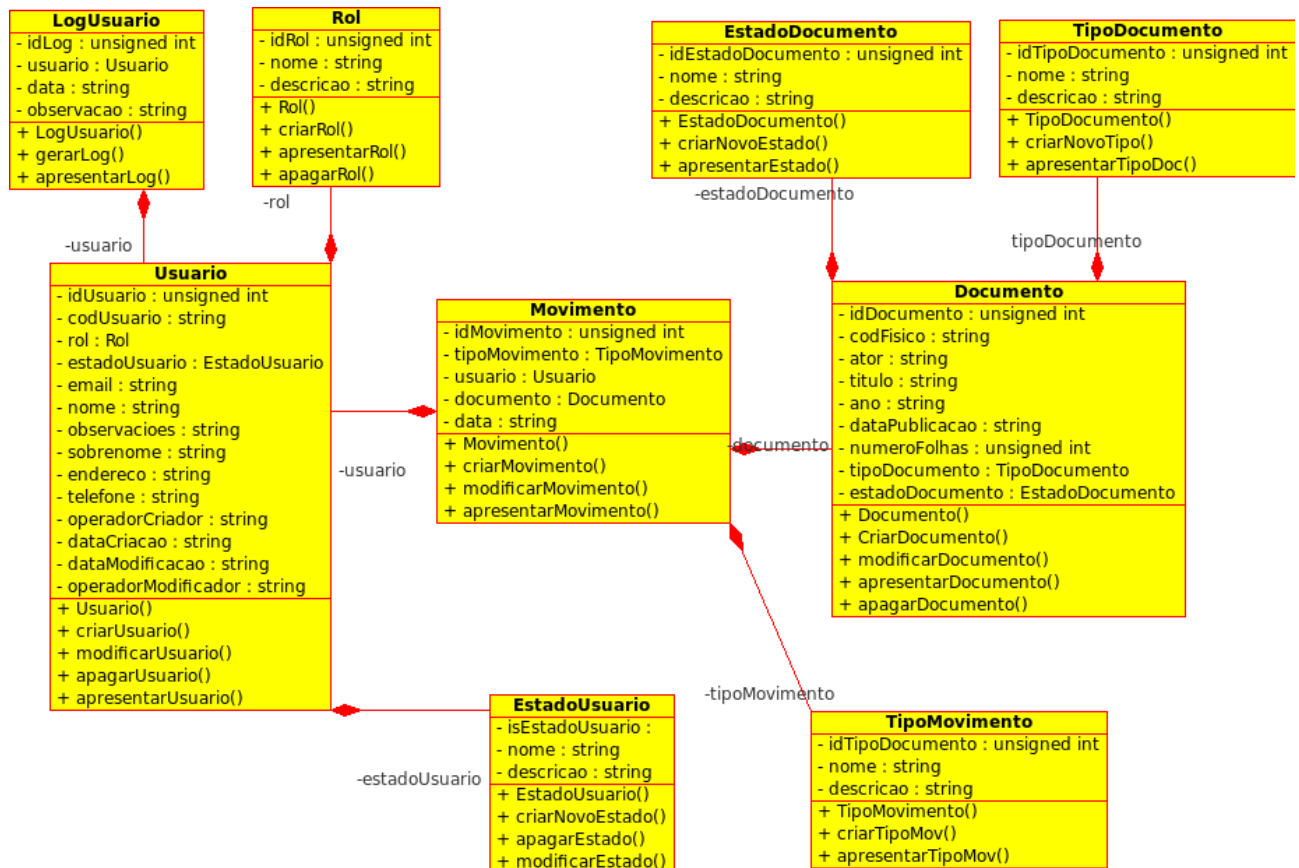


Figura # 2. Diagrama de classes do Sistema Gestão de Biblioteca

3 Arquitetura

3.1 Distribuição de Camadas

- Camada do cliente
- Camada Web(Dois módulos: Administrador(MistiBackOfficeStruts e MistiOnLine)
- Camada de Lógica de negócios(MistiBusiness)
- Camada de Persistência(MistiBusiness)

3.2 Diagrama da arquitetura

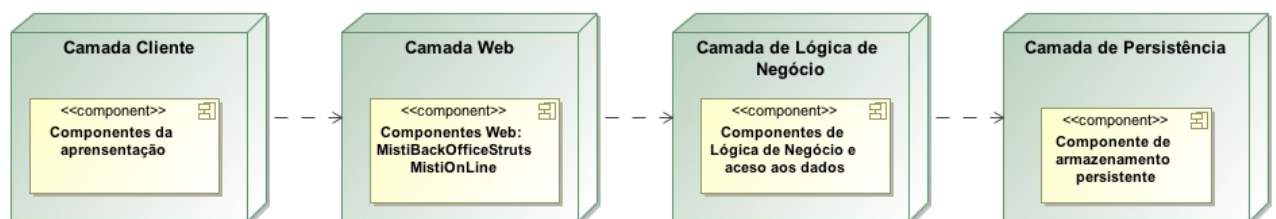


Figura # 3. Diagrama de arquitetura Gestão de Biblioteca

3.3 Especificação da arquitetura

3.3.1 Camada do Cliente

Em esta camada existem componentes que são executados no dispositivo do cliente(Web Browser), para fornecer o comportamento da interface do usuário. Como um exemplo do papel desta camada são:

- Apresentação de informações e mensagens.
- Ativação e seleção de navegação opções.
- Apresentação e campos para capturar informações.
- Realização de eventos.

3.3.2 Camada Web.

Em esta camada existem os componentes que são executados no interior do servidor de aplicativos Web compatível com a especificação J2EE. as principais funções deste camada são:

- Desenvolver o estado de interação com os clientes http, receber as requisições e gerar respostas para cada pedido.
- Invocar os serviços da camada de lógica do negócio.
- Gerenciar o fluxo de navegação nas aplicações usando um ponto de controle centralizado.
- Nesta camada se faz uso do modelo vista controlador(MVC) com Struts e JSF. O projeto *MistiOnLine* esta em JSF e o projeto *MistiBackOfficeStruts* com Struts.

3.3.3 Camada da Lógica do negócio.

As principais funções desta camada são:

- Apresentar de forma abstrata os serviços do negócio às diferentes necessidades da capa de apresentação.
- Definir mecanismos para invocar de forma síncrona ou assíncrona os serviços do negócio.
- Definir a lógica própria do negócio do sistema.
- Administrar o tratamento de transações.

3.3.4 Camada de Persistência.

As principais funções desta camada são:

- Definir e modelar as classes para persistir os dados do banco do dados.
- Definir mecanismos o aceso aos dados do sistema.
- Permitir as operações CRUD.

As camadas (Camada de Lógica de Negócio e de Persistência) são implementadas no projeto *MistiBusiness*.

4 Justificações da escolha das tecnologias

4.1 Ajax, Json e JQuery

O uso deles é fundamental no sistema, tanto para as requisições assíncronas com *ajax* quanto as validações dos campos dos formulários, a troca dos dados são feitas com *Json*.

- A gente fez uso de *jQuery JavaScript Library v1.3.2*[2].
- Para as validações a gente fez uso do Plugin *jQuery validation plug-in 1.5.2*[3].
- Para as requisições em *ajax* e *json* a gente fez uso de a função *\$.getJSON* de *Jquery*.

4.2 JSF

O uso dele foi fundamental para o desenvolvimento do projeto *MistiOnLine*. Os JSF[4] têm muitas vantagens no desenvolvimento Web:

- É um *framework* de desenvolvimento muito flexível.
- Com possibilidades de criação de componentes personalizados para o cliente.

4.3 Struts

O uso dele, ajudou muito no desenvolvimento do projeto *MistiBackOfficeStruts*, a gente fez uso do *Struts2*[5].

- Em comparação com *Struts 1.x*, é mais simples o desenvolvimento e tempo de aprendizado.
- A gente fez uso do *Framework Tiles* para a apresentação das telas.
- *Struts2* fornece o *Dojo*.

4.4 Spring

O uso de *Spring*[6] foi fundamental para a união dos projetos mediante o uso de Injeção de dependências.

- Nos projetos Web(*MistiOnLine* e *MistiBackOfficeStruts*) se chamavam aos Beans definidos no arquivo de configurações de *Spring*, assim, a gente instanciava classes da lógica de negócio
- Além disso, a gente faz uso do suporte oferecido por *Spring* para seu acoplamento com outros *frameworks*, neste caso *Hibernate*.

4.5 Hibernate

Foi desenvolvida no projeto *MistiBusiness*. A tecnologia foi fundamental para a persistência dos dados.

- Foi simples criar arquivos de mapeamentos baseados em *XML*.
- As queries foram feitas na linguagem HQL do *Hibernate*.
- A gente fez uso da semântica orientada a objeto transações.

4.6 Junit e Selenium

O uso dele foi fundamental para testar o desenvolvimento da lógica de negócio.

- Cada função foi testada com *Junit* para verificar se os dados obtidos foram corretos. Além disso a gente fez testes com *Selenium* para testar o comportamento das telas no projeto *MistiBackOfficeStruts*.

4.7 Web Services

A gente pesquiso para o correto uso deles no projeto, mas foi tarde demais, quando encontro uma muito boa utilidade.

- A gente desenvolveu um Web Service para obter o detalhe do documento.
- A *WSDL* oferecida pela *Amazon*[7], é muito flexível e ajuda muito para pesquisar material bibliográfico, os custos, etc. Mas não foi desenvolvida no projeto.

4.8 Rest e Mashups

A gente desenvolveu um mashup usando a API do *googlemaps*[8].

Além disso, como foi falado, na parte do *Web Services*, a *WSDL* da *Amazon*, é oferecida tanto para SOAP quanto para REST.

4.9 Ruby on Rails

A gente debateu sobre o uso de *Ruby on Rails*, mas desistiu, pelo pouco conhecimento da linguagem.

5 Funcionalidades referentes a Web 2.0

5.1 Predições com Weka

A gente desenvolveu predição dos números de logins e a idade, fazendo uso de Weka[9]. Os passos de como ele trabalha são:

- Na tela o funcionário ou administrador preenchem a idade para a predição.
- Pega do banco do dados, todos os usuários com rol de tipo *Usuário*.
- Criamos os atributos selecionados.
- Pega o algoritmo para treina-o. O algoritmo usado é a rede neural *Radial Basis Function* implementado em Weka[10].
- Se faz a predição com a rede treinada e o dado preenchido pelo usuário.
- Apresentação dos resultados.

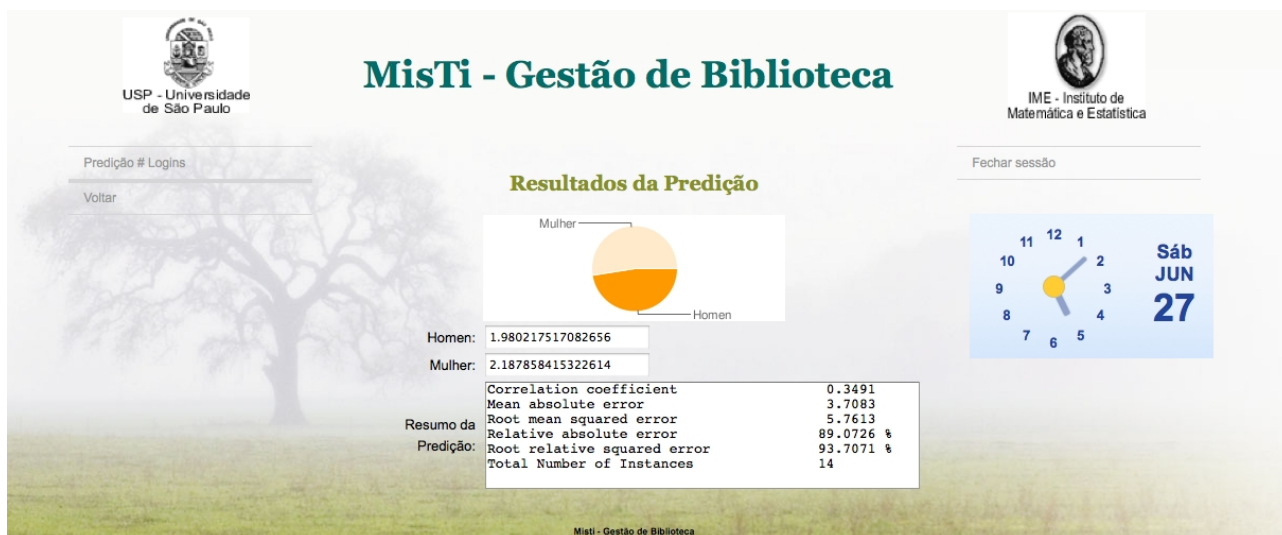


Figura # 4. Resultados da Predição usando Weka

5.2 Recomendações

Foi desenvolvido para realizar recomendações dos documentos da biblioteca aos usuários de acordo com os graus de similaridade com outros usuários, além disso, os usuários podem avaliar documentos para fazer recomendações mais precisas.

O processo consiste dos seguintes passos:

- Os usuários ao fazer alguma reserva de documentos, podem realizar avaliações do documento (na escala de 1-5), assim o usuário fornece ao sistema de suas opiniões.
- O sistema compara essas avaliações de outros usuários com gostos parecidos;
- Baseando-se nos gostos dos usuários o sistema sugere os documentos de outros usuários que tenham maior grau de similaridade com o usuário.

5.3 Busca e indexação de texto

Foi desenvolvido para obter documentos relevantes para uma consulta do usuário, a análise do texto de um livro permite conhecer os termos mais representativos de um documento, e também serve para ordenar os resultados em ordem de relevância.

Bem-vindo a Misti Erika

MISTI Gestor de Bibliotecas, ingresse sua solicitação de pesquisa, se deseja reservar um livro por favor registre-se.

Pesquisa Documento usando LUCENE

Autor:

Titulo:

Descricao:

Se encontro 1 documento.

Autor	Título	Código	Reservar	Visualizar Detalhes
José de Alencar	Senhora	MAC5776	Reservar	Visualizar Detalhes

*** Livros recomendados para leitura do usuário**

Autor	Título	Código	Reservar	Visualizar Detalhes
Erika Guetti	Aos anos de luta	MAC0003	Reservar	Visualizar Detalhes
Edith Zaida	Os homens são do Marte	MAC0002	Reservar	Visualizar Detalhes
Ginha Nader	Um século de sonho	MAC5778	Reservar	Visualizar Detalhes
Deitel and Deitel	Java	MAC0005	Reservar	Visualizar Detalhes

Figura # 2. O sistema apresenta a lista dos livros recomendados para o usuário, além dos resultados de uma consulta.

6 *Grau de participação dos integrantes*

6.1 Alvaro Mamani Aliaga

- Ele fez o projeto de MistiBackofficeStruts.
- Trabalhou o desenvolvimento da parte de Servlets.
- Trabalhou a parte de ajax, json, jquery, Struts e Mashups.
- Desenvolveu a parte de predição com Weka.

6.2 Edith Sonco Mamani

- Ela fez o projeto de MistiBusiness.
- Trabalho tanto a lógica de negócio quanto a persistência dos dados(Spring e Hibernate).
- Ela fez os testes com Junit e Selenium e desenvolveu em Ruby on Rails.
- Desenvolveu a parte de recomendação pela similaridade dos usuários.

6.3 Erika Guetti Suca

- Ela fez o projeto de MistiOnLine.
- Trabalhou a parte de JSF e WebServices.
- Desenvolveu a parte de busca e indexação de documentos.

As tarefas foram desenvolvidas quase equitativamente, desempenhando um bom papel entre os integrantes. Na primeira fase da disciplina os integrantes apresentaram duvidas e dificuldades para adaptar-se, mas pouco a pouco as coisas foram certas.

7 *Aproveitamento da disciplina*

7.1 Alvaro Mamani Aliaga

No começo, as coisas se tornaram muito difíceis, tanto no desenvolvimento dos trabalhos quanto o idioma, mas pouco a pouco a gente se foi adaptando.

A gente tinha pouca experiência no desenvolvimento web, isso ajudou um pouco, mas os tópicos foram muito interessantes, e a experiência tornou-se pouca.

A disciplina ajudou muito a compreender todas as coisas que a gente pode fazer no desenvolvimento web, forneceram muitos conceitos como por exemplo ajax, json. Além deles, fazer testes nas telas com Selenium, e testes com junit. Mesmo que o uso de Mashups.

A gente não tinha conhecimento sobre como aplicar inteligência coletiva o uso dela na web, isso é o ponto mais grosso.

7.2 Edith Sonco Mamani

Durante o andamento da disciplina foram estudadas questões muito interessantes sobre o desenvolvimento para Web 2.0, essas tecnologias proporcionam uma abordagem abrangente para a implementação de sistemas robustos e confiáveis.

O desenvolvimento de algoritmos de inteligência coletiva em nosso projeto permitiu fazer recomendações a usuários, previsões e pesquisas de itens para tornar-o mais atrativo.

Sobre o lado negativo da disciplina, eu acho que as avaliações dos trabalhos por diferentes grupos não foram muito equitativas, em alguns casos se vieram notas muito altas e outras baixas.

7.3 Erika Guetti Suca

Na minha opinião o positivo é:

Encontraram-se aplicabilidade muito interessante nos algoritmos de inteligência coletiva para problemas do dia a dia na internet.

O negativo:

A disciplina não esteve orientada para as pessoas que desconhecem os princípios de desenvolvimento web.

8 Referências

- [1] MasterSoft
http://www.mastersoft.com.br/Tutoriais/Mastermodel/LevantamentoRequisitos/Artefatos/esp_req50_01.PDF
- [2] jQuery JavaScript Library v1.3.2
<http://jquery.com/>
- [3] *jQuery validation plug-in* 1.5.2
<http://bassistance.de/jquery-plugins/jquery-plugin-validation/>
- [4] Java Server Faces
<http://java.sun.com/javaee/javaserverfaces/>
- [5] Struts 2.0
<http://struts.apache.org/2.x/index.html>
- [6] Spring
www.springframework.org
- [7] Amazon WSDL
<http://soap.amazon.com/schemas2/AmazonWebServices.wsdl>

- [8] GoogleMaps
<http://code.google.com/intl/pt-BR/apis/maps/>
- [9] Waikato Environment for Knowledge Analysis
<http://www.cs.waikato.ac.nz/ml/weka/>
- [10] ALAG, S.
Collective Intelligence in Action, ISBN 1933988312, 2009.