

# Sistema web para biblioteca da empresa Marluvas Calçados de Segurança

Ângelo Rafael da Silva<sup>1</sup>, Rodrigo Geoffroy<sup>1</sup>, Nairon Neri Silva<sup>2</sup>

<sup>1</sup>Universidade Presidente Antônio Carlos - Departamento de Ciência da Computação (UNIPAC)

<sup>2</sup>Universidade Presidente Antônio Carlos - Departamento de Administração (UNIPAC)  
Rua Palma Bageto Viol S/N – Barbacena – MG – Brasil

silva-rafael91@hotmail.com, rodrigotgeoffroy@bol.com.br, naironneri@gmail.com

**Resumo.** O presente artigo apresentará o desenvolvimento de algumas funcionalidades de um sistema web para a biblioteca da Empresa Marluvas calçados de Segurança na cidade de Dorés de Campos, MG onde serão identificados os principais problemas no gerenciamento e propor soluções para a otimização da rotina do bibliotecário e do controle do acervo da biblioteca da empresa. Procurando melhorar o funcionamento da biblioteca da empresa Marluvas Calçados de Segurança Para alcançar tal objetivo, será realizado uma pesquisa de campo e levantamentos bibliográficos sobre as tecnologias utilizadas. Para o desenvolvimento do sistema, será utilizada a tecnologia java 2EE e alguns frameworks, dentre eles, JSF e Hibernate.

**Abstract.** The present paper intends to demonstrate the preparation of the environment for the development of a web application for a library and the execution system. In order to improve the library operation at the company Marluvas Calçados de Segurança Ltda, in Dorés de Campos, MG state, it'll be identified the main problems on the management and suggested solutions for the management and to the optimization of the library routine and the collection control. To meet this aim it'll be done a field research and bibliographic studies about the technologies used. In order to develop the system it'll be used the Java 2EE technology and some frameworks, such as JSF and Hibernate, among others.

## 1. Introdução

No cenário da tecnologia atual, é preciso ter acesso às informações de forma rápida e precisa para a tomada de decisões sobre nossas tarefas rotineiras. As informações tornaram-se importantes para o nosso dia a dia, modificando o modo como trabalhamos e interferindo em nossas ações, com o volume de dados é necessário o uso de sistemas de informação, que permitem o acesso a essas informações. O crescimento e a facilidade de acesso à *internet* possibilita desenvolver sistemas que rodam em servidores na *internet*. Há muitas vantagens em optar por sistemas web, dentre elas destaca-se a facilidade de acesso, permitindo o acesso em qualquer dia, lugar e hora desde que haja acesso à *internet*, independente do sistema operacional do cliente. Segundo [ALGAWORKS SOFTWARES E TREINAMENTOS], aplicativos *web* são implementados com base em tecnologias padronizadas como o *Hypertext Markup language* (HTML) e o protocolo *Hypertext Transfer Protocol* (http), que são reconhecidos e utilizados em

várias plataformas, na máquina do cliente é necessário ter instalado somente um aplicativo *browser*, ou seja, navegador de *internet*.

As bibliotecas trabalham com grandes quantidades de informação por essa razão, é preciso um *software* para gerenciar essas informações de forma que os profissionais da biblioteca possam ter acesso fácil a elas. Os sistemas são responsáveis por solucionar problemas relativos ao controle de empréstimos de livros, à data de devolução, aos atrasos, dentre outros.

As bibliotecas têm feito uso da tecnologia da informação, desde a passagem dos manuscritos para a utilização de textos impressos, o acesso a bases de dados bibliográficas armazenadas nos grandes bancos de dados, o uso do *CD-ROM* e o advento das bibliotecas digitais. O uso das diversas tecnologias da informação demonstra, nos últimos 150 anos, que as bibliotecas vêm tentando acompanhar novos paradigmas tecnológicos. [Kleinubing 2006]

Conforme [Kleinubing 2006], os bibliotecários, não podem ignorar a presença do computador e das tecnologias, que procuram facilitar a vida dos usuários e profissionais, aumentando a qualidade de seus produtos e serviços.

De acordo com [Saraiva 2011], vivemos a Era da Informação e o fato de existirem bibliotecas sem automação começa a levantar discussões sobre o papel da informática nesse cenário. Infelizmente, muitas instituições ainda não reconhecem a necessidade de investir nas bibliotecas, sendo que algumas sobrevivem apenas de doações. Outras não conhecem ou não têm acesso à informação.

O presente artigo tem por finalidade desenvolver as funcionalidades de cadastro de usuário, cadastro de autor, cadastro de editora, cadastro de classificação, cadastro de livro, reserva de livro, locação de livro e devolução de livro para uma aplicação *web* e melhorar o funcionamento da biblioteca da empresa Marluvas Calçados de

Segurança na cidade de Dores de Campos, Minas Gerais. Para isso, é preciso identificar os principais problemas de gerenciamento da biblioteca e propor soluções de gerenciamento para a otimização da rotina do bibliotecário e para o controle do acervo da biblioteca da empresa. O artigo se justifica pelo fato de que o avanço tecnológico e o crescimento da tecnologia da informação estarem contribuindo para o crescimento do gerenciamento de tarefas rotineiras que, antes eram realizadas de modo manual. Um exemplo disso é o controle de acervos de bibliotecas, o gerenciamento eletrônico pode facilitar o trabalho do profissional encarregado pela biblioteca, otimizando tal controle. Outro motivo, não menos relevante, é a aplicação, na prática, do aprendizado adquirido durante o curso de ciência da computação, buscando, com isso, a percepção de como tal aprendizado pode mudar a situação de um ambiente real.

A empresa utiliza uma planilha para consulta dos livros que a biblioteca possui, disponibilizada pelo funcionário responsável pelo controle do acervo, e cadernos para controle de locação. Conta com três funcionários de outros setores para o atendimento e funcionamento da biblioteca, não existe funcionários com somente a função de bibliotecário. O atendimento acontece em uma sala que contém quatro armários onde são guardados os livros. Para alcançar os objetivos propostos, será realizado um levantamento bibliográfico contemplando as principais tecnologias utilizadas no desenvolvimento do sistema bibliotecário e pesquisa de campo na empresa com a aplicação real do sistema.

## 2. Referencial bibliográfico

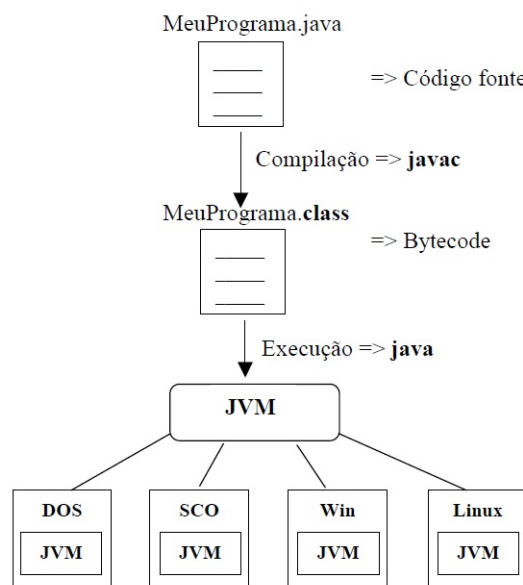
Nesta seção, serão abordadas as principais tecnologias usadas para alcançar os objetivos proposto no artigo.

### 2.1. Java

*Java* é uma linguagem de programação de alto nível que utiliza o paradigma orientado a objetos, aproximando-se da linguagem humana. É fortemente tipada, bem estruturada, segura, dinâmica e robusta. Possui uma ampla variedade de bibliotecas, que contem códigos, classes, métodos e atributos prontos, disponibilizado para utilização, para facilitar e tornar mais rápido o desenvolvimento de sistemas e organizada, sem precisar ficar refazendo o que já está pronto.

Segundo [Deitel 2005], a *Sun Microsystems* financiou um projeto de pesquisa com o nome Green, em 1991, que resultou em um desenvolvimento de uma linguagem baseada em C++ e o criado James Gosling deu o nome de Oak em homenagem a uma árvore de carvalho, mas, descobriu-se que já existia uma linguagem com esse nome, quando uma equipe da Sun visitou a uma cafeteria local o nome java foi sugerido (java é uma cidade de origem de um tipo de café importado) e, a partir de 1995, estourou no mercado, após o anúncio feito pela Sun em uma conferência importante e, segundo [Moreira 2009], foi vendida para Oracle em 2009. Conforme [Braz], em uma linguagem de programação, como, por exemplo a linguagem C, o código fonte é compilado para um determinado sistema operacional. Em Java, o código fonte é compilado para a máquina virtual java (*JVM Java Virtual Machine*). A portabilidade do Java está ligada a JVM que entende o *bytecode* que é gerado por um compilador java, como o *javac*. Esse *bytecode* serve para diferentes sistemas operacionais, já que será interpretado pela JVM para o sistema operacional. Dessa forma, o código escrito pelo programador não precisa se adequar às chamadas de sistema e às outras funções do sistema operacional, quem faz esse papel é a JVM.

De acordo com [Cardim 2010], os sistemas operacionais possuem *application programming interface* (API) diferentes uns dos outros, por exemplo, a API de *input/output* (I/O) do windows 7 é diferente do Linux, fazendo com que, segundo [Braz], código fonte, escrito para uma plataforma, seja modificado em algumas partes para o outro sistema, a JVM é uma camada entre o *bytecode* e o SO. A figura 1 apresenta o esquema de compilação, execução e portabilidade de um programa Java, e ilustra o funcionamento explicado no texto acima.



**Figura 1. Esquema de compilação, execução e portabilidade de um programa Java. [Braz]**

O Java tornou-se a linguagem preferida para implementar aplicativos baseados na *internet* e *software* para dispositivos que se comunicam em uma rede. Ele não é mais utilizado simplesmente para tornar as páginas da *World Wide Web* mais dinâmicas tornou-se a linguagem preferida para atender às necessidades de programação corporativas de muitas empresas.[Deitel 2005]

## 2.2. JavaServer Faces

As páginas *web* estão se tornando cada vez mais atrativas, repletas de recursos e interativas com os usuários. As aplicações ficam hospedadas em servidores na *internet* e os usuários precisam ter somente um *browser* (navegadores). Segundo [ALGAWORKS SOFTWARES E TREINAMENTOS], para ter acesso às páginas, é utilizado um modelo de request-response (requisição e resposta), o cliente envia uma requisição *Hypertext Transfer Protocol* (HTTP) para o servidor na *web* e o servidor responde para o cliente. No Java o início desse modelo foi pela API de *Servlets* e *Java server pages* (JSP), e os servidores *web* são conhecidos como *Servlet Container*. Quando o cliente faz a requisição ao servidor, ele a converte em um objeto do tipo *HttpServletRequest*, o objeto é passado para os componentes *web*, podendo interagir com o banco de dados ou *JavaBeans*, em seguida é gerado um conteúdo dinâmico e transformado em um objeto *httpServletResponse* que é enviado ao cliente.

O *JavaServer Faces* (JSF) é um *framework* para desenvolvimento de aplicações *web* e é baseado no padrão MVC (*Model-View-Controller*), dividindo em três camadas, separando a lógica de negócio, a apresentação e a persistência dos dados. Segundo [ALGAWORKS SOFTWARES E TREINAMENTOS], A camada de modelo representa a lógica de negócio, mantém o estado da aplicação e disponibiliza para o controle o acesso aos dados. A camada de visão é responsável pela exibição dos dados e enviar as ações do usuário para o controle. A camada de controle é responsável por fazer a ligação entre a camada de visão e a camada de modelo recebe as ações do usuário, envia para a camada

modelo e, após o processamento, repassa os dados para a camada de visão. Conforme [ALGAWORKS SOFTWARES E TREINAMENTOS], na tecnologia JSF, o controle é feito por um servlet, chamado *Faces Servlet*, configurado através de arquivos *extensible markup language* (XML) que contem informações sobre o mapeamento de ações e regras de navegação. Observadores de eventos e manipuladores de ações recebem os dados da camada de visão, acessam o modelo e devolvem o resultado para o cliente por meio de *Face Servlet*, o *Faces Servlet* é quem recebe as requisições dos usuários e encaminha para a camada de modelo. A camada de modelo é representada por objetos de negócio, que executam uma lógica de negócio ao receber dados da camada de visão.

Segundo [ALGAWORKS SOFTWARES E TREINAMENTOS], JSF possui componentes que podem ser apresentados de diversas formas para vários clientes, como celular e *browser* da *web*. Permitem também a utilização de componentes plugáveis que permitem que o mesmo componente se apresente de formas diferentes, dependendo do tipo de cliente e do que for adequado no momento.

Para executar aplicações criadas em JSF é preciso um servidor com um *container web* (servidor *web* com funcionalidade para executar o código Java conforme especificações Java EE) instalado que recebe as requisições HTTP, executa componentes Java e retorna para o cliente o código HTML. JSF revolucionou o desenvolvimento *web* e agora está em grande expansão. [ALGAWORKS SOFTWARES E TREINAMENTOS]

### 2.3. MySQL

MySQL é um sistema gerenciador de banco de dados *Open Source* com base na GPL (*General public license*), desenvolvido e distribuído pela corporação Oracle. De acordo com [Oracle 2011], É um sistema gerenciador de banco de dados (SGBD) relacional com suporte a linguagem SQL (*Structured Query Language* Linguagem Estrutural de Consultas).

Segundo [Oracle 2011], SQL é a linguagem mais utilizada para acessar banco de dados e é definida pelo Padrão ANSI/ISO SQL, devido a sua popularização organizações como ANSI e ISO resolveram padronizá-la. Possui um bom desempenho, estabilidade e confiabilidade, e é amplamente utilizada em servidores web. O MySQL fornece suporte a várias plataformas e linguagens, como PHP, JAVA, .NET e *Visual Basic*.

Muitas das atuais grandes organizações da internet, como Google, Facebook e Adobe, utilizam MySql para gerenciar grandes quantidades de dados e aplicações críticas. [Silva 2011]

### 2.4. Hibernate

O processo de mapeamento de objetos em Java, para as tabelas do banco de dados, é chamado mapeamento objeto-relacional (ORM). Para realizar o mapeamento, é utilizado *Java Persistence API* (JPA), que permite ao desenvolvedor armazenar, atualizar e recuperar dados do banco de dados relacional para objetos Java com maior facilidade e menor tempo de desenvolvimento, trabalhando diretamente com objetos e não instruções SQL.

O Hibernate é um *framework open-source* de mapeamento objeto relacional, ferramenta ORM. É bastante poderoso e dá suporte ao mapeamento de associações entre objetos, herança, polimorfismo, composição e coleções [King]. Disponibiliza um meca-

nismo para consulta de dados, o que facilita e reduz o tempo de desenvolvimento. Utiliza arquivos de configurações XML e, em algumas versões mais atuais, permite que o desenvolvedor Java utilize *Annotations* (Anotações) no próprio código para fazer o mapeamento.

Com o Hibernate, ganhamos muito em agilidade e tornamos nossos produtos muito mais flexíveis, fica muito fácil trocar o banco de dados inteiro com quase nenhum esforço [King].

## **2.5. UML**

A *Unified Modeling Language* (UML) é uma linguagem de modelagem livre que permite o desenvolvedor visualizar o produto em diagramas padronizados. Segundo [Melo 2010], a UML não é uma metodologia de desenvolvimento; não diz o que fazer primeiro, ou como projetar o sistema, mas auxilia na visualização do desenho do *software* e na comunicação entre objetos.

### **2.5.1. Diagrama de caso de uso**

O diagrama de caso de uso tem o objetivo de auxiliar a comunicação entre o analista e o cliente. Descreve o cenário que mostra as funcionalidades do sistema do ponto de vista do usuário. Apresenta as principais funcionalidades do sistema. O diagrama de caso de uso é representado por: atores, caso de uso e os relacionamentos entre eles.

### **2.5.2. Diagrama de classe**

O diagrama de classes está contido a uma categoria de diagramas estruturais da UML, que tem como premissa mostrar a estrutura que está proposta para funcionalidade/componente/aplicativo [Melo 2010].

Segundo [Melo 2010], o Diagrama de classes é um dos mais conhecidos e importantes, ajuda a compreender melhor a estrutura e funcionalidades do sistema, ele apresenta as classes dos sistemas, seus atributos e seus métodos.

### **2.5.3. Diagrama de sequência**

O diagrama de sequência tem o objetivo de mostrar como as mensagens entre os objetos são trocadas no decorrer do tempo para a realização de uma operação e representa a sequência de processos.

### **2.5.4. Diagrama de pacotes**

O diagrama de pacotes descreve os pacotes do sistema divididos em agrupamentos mostrando as dependências entre eles. Um pacote é um conjunto de elementos agrupados, esses elementos podem ser classes, diagramas ou até mesmo pacotes. O seu principal objetivo é de agrupar as classes em pacotes.

## 2.6. GlassFish

GlassFish é um servidor de aplicação open source, para a plataforma Java Enterprise edition (Java EE). Utilizado para o desenvolvimento de aplicações e serviços *web*. É um servidor de aplicação de código aberto.

## 3. Desenvolvimento

Esta seção aborda a preparação para o desenvolvimento do sistema web proposto e alguns diagramas gerados na fase de projeto, para construção do sistema foi utilizado o *framework* JSF com servidor GlassFish, Hibernate para tratar a persistência dos dados e banco de dados MySQL.

### 3.1. Sistema web para biblioteca

O sistema *web*, para gerenciamento de uma biblioteca compõem-se de rotinas para melhorar o funcionamento da biblioteca Marluvas Calçados de Segurança, onde o funcionário, por meio de seu login, que é o número do registro do funcionário na empresa, e senha consiga acesso fácil ao acervo, reserva de livros e renovação do tempo de locação do livro, e o bibliotecário, por meio de seu login e senha, possuam um controle maior sobre as informações pertinentes ao usuário e acervo, tais como consultar locação, devolução, reservas e verificar pendências.

As principais funcionalidades do sistema para o bibliotecário são:

1. Cadastrar usuário (funcionário da empresa);
2. Alterar dados do usuário;
3. Consultar dados, locação e devolução;
4. Excluir usuários;
5. Cadastrar livros;
6. Alterar dados do Livro;
7. Consultar os dados do livro e histórico de locação;
8. Exclusão de livros;
9. Efetuar locação e devolução;
10. Cadastrar Editora;
11. Consultar Editora;
12. Alterar Editora;
13. Apagar Editora.

### 3.2. Modelagem do Sistema web para biblioteca

Para construir o sistema *web* para biblioteca foram utilizados os seguintes diagramas UML, diagrama de caso de uso, diagrama de classes, diagrama de sequência, diagrama de pacote e Modelo Relacional (MR), isso permite maior compreensão do sistema.

#### 3.2.1. Diagrama de caso de uso

O sistema possui dois atores: o bibliotecário e o funcionário (Cliente); O bibliotecário é responsável pelos dados e pelos cadastros de usuário, livro, autor, editora e pela locação e devolução dos livros. O cliente acessa o sistema para consulta de seu histórico, de livros,

de fazer reservar e de renovação da data de entrega do livro. A figura 2, apresenta as funcionalidades do sistema e a responsabilidade de cada ator.

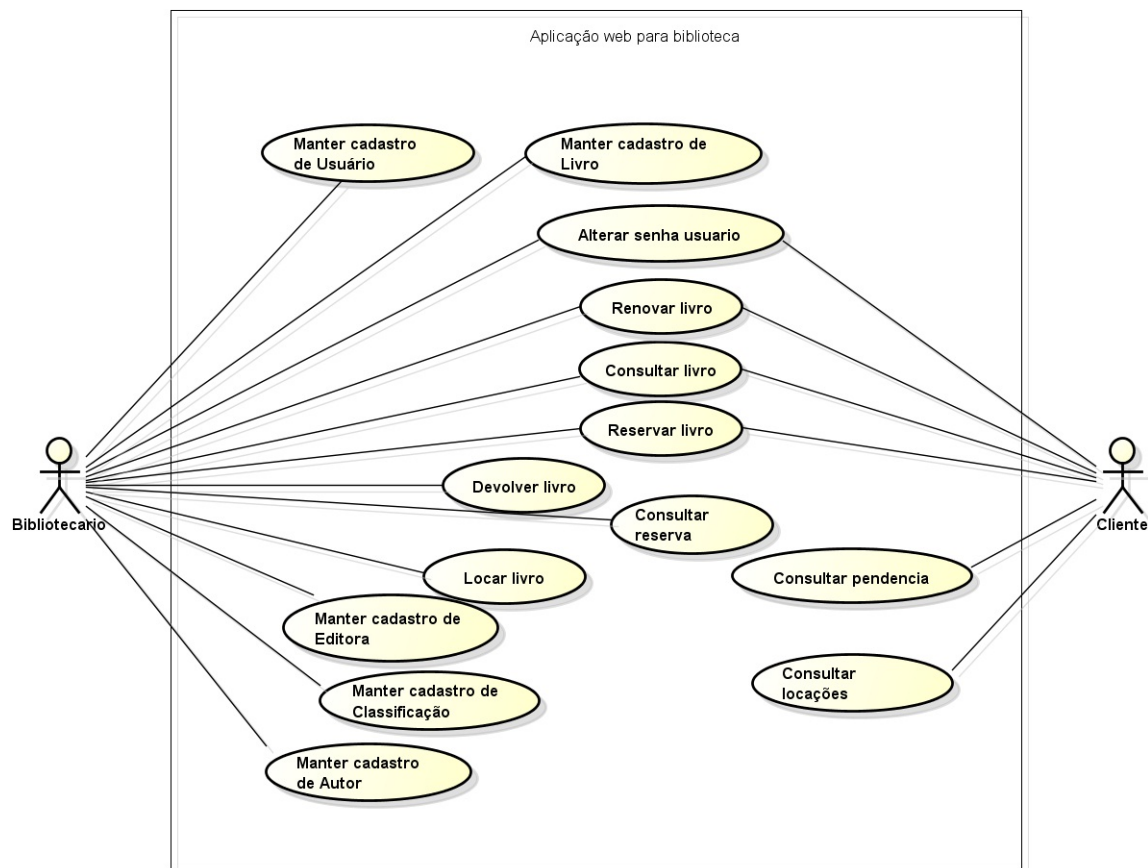


Figura 2. Diagramas de caso de uso.

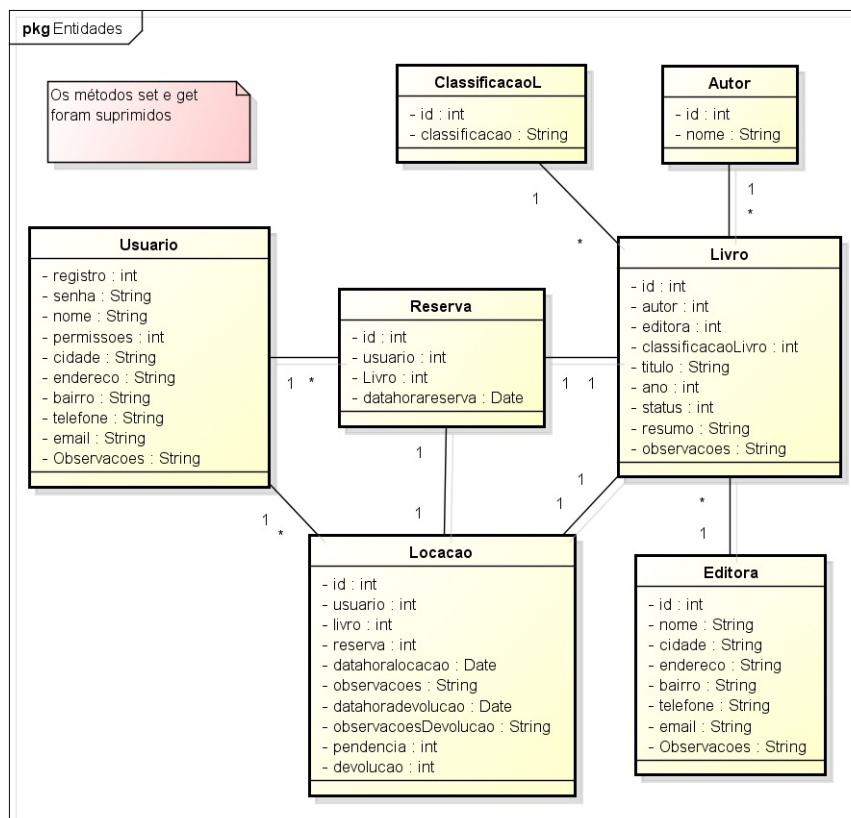
### 3.2.2. Diagrama de classes

As classes do pacote tcc.Entidades consistem na camada de modelo, responsável por fazer o mapeamento com o banco de dados, são nessas classes que estão os atributos, métodos de acesso *getter*<sup>1</sup> e *setter*<sup>2</sup> e as consultas aos dados e também realizam a persistência dos dados no banco. A figura 3, mostra o diagrama de classes do pacote tcc.Entidades.

<sup>1</sup>Usado para encapsular atributos, evitando que eles sejam acessados diretamente. Eles são utilizados para acesso aos atributos privados das classes.

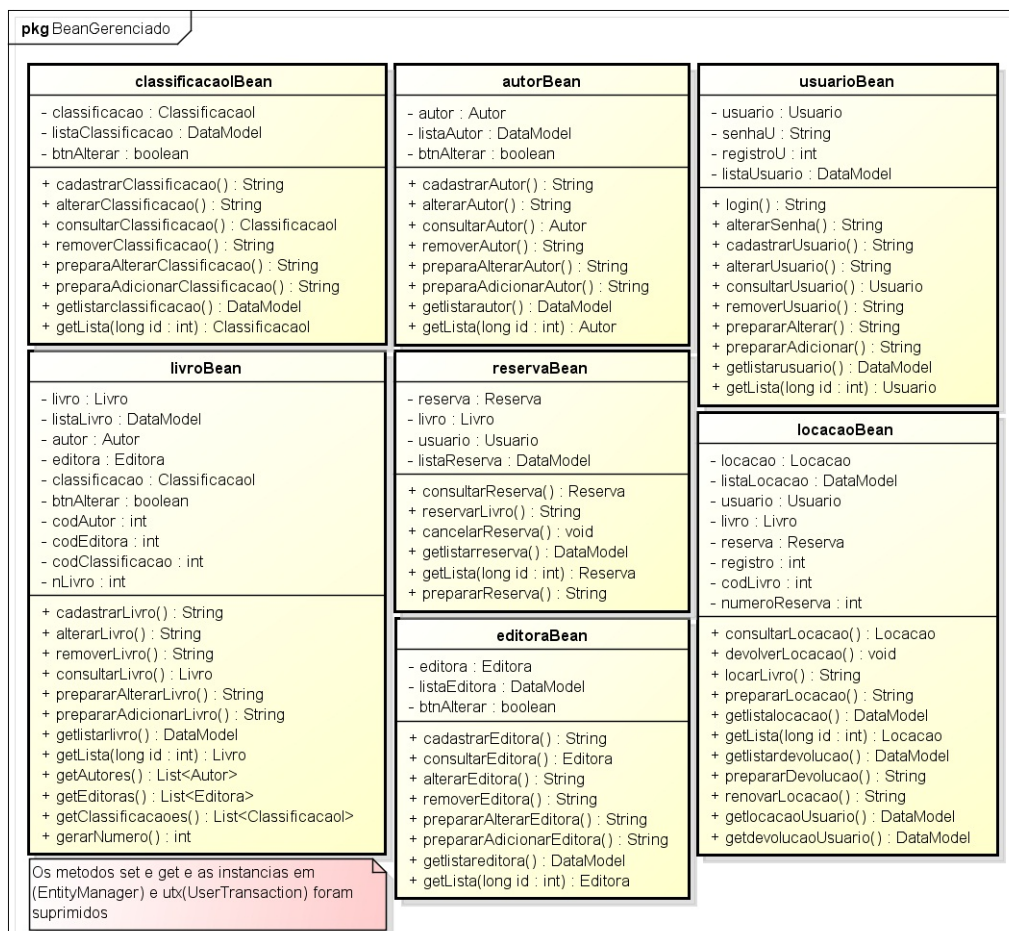
<sup>2</sup>Usado para encapsular atributos, evitando que eles sejam acessados diretamente. Eles são utilizados para acesso aos atributos privados das classes.





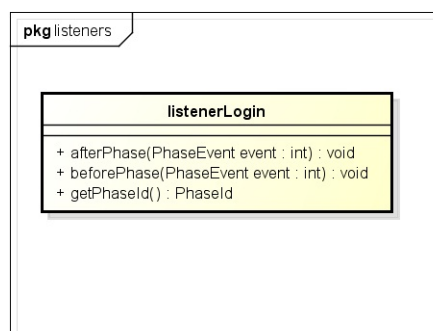
**Figura 3. Diagrama de classes do Modelo.**

As classes do pacote `tcc.BeanGerenciado` pertencem à camada de controle (regra de negócio), que é responsável pela ligação entre a camada de visão e a camada modelo (responsável pela persistência da aplicação). A figura 4, apresenta o diagrama de classes do pacote `tcc.BeanGerenciado`.



**Figura 4. Diagrama de classes do Controle.**

A classe listenerLogin do pacote tcc.listeners representa o ouvinte login responsável pela regra de acesso as páginas de acordo com a sua permissão, responsável por direcionar o usuário para as páginas de usuário e bibliotecário para as páginas de Adm e não permitindo que um usuário consiga acessar a parte de Adm do sistema. A figura 5, apresenta o diagrama de classes do pacote tcc.listeners



**Figura 5. Diagrama de classes do listeners.**

### 3.2.3. Diagrama de Sequência

O diagrama de sequência mostra como as mensagens entre os objetos são trocadas no decorrer do tempo para a realização da operação. Na figura 6, é apresentado o diagrama de sequência ilustrando a troca de mensagens quando se realiza o *login*, mostrando o caso de uso detalhado anteriormente.

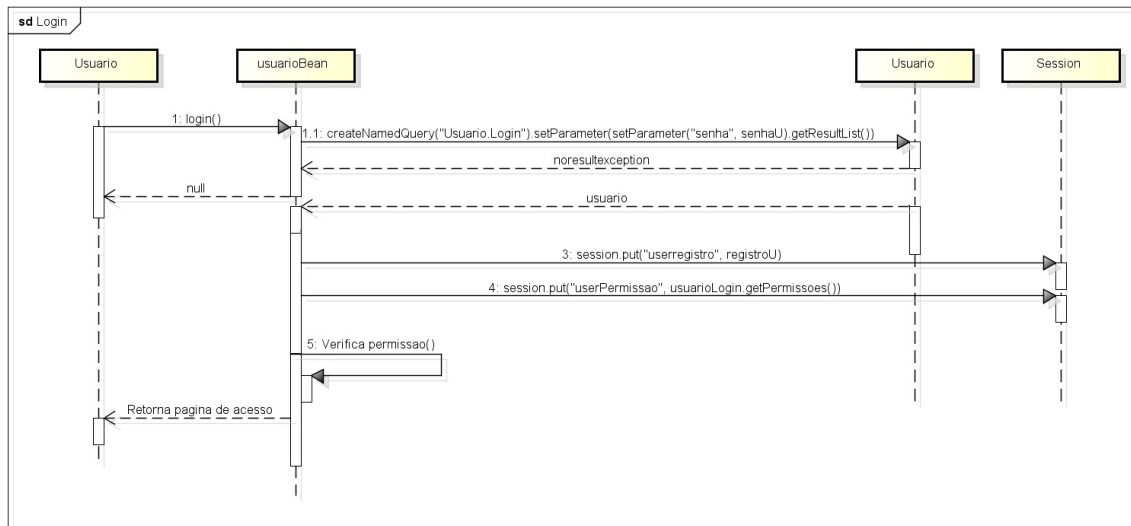


Figura 6. Diagrama de sequência: Login.

Na figura 7, apresenta o diagrama de sequência ilustrando a troca de mensagens quando se realiza a locação em caso de sucesso.

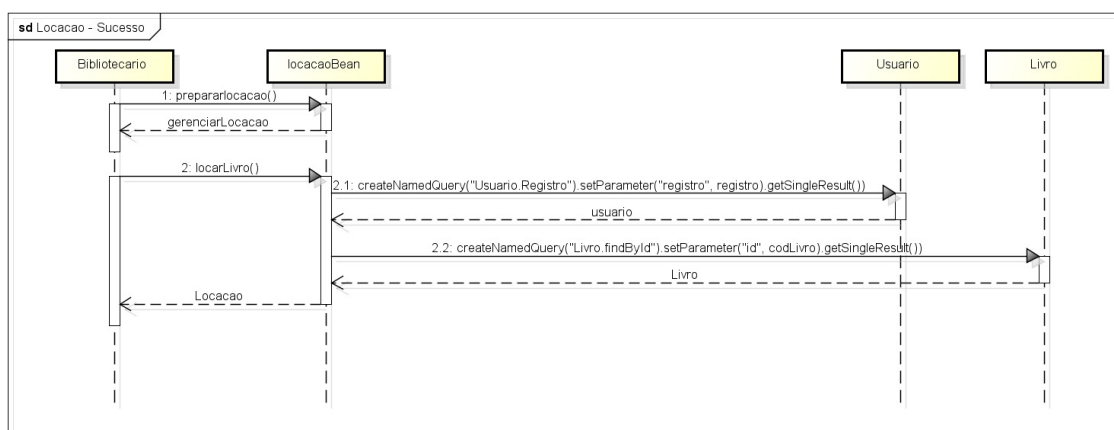
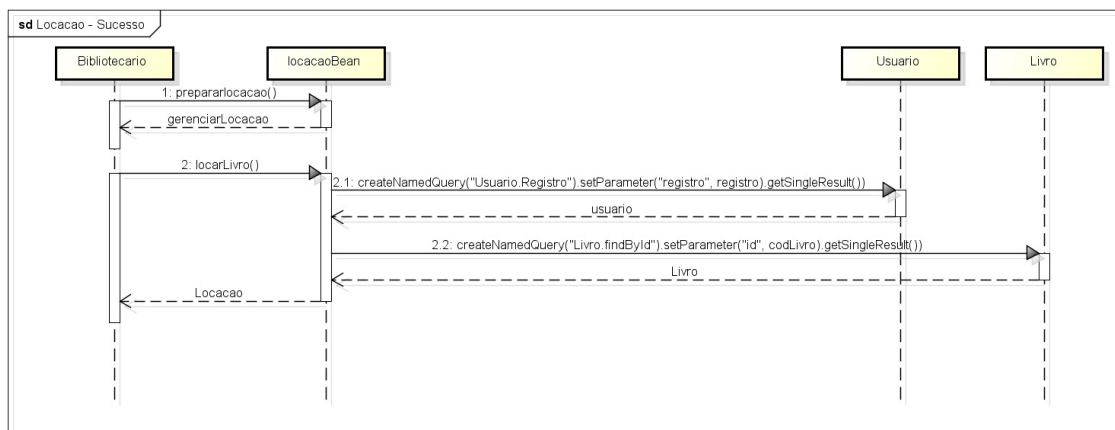


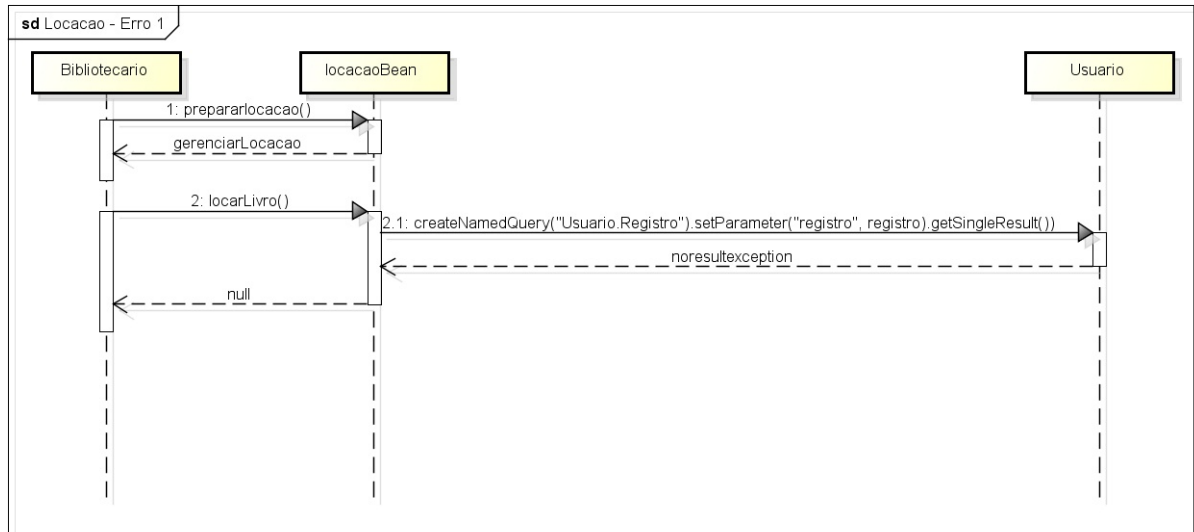
Figura 7. Diagrama de sequência: Locação.

Na figura 8, apresenta o diagrama de sequência ilustrando a troca de mensagens quando se realiza a locação para um usuário não cadastrado.



**Figura 8. Diagrama de sequência: Locação.**

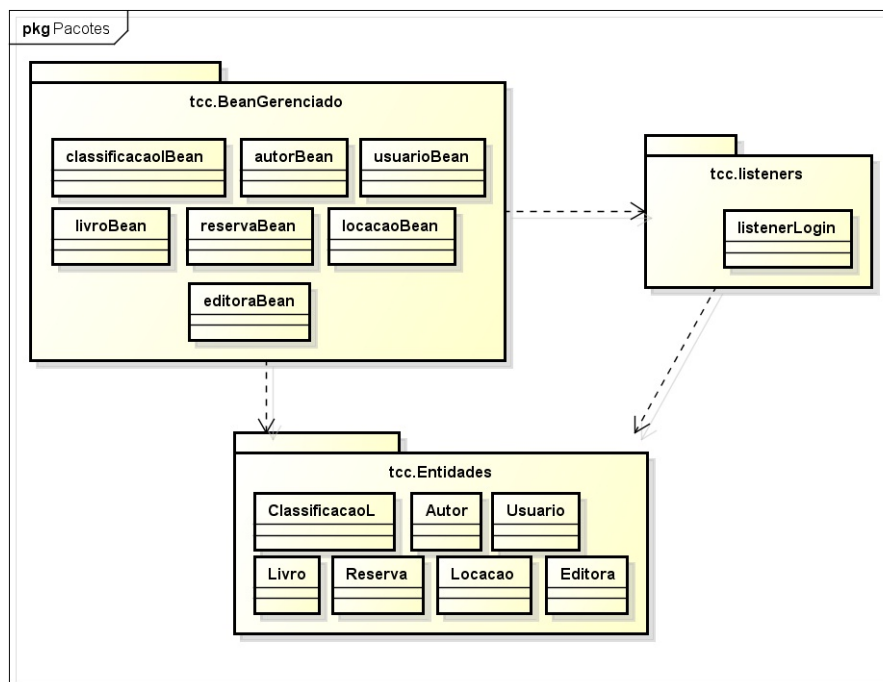
Na figura 9, apresenta o diagrama de sequência ilustrando a troca de mensagens quando se realiza a locação para um livro não cadastrado.



**Figura 9. Diagrama de sequência: Locação.**

### 3.2.4. Diagrama de Pacotes

O sistema divide em três pacotes conforme mostra a figura 10.



**Figura 10. Diagrama de pacotes.**

### 3.2.5. Modelo Relacional

O Modelo Relacional representa os dados em um banco de dados como um conjunto de tabelas, essas tabelas terão um nome, que é único, e um conjunto de atributos com seus nomes e conjunto de valores atômicos. O modelo de relacional pode ser visto na figura 11, pode ser visto o conjunto de tabelas com os seus nomes, atributos e valores.

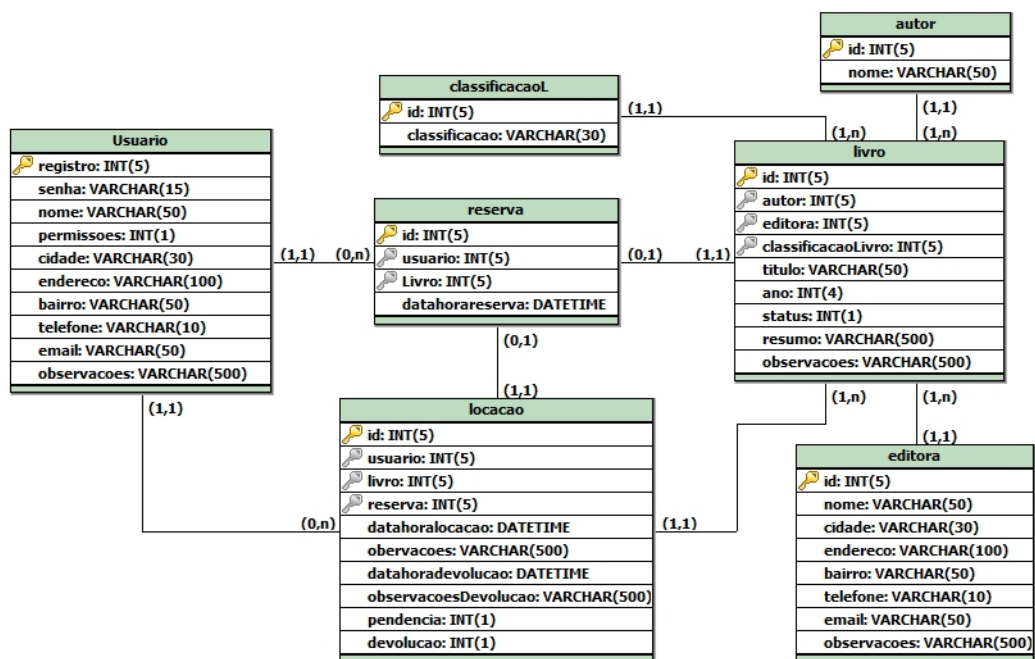


Figura 11. Modelo Relacional.

### 3.3. Páginas do Sistema

Para construção das paginas foi utilizado JSF, HTML e CSS. A página para o login pode ser vista na figura 12.

A imagem mostra a interface de login do sistema. No topo direito, há o logo da **MARLUVAS** com o slogan "Nosso negócio é proteção.". No centro, há um formulário de login com o título "Biblioteca Login". O formulário contém campos para "Login:" (com o valor "0" preenchido) e "Senha:", e um botão "Login" na base.

Figura 12. Tela de login do Sistema.

A figura 13, mostra a página do bibliotecário.





Figura 13. Tela de Administração do bibliotecário.

A figura 14, mostra a página do usuário.



Figura 14. Tela do usuário.

A figura 15, apresenta a tela para cadastrar o livro.

## Cadastrar Livro

Número:

9

Título:

Como se tornar um líder servidor

Autor:

James C. Hunter

▼

Editora:

Sextante

▼

Classificação:

Administração

▼

Ano:

Status:

☒ Disponível
 ☐ Alocado
 ☐ Reservado

Resumo

Observações

[Inserir livro](#)

[Alterar livro](#)

**Figura 15. Tela para cadastrar o livro.**

A figura 16, mostra a página que o bibliotecário gerencia as locações.

Gerenciar Locação								
Locacao N°	Usuario	Livro	Data/Locação	Observações na Locação	Previsão de Entrega	Observações na Devolução	Renovar	Devolver
17	46140 - Rafael	15 - Éramos Seis	25/11/2012		25/12/2012		Renovar	Devolver

[Locar](#)  
[Devoluções](#)  
[Menu](#)

**Figura 16. Tela para Gerenciar Locação.**



A figura 17, apresenta a tela que o bibliotecário consulta o histórico de locações.

Devoluções						
Locacao N°	Usuario	Livro	Data/Locação	Observações na Locação	Data/Devolução	Observações na Devolução
15	46140 - Rafael	14 - Como ser um Vendedor de Sucesso	25/11/2012		25/12/2012	
16	46140 - Rafael	15 - Éramos Seis	25/11/2012		25/12/2012	
19	2 - usuario	15 - Éramos Seis	26/11/2012		26/11/2012	
Locações						

**Figura 17. Tela para Consulta do hitórico de devoluções.**

#### 4. Conclusão

O sistema desenvolvido gerará facilidades e vantagens, dentre elas a parte de controle do acervo e acesso às informações de uma forma rápida, sendo capaz de responder perguntas como: Quais livros estão alocados? Quem está com o livro? Ultrapassou a data de devolução? E várias outras, ajudando na tomada de decisões. Um sistema pode gerenciar as informações de forma rápida e precisa, com muita facilidade e precisão, evitando erros. Para gerenciar uma biblioteca o sistema deve gerar facilidades no controle do acervo permitindo ao bibliotecário a automatização dos cadastros de livros, usuários, autores, classificações, locação, devolução e reserva.

O que é feito manualmente, através de planilhas e anotações em cadernos, como controle de locação, lista de livros e consultas, pretende-se tornar a execução dessas tarefas de forma mais fácil, com o *software* é possível executar as mesmas tarefas em um tempo menor e com mais precisão isso porque o acesso à informação deverá ser melhor, mais fácil e mais rápido levando a conclusão que a informática está sempre trazendo facilidades, hoje em dia existem muitos serviços informatizados e para algumas empresas aliadas a suas estratégias permitem o ganho maior em suas produções tornando-se um diferencial no mercado e conseguindo melhores preços e melhores vendas. A maior parte das empresas e comercio estão utilizando sistemas informatizados. O sistema desenvolvido não foi implantado ainda está funcionando em ambiente local para testes e são esperados os benefícios mencionados.

O impacto gerado pela informática no mercado e em varias áreas é muito grande podemos gerar melhorias, conforto, velocidade e lucros maiores. O conhecimento adquirido durante o curso me permite criar ferramentas, como este *software* web para biblioteca, para facilitar muitos serviços.

Para projetos futuros pode-se aprimorar o layout das páginas, criar novas funcionalidades como gerar códigos de barras para os livros, fazer etiquetas de endereçamento para guardar o livro, ajudando na organização e facilitando a localização e consulta de pendencias.

## Referências

ALGAWORKS SOFTWARES E TREINAMENTOS, *Desenvolvimento web com java-server faces Apostila*. (2 ed.). [s.l. : s.n.]. set. 2010.

BRAZ, CHRISTIAN CLEBER MASDEVAL, *Introdução à Linguagem Java*. 77f. Disponível em: [https://docs.google.com/viewer?a=v&q=cache:B4RnRbrN7k4J:ruizuikeda.googlecode.com/files/Introducao\\_java.pdf+Introducao\\_java.pdf&hl=pt-BR&gl=br&pid=bl&srcid=ADGEESg\\_FfcGs0dgcxTab-POY28tgc59MpEzBYv53yv\\_3QqrxXax3EmQz1EJdRfHQigGd8mxSmemGjtUAc5Wc8e7rTOzWLXvrPlx\XXpJjcEw5-iVZIWso3DpWcL8UovbpDU\\_pPd56HJi&sig=AHIEtbToS2zUKqDIF2vE3V9m3XhSE09GBA](https://docs.google.com/viewer?a=v&q=cache:B4RnRbrN7k4J:ruizuikeda.googlecode.com/files/Introducao_java.pdf+Introducao_java.pdf&hl=pt-BR&gl=br&pid=bl&srcid=ADGEESg_FfcGs0dgcxTab-POY28tgc59MpEzBYv53yv_3QqrxXax3EmQz1EJdRfHQigGd8mxSmemGjtUAc5Wc8e7rTOzWLXvrPlx\XXpJjcEw5-iVZIWso3DpWcL8UovbpDU_pPd56HJi&sig=AHIEtbToS2zUKqDIF2vE3V9m3XhSE09GBA)

CARDIM, GUILHERME PINA, *Pesquisa de Sistema operacional i*. 2010. 19f. *Pesquisa realizada para identificar diferenças entre windows e linux*, Departamento de Matemática, Estatística e Computação, Presidente Prudente, 2010. Disponível em: [http://gpcardim.files.wordpress.com/2010/08/pesquisa\\_so.pdf](http://gpcardim.files.wordpress.com/2010/08/pesquisa_so.pdf)

DEITEL, HARVEY M.; DEITEL, PAUL. J., *Java: Como programar*. (6 ed.). São Paulo: Prentice-Hall. 2005

KING, GAVIN ET AL. = E OUTROS, *Hibernate Persistência relacional para java idiomático Documentação de referência hibernate*. [s.l. : s.n.]

KLEINUBING, LUÍZA DA SILVA, *SISTEMAS DE GERENCIAMENTO DE BIBLIOTECAS: um estudo de caso da implantação do Sistema Pergamum na Rede de Unidades de Informação do SENAI/SC*. 2006. 54 f. *Trabalho de conclusão de curso (Bacharel em Biblioteconomia)*, Universidade do Estado de Santa Catarina, Florianópolis, 2006

MELO, PAULO VITOR SALVIANO DE, *REENGENHARIA DO SOFTWARE QUALIFICAR FORNECEDORES*. 2010. 12 f. *Trabalho de conclusão de curso (Bacharel em Ciências da Computação)*, Universidade Presidente Antônio Carlos, Barbacena, 2010

MOREIRA, DANIELA, *Oracle compra Sun por US\$ 7,4 bilhões*. [s.l. : s.n.], 2009. Disponível em: <http://info.abril.com.br/noticias/negocios/oracle-compra-sun-por-us-7-4-bilhoes-20042009-5.shl>

- ORACLE, ORACLE CORPORATION, *Why MySQL*. [s.l. : s.n.], 2011. Disponível em: <http://www.mysql.com/why-mysql>. Acessado em: 30 Mai. 2012.
- SARAIVA, BRUNA GABRIELI GOMES ET AL.= E OUTROS, *Uso de softwares livres para automação de bibliotecas*. 4 f. Disponível em: <http://ueads1.textolivre.pro.br/2011.1/papers/upload/57.pdf>. Acesso em: 14 Fev. 2012
- SILVA, NAIRON NERI, *JREVIEW: PLATAFORMA COLABORATIVA DE AVALIAÇÃO DE PRODUTOS*. 2011. 61 f. Monografia (Pós-Graduação Lato Sensu em Orientação a Objetos e Aspectos), Universidade Federal de Lavras, Lavras, 2011