

### 3. METODOLOGIA

O sistema foi construído utilizando-se um processo de engenharia de software. Por se tratar de um sistema pequeno e para fins acadêmicos, é ideal que seja aplicado um processo simples e objetivo, como o XP.

XP ou Programação extrema é uma metodologia ágil para equipes pequenas que irão desenvolver um software com requisitos vagos e em constante mudança .(XP,2010)

Foram avaliadas as tecnologias de mercado que se adequavam à Web 2.0 e RIA para a construção do software. As tecnologias avaliadas estão de acordo com os padrões definidos para W3C e além disso, recursos citados por nomes como O'Reilly, Shutta, Garret entre outros para a construção de aplicações Web 2.0 também foram levados em consideração para a construção do sistema.

O sistema foi criado para ser executado em um servidor Web que foi definido no momento inicial da construção. As tecnologias escolhidas para a construção do sistema atendem à necessidade do coordenador de laboratórios visando o uso real do sistema.

Para levantamento de requisitos foram realizadas entrevistas com o coordenador de laboratórios e/ou professores que fazem a utilização de laboratórios de informática.

A partir de um levantamento mínimo de requisitos foram construídas *userstories* que definiram as reais necessidades do sistema. Ainda atendendo à metodologia XP, após a definição dos requisitos, foram realizadas construções de pequenas versões do software para a validação juntamente com o coordenador de laboratórios.

Após a validação do coordenador foram realizados testes para garantir a qualidade do software.

A documentação gerada é pequena visto que o sistema deve por si só ser intuitivo e com boa usabilidade.

### 3.1. Ferramentas Utilizadas

Para atender os requisitos da aplicação, foi realizado um estudo sobre qual seria a arquitetura de software ideal para o sistema e que se adequasse também aos requisitos de infraestrutura da PUCMinas. Com base nisso ficou decidido que o sistema seria construído com as seguintes tecnologias:

PHP: “É uma linguagem de script open source de uso geral, muito utilizada e especialmente guarnecida para o desenvolvimento de aplicações Web embutível dentro do HTML.”(PHP,2011). A versão utilizada é a 5.3.1 que pode ser baixada na seção de downloads da página oficial do PHP(PHP,2011);

CakePHP: “CakePHP é um framework de desenvolvimento rápido para PHP que fornece uma arquitetura extensível para desenvolvimento, manutenção, e distribuição de aplicações.”(CAKEPHP,2011). A versão utilizada é a 1.3.6 que pode ser baixada na seção de downloads da página oficial do CakePHP(CAKEPHP,2011).

jQuery: jQuery é uma biblioteca javascript simples e concisa que simplifica o trabalho com objetos HTML, eventos, animações e interações ajax para um rápido desenvolvimento web.”(JQUERY,2011). A versão utilizada foi a 1.4.4 que pode ser baixada na seção de downloads da página oficial do jQuery(JQUERY,2011);

Apache Server: É um servidor HTTP e é o mais bem sucedido servidor web livre.”(WIKIPEDIA,2010). A versão utilizada é a 2.0 e pode ser baixada na seção de downloads da página oficial do Apache(APACHE,2011);

Mysql: “MySQL é um sistema de gerenciamento de banco de dados (SGBD”), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface.”(WIKIPEDIA,2010). A versão utilizada é a 5.1.44 e pode ser baixada na seção de downloads da página oficial do Mysql(MYSQL,2011)

## **4. IMPLEMENTAÇÃO SISTEMA ACADLABS**

O sistema construído para a gestão de laboratórios de informática da PUCMinas foi denominado AcadLabs. O sistema tem o objetivo de sistematizar o controle e gestão de alocação de laboratórios sob a gerência do coordenador de laboratórios. Construído na plataforma Web o coordenador de laboratórios de informática terá em suas mãos a ferramenta adequada para montar os horários conforme as solicitações dos professores. O sistema faz grande uso de componentes ricos para internet, levando ao usuário boa experiência de utilização de interface.

### **4.1. Descrição do Sistema**

O sistema foi construído para que o coordenador de laboratórios pudesse realizar suas tarefas de gestão de alocação laboratórios.

O sistema possui uma interface rica, componentes ricos, chamadas assíncronas (AJAX) e conceitos de web2.0. A idéia adotada foi criar um sistema com uma interface simples e intuitiva e que tivesse uma boa usabilidade.

Os professores irão solicitar por e-mail ao coordenador um laboratório para um determinado dia e horário e então o coordenador irá registrar essa reserva no sistema. Dessa forma o sistema permitirá ao coordenador visualizar quais são os laboratórios disponíveis por dia e horário. Para que o sistema se tornasse completo foram implementadas funcionalidades que alimentam o sistema com as informações básicas, como por exemplo, cadastro de laboratórios, cadastro de softwares e cadastro de disciplinas.

Para obter uma solução completa, relatórios podem ser gerados para que o coordenador possa passar à secretaria e aos professores. Existem relatórios referentes à alocação dos laboratórios e de softwares por laboratório.

## 4.2. Especificação de Requisitos

O sistema deveria estar disponível na plataforma Web. A utilização de conceitos Web2.0 é obrigatória e recursos que tornem o sistema com uma interface rica também devem ser utilizados. O sistema será construído com base no modelo de dados conforme a figura 6.

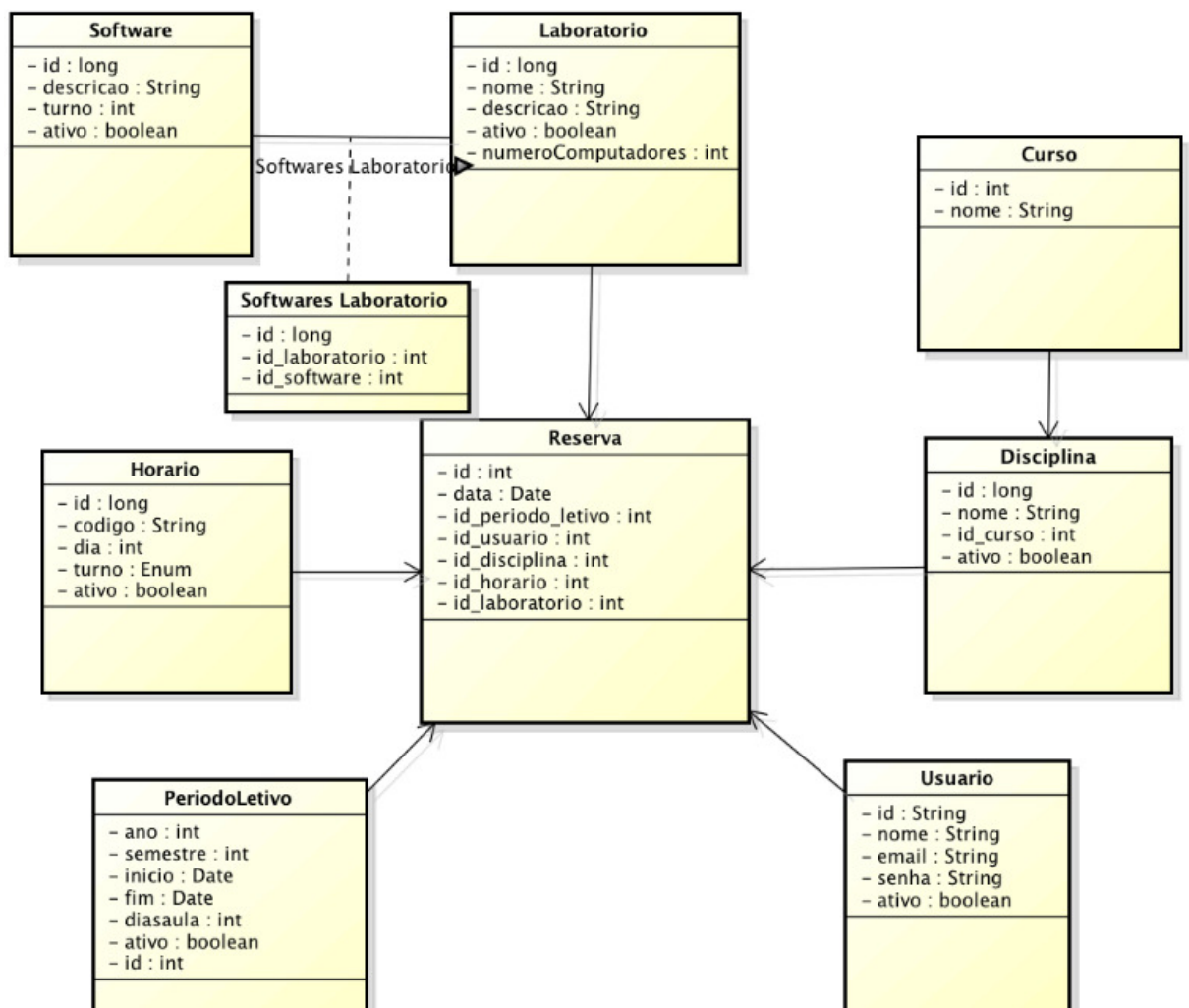


Figura 6: Modelo UML de classes de domínio do sistema AcadLabs.

Fonte: Elaborado pelo autor.

O modelo representado pela Figura 6 possui nove entidades. A entidade Software armazenará os softwares/sistemas de computador cadastrados pelo administrador do sistema. Esses softwares poderão ser associados aos laboratórios que serão cadastrados na entidade Laboratório através da entidade Softwares Laboratório. A entidade Curso irá armazenar os cursos

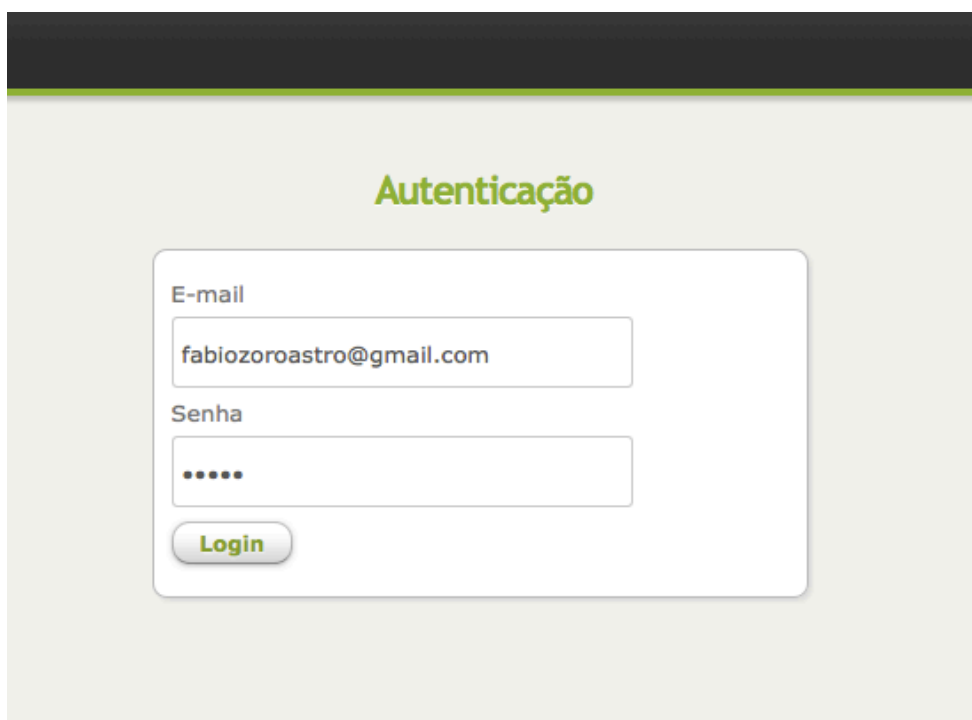
existentes na PUC Minas São Gabriel. As disciplinas poderão ser cadastradas na entidade Disciplina sendo que devem possuir um relacionamento com Curso.

A entidade Período Letivo armazena o início e término de um período letivo. A entidade horário armazena os horários que a PUC Minas São Gabriel disponibiliza seus laboratórios de informática. A entidade Reserva será alimentada com os registros de reservas que o usuário irá registrar ao longo de um período letivo.

A entidade usuário irá armazenar apenas um usuário que é o coordenador de laboratórios.

#### 4.3.1 Autenticação

Deveria existir o mecanismo de autenticação de usuários por parte do servidor web. A qualquer momento o usuário poderá sair do sistema clicando na opção de *logout* do sistema. Apenas o coordenador de laboratórios terá acesso ao sistema e somente ele poderá reservar os laboratórios. A figura 7 representa esta funcionalidade.



A imagem mostra uma interface web de autenticação. No topo, há uma barra preta com uma linha decorativa verde. Abaixo, o título "Autenticação" aparece em verde. O formulário principal é branco e contém dois campos de entrada: "E-mail" com o texto "fabiozoroastro@gmail.com" e "Senha" com pontos para ocultar o texto. Abaixo dos campos, há um botão "Login" com um efeito 3D e uma sombra.

Figura 7: *Screenshot* da funcionalidade de autenticação de usuário.

Fonte: Elaborado pelo autor.

#### 4.3.2 Disponibilidade de menu

Para facilitar o entendimento do usuário, o sistema deverá possuir duas divisões de menu. Uma para a parte de cadastros gerais e uma para a parte de reserva e administração. A figura 8 representa esse requisito.

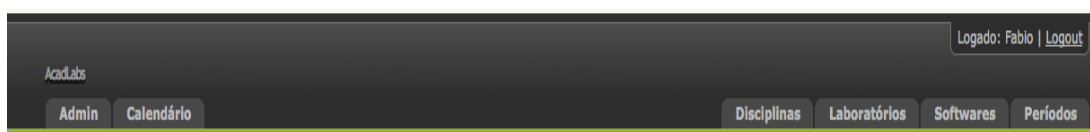


Figura 8: *Screenshot* da funcionalidade de divisão de menus.

Fonte: Elaborado pelo autor.

#### 4.3.3 Manutenção de disciplinas

A listagem das disciplinas deverá ocorrer com paginação a cada quinze itens e deve ser feita com a utilização de AJAX. Deve haver um filtro para pesquisa de disciplina por descrição. Deve existir um botão com a opção de incluir uma nova disciplina. Para alterar o registro o usuário terá a opção de clicar em cima da disciplina e então será redirecionado para a tela de edição. A figura 9 representa essa funcionalidade.

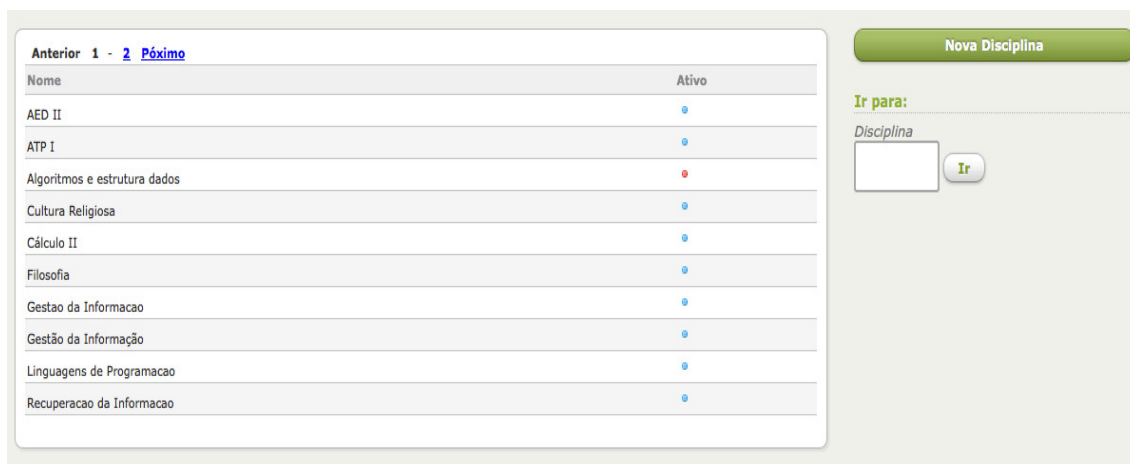


Figura 9: *Screenshot* da funcionalidade de listagem e pesquisa de disciplinas.

Fonte: Elaborado pelo autor.

As disciplinas poderão ser cadastradas pelo usuário, sendo que elas devem ser associadas a algum curso previamente existente na tabela de cursos. O tipo de exclusão será lógica para que não haja danos em reservas já cadastradas para o período atual ou períodos anteriores. Quando não for possível cadastrar uma disciplina por falta de informações, o usuário deverá ser informado sobre os problemas encontrados. A figura 10 representa essa funcionalidade.

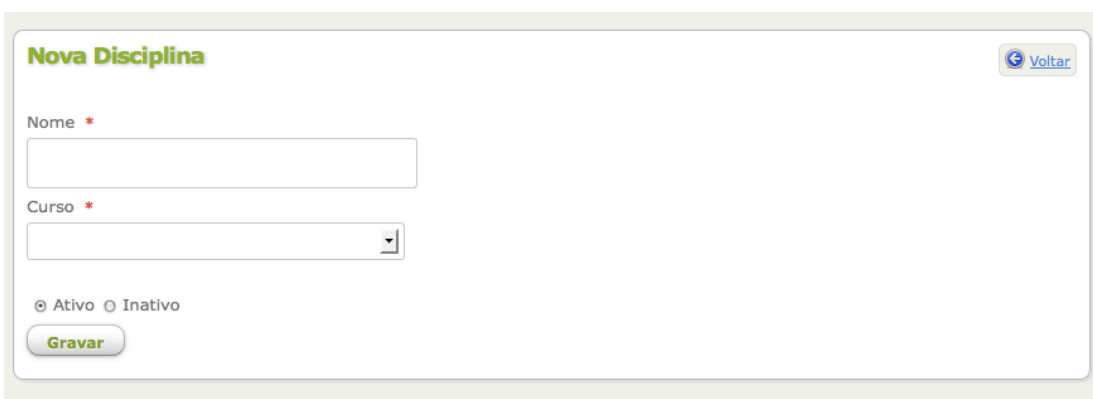
A screenshot mostra uma interface web para o cadastro de uma nova disciplina. No topo, há o título "Nova Disciplina" em verde e um botão "Voltar" com uma seta azul. Abaixo, há um formulário com dois campos obrigatórios: "Nome" e "Curso", ambos marcados com uma estrela vermelha. O campo "Nome" é uma caixa de texto vazia, e o campo "Curso" é uma caixa de seleção com uma seta para baixo. Abaixo dos campos, há duas opções de rádio: "Ativo" (selecionada) e "Inativo". No final do formulário, há um botão "Gravar" em verde.

Figura 10: *Screenshot* da funcionalidade de cadastro de disciplinas.

Fonte: Elaborado pelo autor.

#### **4.3.4 Manutenção de laboratórios**

A listagem dos laboratórios deverá ocorrer com paginação a cada quinze itens e deve ser feita com a utilização de AJAX. Deve haver um filtro para pesquisa do laboratório por nome. Deve existir um botão com a opção de incluir um novo laboratório e um botão para associar softwares aos laboratórios. Para alterar o registro o usuário terá a opção de clicar em cima do laboratório e então será redirecionado para a tela de edição. A figura 11 representa essa funcionalidade.

Anterior		Póximo	
Nome	Nº de Computadores	Ativo	
J4	50	<input checked="" type="radio"/>	
J8	0	<input checked="" type="radio"/>	
J9	30	<input checked="" type="radio"/>	
L2	1	<input checked="" type="radio"/>	
UUUUUUU	0	<input checked="" type="radio"/>	

**Novo Laboratório**

**Associar Softwares**

Ir para:

Laboratório

Figura 11: *Screenshot* da funcionalidade de listagem e pesquisa de laboratórios.

Fonte: Elaborado pelo autor.

Os laboratórios poderão ser cadastrados pelo usuário, sendo que eles podem ter softwares associados. O tipo de exclusão será lógica para que não haja danos em reservas já cadastradas para o período atual ou períodos anteriores. Quando não for possível cadastrar um laboratório por falta de informações, o usuário deverá ser informado sobre os problemas encontrados. As figuras 12 e 13 representam essa funcionalidade.

### Novo Laboratório

[Voltar](#)

Novo Laboratório \*

Nº de Computadores \*

Descrição

☒ Ativo ☐ Inativo

Figura 12: *Screenshot* da funcionalidade de cadastro de laboratórios.

Fonte: Elaborado pelo autor.

### Associação de Softwares

[Voltar](#)

Laboratórios \*

Softwares \*

☐ Adobe Suite Full

☐ Eclipse

☒ Latex

☒ Macromedia Suite

☒ Office

☒ Telnet

☒ Weka

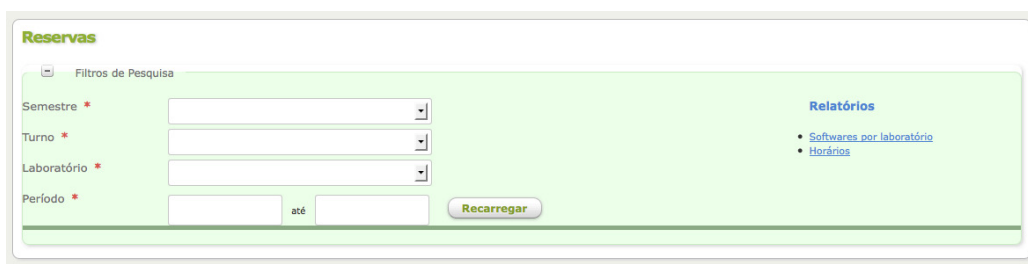


Figura 13: *Screenshot* da funcionalidade de associações de softwares à laboratórios.

Fonte: Elaborado pelo autor.

#### 4.3.5 Manutenção de reservas

As reservas devem ser feitas por período letivo, laboratório, turno e data conforme representado na figura 14. Todos os filtros são obrigatórios. Os campos de data que determinam o período a ser exibido devem usar componentes ricos para digitação de data. A tela de filtro poderá ser expandida ou inibida para que o usuário possa ter a tela completa para realização de reservas.



The screenshot shows a web interface titled 'Reservas'. It features a 'Filtros de Pesquisa' (Search Filters) section with four mandatory fields: 'Semestre' (Semester), 'Turno' (Shift), 'Laboratório' (Laboratory), and 'Período' (Period). Each field has a dropdown menu. The 'Período' field is split into two parts with 'até' (until) in between. A 'Recarregar' (Reload) button is located below the filters. To the right, under the heading 'Relatórios' (Reports), there are two links: 'Softwares por laboratório' and 'Horários'.

Figura 14: *Screenshot* dos filtros para a geração de relatórios e criação de reservas.

Fonte: Elaborado pelo autor.

Caso o coordenador deseje remover uma ou mais reservas, ele terá as duas opções visíveis na reserva para executar. O ícone vermelho permitirá o usuário remover apenas uma reserva e o ícone de lixeira permitirá o usuário remover todas as reservas do dia selecionado até o data fim informada no filtro de período. A figura 15 representa essa funcionalidade.

Turno	01/02/2011 Terça-Feira	02/02/2011 Quarta-Feira	03/02/2011 Quinta-Feira	04/02/2011 Sexta-Feira	05/02/2011 Sábado	07 Seg
07h00 - 07h50 (07h40 - 08h30) *	+	+	+	+	+	
07h50 - 08h40 (08h30 - 09h20) *	+	+	Ocupado Sistemas de Informacao Gestao da Informacao	+	+	
08h50 - 09h40 (09h30 - 10h20) *	+	Ocupado Sistemas de Informacao Linguagens de Programacao	+	+	+	
09h40 - 10h30 (10h20 - 11h10) *	+	+	+	+	+	
10h40 - 11h30 (11h20 - 12h10) *	+	+	+	+	+	
11h30 - 12h20 (12h10 - 13h00) *	+	+	+	+	+	

Figura 15: Screenshot da tela de reservas existentes, remoção de reservas e criação de novas.

Fonte: Elaborado pelo autor.

Somente poderão ser criadas reservas para horários livres. Deverá existir a funcionalidade de criar reserva contínua, que consiste em criar reservas para o determinado dia da semana durante todo o período letivo. A reserva deverá ser persistida e imediatamente atualizada no quadro de reservas para os filtros informados. As figuras 15 e 16 representam essa funcionalidade.

Realizar Reserva

02/02/2011

Disciplina  
AED II

Recorrente  
☒

Reservar
Cancelar

Figura 16: Screenshot da tela de criação de nova reserva.

Fonte: Elaborado pelo autor.

#### 4.3.6 Relatórios

Os relatórios estarão disponíveis na mesma página de reservas pois os filtros da pesquisa de reserva será reaproveitado para a geração de relatórios. Deverá haver um relatório que informe a quantidade de computadores e quais são os softwares associados à um laboratório. O outro relatório irá informar quais são as reservas para o laboratório e período e informado. Os relatórios deverão ser semelhantes aos das figuras 17 e 18.

O relatório de reservas poderá ser montado com base no período informado, isso indica que o usuário poderá ver as reservas de uma semana, de um dia ou de todo o semestre. Caso o usuário informe o turno, apenas as reservas daquele turno serão exibidas, caso contrário, serão exibidas as reservas para todos os turnos.

<b>Laboratório</b>	14
<b>Número de computadores</b>	50
<b>Softwares</b>	
Macromedia Suite	Disponível
Latex	Disponível
Office	Disponível
Telnet	Disponível
Weka	Disponível

Figura 17: *Screenshot* do relatório de softwares por laboratório.

Fonte: Elaborado pelo autor.

Laboratório		14					
Número de computadores		50					
Turno	01/02/2011 Terça-Feira	02/02/2011 Quarta-Feira	03/02/2011 Quinta-Feira	04/02/2011 Sexta-Feira	05/02/2011 Sábado	07/02/2011 Segunda-Feira	08/02/2011 Terça-Feira
07h00 - 07h50 (07h40 - 08h30) *							
07h50 - 08h40 (08h30 - 09h20) *			Gestao da Informacao (Sistemas de Informacao)				
08h50 - 09h40 (09h30 - 10h20) *		Linguagens de Programacao (Sistemas de Informacao)					
09h40 - 10h30 (10h20 - 11h10) *							
10h40 - 11h30 (11h20 - 12h10) *							
11h30 - 12h20 (12h10 - 13h00) *							

Figura 18: *Screenshot* do relatório de reservas por período.

Fonte: Elaborado pelo autor.

#### 4.4. Contribuições à comunidade

O sistema deve apresentar mensagens não obstrutivas ao usuário quando as ações forem executadas. Tal comportamento é um requisito de web 2.0. As informações armazenadas no banco de dados devem ser íntegras, ou seja, armazenar apenas informações válidas. A figura 19 representa essa funcionalidade.

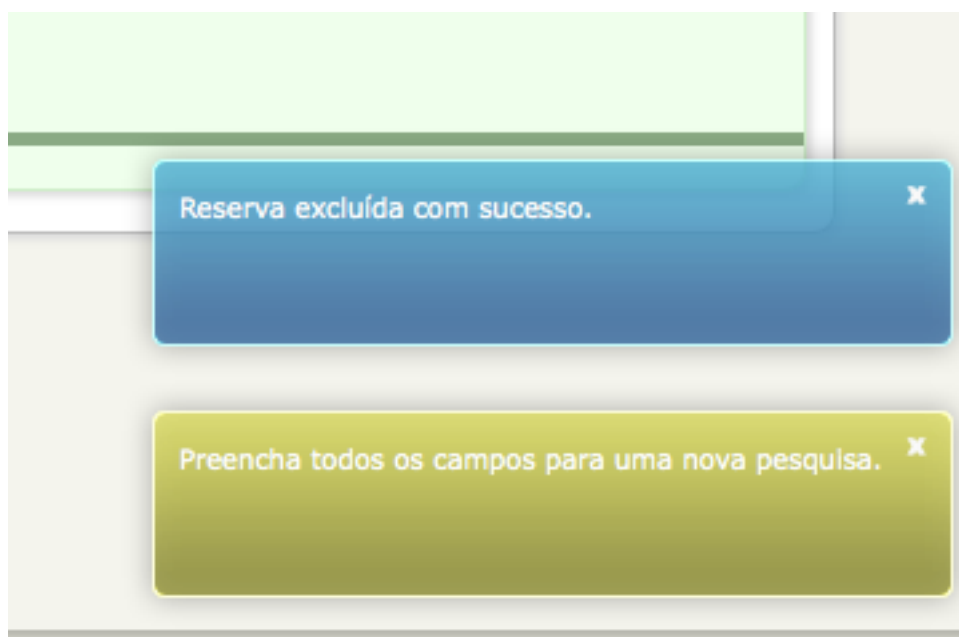


Figura 19: *Screenshot* do componente de mensagens estilo *growl*.

Fonte: Elaborado pelo autor.

Para atender o requisito de exibir as mensagens de status de ações ao usuário de forma não obstrutiva foi realizada uma profunda pesquisa para a utilização de um componente adequado. Dos componentes encontrados nenhum se encaixava bem com a proposta.

Portanto, com base nos estudos das tecnologias utilizadas para a construção do projeto, foi criado um componente para exibir mensagens flash – Aquelas que são exibidas e em seguida são removidas do escopo -, e utilizado no projeto. O resultado foi positivo pois além de atender o requisito, o padrão de desenvolvimento *Helper* foi utilizado e o componente atende completamente os requisitos de Web2.0 e RIA. O componente criado ficou bem codificado e estruturado de forma que ele poderia ser utilizado em qualquer projeto web que possuísse em sua arquitetura o framework CakePhp e JQuery.

Após a criação do componente, o mesmo foi publicado através do link (ZOROASTRO, 2011) no site do fornecedor do CakePhp(BAKERY,2011). Batizado de Growl Flash Messages o artigo com divulgação do componente teve mais de quatro mil acessos confirmados até quatro de junho de 2011.

Esse componente se tornou uma contribuição oficial à comunidade de desenvolvedores PHP podendo ser utilizado livremente sob a licença livre.

## 5. CONSIDERAÇÕES FINAIS

O problema de alocação de laboratórios é uma realidade da PUC Minas São Gabriel. Não há nenhum processo definido para a realização das tarefas de gestão o que ocasiona em um conhecimento tácito que fica apenas com o atual coordenador de laboratórios. A proposta desse trabalho é construir um sistema baseado na plataforma web com larga utilização de conceitos atuais, como por exemplo, *Web2.0*, *AJAX*, *RIA*. O sistema foi denominado como AcadLabs. Foram estudados conceitos como JSON, AJAX, requisições síncronas, requisições assíncronas, padrões de projeto. A principal facilidade que o sistema proporciona é a gestão das reservas de laboratórios. Com este processo informatizado e disponível na Web, o coordenador de laboratórios poderá realizar as reservas de laboratórios onde estiver e poderá gerar relatórios gerenciais para distribuir para professores, secretaria acadêmica e colocar na porta dos laboratórios como é feito atualmente.

Na tentativa de utilizar as novas tecnologias como HTML5, AJAX e RIA foram encontrados problemas de renderização dos componentes visuais, diferenças entre navegadores e até mesmo uma percepção de que a framework CakePHP já parece estar um pouco defasada por não utilizar *tags* do *HTML* nativamente.

Em um certo momento existiu a necessidade de utilizar um componente rico para atender ao requisito de exibir mensagens não obstrutivas e não havia nada implementado na internet que poderia ser utilizado. Portanto, a partir dessa necessidade foi criada uma implementação de um componente de mensagens não obstrutivas com base nos DesignPatterns *Flash Messagee Helper*. Esse componente criado tem como objetivo exibir mensagens ao usuário sendo que as mesmas não obstruem a interação do usuário com o sistema. Após a criação desse componente, o mesmo foi extraído do projeto AcadLabs e disponibilizado no site do fornecedor da biblioteca CakePHP, o Bakery(Bakery,2011). Para maior entendimento e download do componente basta acessar o endereço disponível na seção de artigos do Bakery(ZOROASTRO,2011).

As tecnologias utilizadas para a construção do projeto são predominantes no mercado e podem servir até mesmo de estudo para outros estudantes que queiram aprender sobre desenvolvimento real de um sistema de informação.

Em trabalhos futuros seria interessante incrementar um novo módulo ao sistema capaz de permitir aos professores solicitarem a alocação de laboratórios e recebessem uma notificação de deferimento e/ou indeferimento de autorizações de suas solicitações. Como o sistema ficará disponível na internet é interessante que o sistema tenha uma evolução na parte de segurança com a utilização de uma camada SSL. O sistema foi inicialmente construído para a unidade São Gabriel da PUC Minas, porém ele pode ser adaptado para outras unidades caso seja necessário executar a aplicação em um mesmo servidor, pois se for executar em servidores diferentes não haverá a necessidade de alterar nada no sistema.

## REFERÊNCIAS

APACHE. **Servidor HTTP**. Disponível em:

<<http://httpd.apache.org/>> Acesso em 29 de maio de 2011.

ASLESON, Ryan. Shutta, Nathaniel, **Fundamentos do Ajax**, 2006.

BAKERY. **Proprietária do CakePHP**. Disponível em:

<<http://bakery.cakephp.org/>> Acesso em 29 de maio de 2011.

CAKEPHP. **Framework MVC para PHP**. Disponível em:

<<http://www.cakephp.com.br/>> Acesso em 29 de maio de 2011.

DEITEL, Harvey M. **Internet & World Wide, Como Programar**, 2006.

GARRETT, Jesse James. **Ajax: A New Approach to Web Applications**, 2005.

Disponível em <<http://www.adaptivepath.com/ideas/essays/archives/000385.php>>.

Acesso em 24 de outubro de 2010

JQUERY. **Biblioteca Javascript JQuery**. Disponível em:

<<http://jquery.com/>> Acesso em 29 de maio de 2011.

JSON. **JavaScriptObjectNotation**. Disponível em:

<<http://json.org/>> Acesso em 15 de novembro 2010.

MACEDO, Marcelo da Silva, **Construindo sites adotando padrões Web**, 2004.

MYSQL. **Sistema gerenciador de banco de dados MYSQL Oracle**.

Disponível em:

<<http://www.mysql.com/>> Acesso em 29 de maio de 2011.

O'REILLY, Tim. **Web 2.0 Principles and Best Practices**, 2007.

PHP. **Linguagem de Script**. Disponível em:

<<http://www.php.net/>> Acesso em 29 de maio de 2011.

W3C. **Extensible markup language (xml)**. Disponível em:

<<http://www.w3.org/XML/>> Acesso em 20 de outubro 2010.

W3C. **People Berners-Lee**. Disponível em:

<<http://www.w3.org/People/Berners-Lee/>> Acesso em 24 de outubro 2010.

WIKIPEDIA. **Aplicações Ricas para Internet**. Disponível em:

<[http://pt.wikipedia.org/wiki/Internet\\_rica](http://pt.wikipedia.org/wiki/Internet_rica)> Acesso em 24 de outubro 2010.



WIKIPEDIA. **Aplicações Ricas para Internet**. Disponível em:  
<<http://pt.wikipedia.org/wiki/MySQL>> Acesso em 24 de outubro 2010.

WIKIPEDIA. **Aplicações Ricas para Internet**. Disponível em:  
<[http://pt.wikipedia.org/wiki/Servidor\\_Apache](http://pt.wikipedia.org/wiki/Servidor_Apache)> Acesso em 24 de outubro 2010.

XP. **Extreme Programming**. Disponível em:  
<<http://www.extremeprogramming.org/>> Acesso em 15 de novembro 2010.

ZOROASTRO. **Artigo com componente RIA para mensagens não obstrutivas**. Disponível em:  
<[http://bakery.cakephp.org/articles/fabiozoroastro/2011/04/15/growl\\_flash\\_messages](http://bakery.cakephp.org/articles/fabiozoroastro/2011/04/15/growl_flash_messages)>  
Acesso em 29 de maio de 2011.

## APÊNDICES

### APÊNDICE A – Relatório de reservas elaborado sem a utilização do AcadLabs



Pontifícia Universidade Católica de Minas Gerais – PUC Minas no São Gabriel  
Centro de Recursos Computacionais (CRC)  
Horário dos Laboratórios de Informática

Laboratório: J1						Número de computadores: 30	
Horário	Segunda	Terça	Quarta	Quinta	Sexta	Horário	Sábado
07h00-08h40			Oficina de Com. Visual Grupo 1 (Com. Integrada)		Motores Gráficos Grupo 1 (Jogos Digitais)	07h40-09h20	Oficina de Com. Visual Grupo 1 (Com. Integrada)
08h50-10h30	Produção de Sites (Com. Integrada)			Ferramentas de Autoria (Jogos Digitais)	Ferramentas de Autoria (Jogos Digitais)	09h30-11h10	
10h40-12h20		Estatística Aplicada Grupo 1 (Administração)	Oficina de Com. Visual Grupo 2 (Com. Integrada)	Produção de Sites (Com. Integrada)		11h20-13h00	Oficina de Com. Visual Grupo 2 (Com. Integrada)
12h30-13h30						13h10-14h50	
13h30-15h10						15h00-16h00	
15h20-17h00							
17h10-17h50							
19h00-20h40	Informática Grupo 2 (Administração)	Lab. Computação I Turma 2 – Grupo 1 (Sist. Informação)	Produção Gráfica (Publ. Propaganda)		Estatística Aplicada à Administração Grupo 1 (Administração)		
20h50-22h30	Lab. Computação I Turma 1 – Grupo 1 (Sist. Informação)		Projeto de Sistemas de Informação Grupo 3 (Sist. Informação)		Tópicos Especiais (Sist. Informação)		

horario\_lab\_inf-1\_2011\_03052011

Figura 20: Relatório atual utilizado pelo coordenador de laboratórios.

Fonte: Elaborado pelo autor.

## APÊNDICE B – Levantamento inicial de Requisitos

### Levantamento inicial de requisitos

#### 1. Cadastro de Laboratorios(CRUD)

##### Fluxo Incluir/Alterar

- Valores a serem registrados:
  - Código
  - Descrição
  - Ativo
- Associar Softwares ao Laboratório
- 

#### 2. Cadastro de Softwares(CRUD)

##### Fluxo Incluir/Alterar

- Valores a serem registrados:
  - Código
  - Descrição

#### 3. Cadastro de disciplinas CRUD

#### 3. Cadastro de Período Letivo(CRUD)

##### Fluxo Incluir/Alterar

- Valores a serem registrados:
  - Ano
  - Semestre
  - Início
  - Término
  - (binary)Dias com aula(1 segunda, 3 terça, 7 quarta...)
  - Ativo

#### 4. Criação do script com dados de horários

- Fluxo Principal
- Criação do script com os dados dos horários

#### 5. Associação de horários

##### Fluxo principal(Associação)

- Listagem dos horários por Semestre/Turno
- Listar apenas dias cadastrados(Dias com aula)
- Associar um laboratório a um horário. Deverá haver um filtro para exibir os laboratórios por software;
  - Deseja vincular durante todo o período letivo?
    - Esta opção cria apenas um registro com o identificador de sempre = 1;

##### Fluxo Trocar Semestre/Turno

- Reiniciar a tela carregando as informações já existentes

##### Fluxo remover Reserva

- Remove o registro fisicamente
- Para reservas do tipo SEMPRE:
  - Como é apenas um registro, todas as reservas deste laboratório serão perdidas

#### 6. Relatório

##### Fluxo principal

- Exibe relatório dos reservas por semestre/turno

Figura 21: Documento inicial de levantamento de requisitos.

Fonte: Elaborado pelo autor.

## APÊNDICE C – Problemas encontrados

### Problemas encontrados:

Na tentativa de usar HTML 5 para os componentes(Como por exemplo *Spinner*) ocorriam varias interpretações diferentes para os browsers testados.

A ferramenta *CakePHP* ainda não suporta componentes HTML 5. Dessa forma, para aproveitar o que o framework lhe dispõe às vezes você não consegue utilizar os componentes mais modernos.

Quero usar no sistema a funcionalidade de mensagens não intrusivas. Um bom exemplo é de quando gravamos um registro com sucesso, a mensagem é exibida no estilo *growl (osX)*.

Usar o sistema internacionalizado iria atrasar o desenvolvimento.

• Figura 22: Documento com registros de problemas encontrados durante a codificação.

Fonte: Elaborado pelo autor.