

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

GUILHERME ROSA SEVERO

## **Sistema de Apoio ao Pesquisador**

Trabalho de Graduação.

Prof. Dr. Leandro Krug Wives  
Orientador

Prof. Dr. Marco Aurélio Chaves Cepik  
Coorientador

Porto Alegre, dezembro de 2013

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do Curso de Ciência da Computação: Prof. Raul Fernando Weber

Bibliotecário-Chefe do Instituto de Informática: Alexander Borges Ribeiro

## **AGRADECIMENTOS**

Agradeço a primeiramente a minha família por ter me apoiado incondicionalmente, ter me criado com respeito e me passado valores. Sem eles com certeza eu não estaria concluindo esta etapa da minha vida.

Aproveito para agradecer meus amigos e meus colegas de graduação por participarem dessa jornada incrível que foi a graduação na UFRGS. Entre festas e grupos de estudos conseguimos alcançar a graduação no curso que é hoje o melhor curso de ciência da computação do país na universidade que também está ranqueada como a melhor universidade do país. Isso é uma oportunidade que eu agradeço muito poder ter tido o privilégio de fazer parte dessa história.

Agradeço meus colegas de trabalho do CEGOV que com certeza são os responsáveis pelo meu amadurecimento profissional e pessoal. Com eles tive a oportunidade de conhecer pessoas diferentes, de culturas diversas que me fizeram enxergar o mundo de uma forma diferente. Foi uma experiência única poder fazer parte do CEGOV e da “Família Cepik”.

Agradeço os professores e técnicos do Instituto de Informática e Matemática pela oportunidade de receber seus conhecimentos de suas áreas específicas. Não só conhecimento, ao longo do curso vocês me moldaram como um adulto responsável e de caráter por me mostrar os valores pessoais e profissionais.

Fica aqui meu muito obrigado. Muito obrigado por fazerem parte da minha vida, eu agradeço muito cada oportunidade que tive ao lado de vocês.

## SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS</b> .....	<b>V</b>
<b>LISTA DE FIGURAS</b> .....	<b>VI</b>
<b>LISTA DE TABELAS</b> .....	<b>VIII</b>
<b>RESUMO</b> .....	<b>IX</b>
<b>ABSTRACT</b> .....	<b>X</b>
<b>1 INTRODUÇÃO</b> .....	<b>1</b>
<b>2 ANÁLISE DE REQUISITOS</b> .....	<b>3</b>
2.1 As Fases de um Projeto.....	3
2.2 Requisitos do Sistema.....	4
2.2.1 Requisitos Funcionais.....	4
2.2.2 Requisitos Não Funcionais.....	5
<b>3 FERRAMENTAS UTILIZADAS</b> .....	<b>6</b>
3.1 Aplicação Web.....	6
3.2 Linguagens e Tecnologias Utilizadas.....	6
3.2.1 HTML.....	7
3.2.2 CSS.....	7
3.2.3 Javascript.....	8
3.2.4 PHP.....	9
3.2.5 Framework Yii PHP.....	10
3.2.5.1 Active Record.....	11
3.2.5.2 Controllers.....	13
3.2.5.3 Yii Code Generator (Gii).....	14
<b>4 O SISTEMA DE APOIO AO PESQUISADOR (SIPESQ)</b> .....	<b>17</b>
4.1 Descrição das funcionalidades.....	17
4.1.1 Projetos.....	18

4.1.2	Pessoas .....	18
4.1.3	Atividades .....	18
4.1.4	Acervo .....	18
4.1.5	Contatos .....	18
4.1.6	Wiki .....	19
4.1.7	Gerencial .....	19
<b>5</b>	<b>O SISTEMA DESENVOLVIDO .....</b>	<b>20</b>
<b>5.1</b>	<b>Metodologia Utilizada .....</b>	<b>20</b>
<b>5.2</b>	<b>Modelagem da Base de Dados .....</b>	<b>23</b>
<b>5.3</b>	<b>Entidades da Base de Dados .....</b>	<b>25</b>
5.3.1	Pessoa .....	25
5.3.2	Grupo .....	25
5.3.3	Atividade .....	25
5.3.4	AtividadeCategoria .....	26
5.3.5	AtividadePasso .....	27
5.3.6	Projeto .....	27
5.3.7	ProjetoArquivo .....	27
5.3.8	Rubrica .....	27
5.3.9	ProjetoOrcamento .....	27
5.3.10	ProjetoVerba .....	27
5.3.11	ProjetoDesembolso .....	28
5.3.12	ProjetoDespesa .....	28
5.3.13	PermissaoProjeto .....	28
5.3.14	Contato .....	28
<b>5.4</b>	<b>Implementação da Base de Dados e Geração de Código .....</b>	<b>28</b>
<b>5.5</b>	<b>Visões do Sistema .....</b>	<b>30</b>
5.5.1	Tela Inicial do Sistema .....	31
5.5.2	Visões de Projeto .....	33
5.5.3	Visões de Pessoas .....	43
5.5.4	Visões dos menus do sistema .....	43
5.5.4.1	Menu de Operações .....	43
5.5.4.2	Menu de Configurações .....	44
5.5.4.3	Notificações .....	44
5.5.4.4	Busca do sistema .....	45
5.5.5	Visões de Grupos .....	46
5.5.6	Visões de Atividades .....	47
5.5.7	Visão do Relatório Geral .....	50
5.5.8	Visão da Wiki .....	51
<b>5.6</b>	<b>Parâmetros do Sistema .....</b>	<b>53</b>
<b>5.7</b>	<b>Análise do Sistema .....</b>	<b>53</b>
<b>5.8</b>	<b>Usabilidade do Sistema .....</b>	<b>57</b>
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>62</b>
<b>6.1</b>	<b>Limitações do Sistema .....</b>	<b>62</b>
<b>6.2</b>	<b>Trabalhos Futuros .....</b>	<b>62</b>

**REFERÊNCIAS..... 64**

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
AR	Active Record
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CEGOV	Centro de Estudos Internacionais sobre Governo
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CPD	Centro de Processamento de Dados
CRUD	Create Retrieve Update Delete
CSS	Cascading Style Sheets
DOM	Data Object Model
FAURGS	Fundação de Apoio da Universidade Federal do Rio Grande do Sul
Gii	Yii Code Generator
HTML	HyperText Transfer Protocol
JSON	JavaScript Object Notation
MVC	Model-View-Controller
PHP	PHP: Hipertext Preprocessor
SGBD	Sistema de Gerenciamento de Banco de Dados
SIPESQ	Sistema de Apoio ao Pesquisador
SQL	Structured Query Language
UFRGS	Universidade Federal do Rio Grande do Sul
URL	Uniform Resource Locator
Yii	Yes, it is!

## LISTA DE FIGURAS

Figura 3.1: Código de exemplo de utilização de HTML em conjunto com CSS .....	7
Figura 3.2: Alteração de cor de fundo de elementos da classe “fundo-branco” .....	8
Figura 3.3: Exemplo de manipulação do DOM por Javascript .....	9
Figura 3.4: Exemplo de manipulação do DOM por Javascript reescrito com jQuery .....	9
Figura 3.5: Exemplo de código PHP para geração de página HTML .....	10
Figura 3.6: Código HTML gerado .....	10
Figura 3.7: Tabelas exemplo de um banco de dados relacional .....	11
Figura 3.8: Diagrama de classes .....	11
Figura 3.9: Código PHP da classe AR Projeto .....	12
Figura 3.10: Código PHP da classe AR Pessoa .....	12
Figura 3.11: Utilização do modelo AR para acesso ao banco de dados .....	13
Figura 3.12: Código da classe ProjetoController .....	14
Figura 3.13: Código da View index .....	14
Figura 3.14: Hierarquia de pastas do webapp criada pelo Yii .....	15
Figura 3.15: Geração de CRUD a partir do Yii Code Generator .....	16
Figura 4.1: Árvore de funcionalidades do sistema .....	17
Figura 5.1: Diagrama Entidade-Relacionamento do sistema proposto .....	24
Figura 5.2: Notação Crow’s Foot .....	25
Figura 5.3: Diagrama Entidade-Relacionamento do banco de dados .....	29
Figura 5.4: Fluxograma de criação de páginas a partir de tabelas da base de dados .....	30
Figura 5.5: Esquema de layout do sistema .....	31
Figura 5.6: Tela de login do sistema .....	32
Figura 5.7: Tela inicial do sistema .....	33
Figura 5.8: Formulário de informações de um projeto .....	34
Figura 5.9: Formulário de seleção de equipe técnica de um projeto .....	35
Figura 5.10: Página inicial de projetos .....	36
Figura 5.11: Página de informações de um projeto .....	37
Figura 5.12: Página de financeiro de um projeto .....	37
Figura 5.13: Visão de tabelas do financeiro de um projeto .....	38
Figura 5.14: Visão de gráficos do financeiro de um projeto .....	38
Figura 5.15: Página de atividades de um projeto .....	39
Figura 5.16: Página de documentos de um projeto .....	40
Figura 5.17: Página gerencial de um projeto .....	40
Figura 5.18: Formulário de adição de permissão a um projeto .....	41
Figura 5.19: Menu de operações de um projeto .....	41
Figura 5.20: Relatório de um projeto .....	42
Figura 5.21: Página de um usuário .....	43
Figura 5.22: Menu de operações sobre atividades .....	43

Figura 5.23: Menu de configurações do sistema .....	44
Figura 5.24: Notificações do sistema.....	45
Figura 5.25: Busca do sistema.....	45
Figura 5.26: Página de grupos .....	46
Figura 5.27: Formulário de um grupo.....	47
Figura 5.28: Visão da página de atividades.....	48
Figura 5.29: Formulário de criação de atividade.....	49
Figura 5.30: Visão da página de uma atividade .....	50
Figura 5.31: Relatório geral do sistema .....	51
Figura 5.32: Página inicial da Wiki .....	52
Figura 5.33: Página de Rotinas Administrativas da Wiki .....	53
Figura 5.34: Resultados do estudo sobre usabilidade. Primeira parte .....	58
Figura 5.35: Resultados do estudo sobre usabilidade. Segunda parte .....	59
Figura 5.36: Resultados do estudo sobre usabilidade. Terceira parte.....	60
Figura 5.37: Resultados do estudo sobre usabilidade. Quarta parte .....	61

## **LISTA DE TABELAS**

Tabela 3.1: Descrição dos objetos e seus atributos.....	13
Tabela 5.1: Histórias propostas para o desenvolvimento do sistema .....	21
Tabela 5.2: Categorias de Atividades .....	26
Tabela 5.3: Análise da cobertura de requisitos.....	54

## RESUMO

Com o crescimento do investimento em projetos de pesquisa no Brasil e o conseqüente crescimento do número de projetos de pesquisa, este trabalho apresenta a concepção e o desenvolvimento de um sistema Web para auxiliar o pesquisador no gerenciamento de rotinas administrativas dos projetos. A partir da análise de requisitos das rotinas que envolvem a realização de um projeto de pesquisa e utilizando metodologias baseadas no Manifesto Ágil, o sistema demonstra o uso de tecnologias web de código aberto para o desenvolvimento de um sistema robusto e de fácil uso para gerenciar projetos, pessoas e atividades relacionadas ao pesquisador e sua equipe de trabalho.

**Palavras-Chave:** Web, internet, gerencia de projetos, sistema de apoio.

## **Researcher's Supporting System**

### **ABSTRACT**

Brazil is currently experiencing a surge in investments related to research activities. Consequently, the country has registered a growth in the number of research projects between 2004 and 2011. This paper presents the design and development of a web-based system to assist researchers in a series of tasks related to the management of administrative routines. The system is built with the support of Requirement Analysis of the cycle of a research project and Agile Manifesto-based methodologies. The paper demonstrates the use of open source web technologies to the development of a robust and easy-to-use system that encompasses the management of financial, human and material resources related to a specific research.

**Keywords:** Web, internet, project management, supporting system.

# 1 INTRODUÇÃO

Com a falta de um sistema informatizado para o gerenciamento de projetos de pesquisa pela Universidade Federal do Rio Grande do Sul (UFRGS) e com a crescente demanda de projetos de pesquisa, a complexidade de lidar com estes projetos cresce exponencialmente, em especial porque há diferentes órgãos de financiamento e também diferentes envolvidos, sejam eles pesquisadores, gestores, executores, etc. Cada um deles possui políticas diferenciadas, seja em relação à aplicação dos recursos quanto com a prestação de contas. Os órgãos de fiscalização também constantemente avaliam, inspecionam e solicitam relatórios detalhados, tornando o processo ainda mais complexo para os diferentes envolvidos, em especial para o coordenador.

Segundo dados da CAPES, em relação à execução orçamentária de 2004 a 2011, somente em bolsas de pesquisa, houve um crescimento de cerca de 872 milhões em 2008 para 2,1 bilhões de reais em 2011, representando um crescimento de aproximadamente 149% (CAPES, 2011). Segundo dados do CNPq, somente em 2012 a UFRGS deu origem a 354 novos projetos pelo CNPq (CNPQ, 2012). Com tantos projetos, se faz mais que necessário a utilização de sistemas de informação que gerenciem, de forma amigável e eficiente, as fases de um projeto de pesquisa, passando por elaboração, tramitação, andamento e prestação de contas; a equipe que atua em determinado projeto de pesquisa; as atividades de cada membro da equipe no projeto. Enfim, é necessário um sistema que auxilie na documentação e geração de relatórios dos projetos.

O presente trabalho tem como objetivo descrever a especificação e o desenvolvimento do SIPESQ (Sistema de Apoio ao Pesquisador). Como o próprio nome diz, o SIPESQ visa apoiar o pesquisador nas rotinas administrativas e demandas institucionais oriundas das diferentes fundações de amparo à pesquisa. Com uma interface web amigável e eficiente, o sistema busca integrar de forma inteligente os dados de projetos auxiliando-os em todas suas fases. A versão preliminar do sistema está sendo testada dentro do Centro de Estudos Internacionais sobre Governo (CEGOV).

Atualmente, o sistema está auxiliando mais de 20 professores do centro a gerirem as atividades cotidianas de pesquisa, diminuindo os custos de transação de administração de mais de 40 projetos de pesquisa. Entre os principais objetivos do sistema encontram-se: gestão integrada de dados pessoais do pesquisador e da sua equipe; Atividades do pesquisador e da sua equipe; Fluxo de bolsas; Projetos (controle financeiro do projeto, atividades relacionadas ao projeto, equipe, prazos e relatórios); Patrimônios e bens de capital que estão sob responsabilidade dos pesquisadores; Acervo de livros, inscrições em portais de periódicos, agenda de contatos; Rotinas administrativas; Relatórios institucionais.

No decorrer deste trabalho será mostrado como o sistema foi concebido. Sua análise de requisitos, seu desenvolvimento, as tecnologias utilizadas no sistema e por último uma

análise com os usuários do sistema para verificar a eficiência e se o sistema cumpriu sua proposta inicial.

Nesse sentido, o próximo capítulo descreve os requisitos do sistema. O Capítulo 3 descreve como o sistema foi desenvolvido, incluindo sua arquitetura e tecnologias relacionadas. O Capítulo 4 apresenta uma análise do sistema. Por fim, o último capítulo apresenta as conclusões, incluindo considerações finais, limitações e trabalhos futuros.

## 2 ANÁLISE DE REQUISITOS

Neste capítulo serão apresentadas as fases de um projeto dentro da Universidade e como podemos, com a ajuda da computação, facilitar a gerência de processos dos projetos. Como poderíamos utilizar um sistema para auxiliar os pesquisadores em todas as fases do projeto, desde Planejamento, Execução até chegar na Finalização do projeto. A partir disso, então, será definido os requisitos do sistema, tanto os requisitos funcionais quanto os não funcionais.

### 2.1 As Fases de um Projeto

Um projeto pode ser dividido em diferentes fases para oferecer um melhor controle gerencial, oferecendo enfoque adequado a cada fase do projeto. Coletivamente, essas fases são conhecidas como ciclo de vida de um projeto (PMI, 2008).

Para o sistema proposto os projetos são divididos em três ciclos Planejamento, Execução e Finalização. Cada ciclo tem seu tempo e riscos específicos e estão subdivididos em etapas.

Para a fase de Planejamento o sistema subdivide esta fase nas etapas de Elaboração, Negociação e Tramitação do projeto. Para a fase de Execução, a subdivisão da fase é composta pelas fases Em andamento: início, Em andamento: meio, Em andamento: fim. Para a fase final, o sistema a subdivide em Prestação de Contas e Encerrado.

Na fase de Planejamento do projeto é preciso especificar escopo, plano de trabalho, prazos, levantamento financeiro e acordos. Nesta mesma fase também há a definição do tipo de instrumento jurídico, fundação de apoio, tramitação interna, tramitação externa e convênios. Ou seja, nesta fase está todos os trâmites burocráticos ao se criar um projeto de pesquisa.

A fase de Execução de um projeto é a fase mais carente de um sistema computacional. Nessa fase, tem-se o gerenciamento de pessoas, custos e aquisições, documentação, atividades e produtos, ou seja, nesta fase, está a parte mais dinâmica de projeto. Nessa fase do projeto que é interessantíssimo um sistema computacional que documente todas as informações e ações do projeto e de sua equipe de modo a tirar proveito na fase final do projeto, fase de finalização. Guardando as informações em meio digital, é possível obter maior controle sobre o andamento do projeto e fácil recuperação de tais informações. É de interesse do pesquisador a utilização de um sistema que forneça rápida recuperação de informações sobre o andamento do projeto, informações sobre o financeiro deste projeto e outras informações relevantes no auxílio à conclusão do projeto. Tal sistema ajudaria o pesquisador na fase mais importante do projeto, que é fase de Finalização e prestação de contas. Um sistema que auxilie na geração de relatórios e discriminação dos gastos de um projeto é de vital importância, pois pouparia muito tempo do pesquisador, além de auxiliá-

lo a documentar corretamente seus gastos, evitando problemas futuros com a prestação de contas. Deste modo, o sistema poderia gerar relatórios, gráficos de custos e aquisições, tabelas, lista de atividades e produtos entregues no decorrer do projeto e todo tipo de dados útil na finalização do projeto.

É neste âmbito que o SIPESQ se propõe a resolver tais problemas na tramitação de projetos. A ajuda do SIPESQ seria de fundamental importância para o pesquisador, pois seu objetivo é fornecer uma interface amigável e intuitiva para a documentação de projetos. Tal sistema seria capaz fornecer um ambiente para os pesquisadores em que eles tivessem total controle sobre o andamento de projetos e suas atividades. Sendo assim, podemos criar uma lista de requisitos para tal sistema. A seguir serão listados os requisitos funcionais e não funcionais de do sistema proposto.

## **2.2 Requisitos do Sistema**

A partir da análise do ciclo de vida de um projeto dentro da Universidade e de reuniões específicas com a equipe de apoio técnico do CEGOV para definir os objetivos do e funcionalidades do sistema a ser desenvolvido, foi levantado os requisitos funcionais e não funcionais de um sistema que atendesse o mínimo de demandas necessárias para o auxílio nos projetos. Deste modo foram definidos os seguintes requisitos funcionais e não funcionais do sistema proposto.

### **2.2.1 Requisitos Funcionais**

A lista a seguir representa aquelas funcionalidades que devem obrigatoriamente estar presentes no sistema:

- O sistema deve ser principalmente baseado em três grandes áreas: Usuários do sistema; Projetos de Pesquisa e, por último, Atividades desses usuários em tais projetos.
- O sistema deve possuir gerenciamento de permissões, para que os usuários tenham níveis diferentes de acesso ao sistema. Estas permissões devem ser flexíveis, podendo ser específica de um projeto ou valer para o sistema inteiro.
- Usuários administradores devem gerenciar permissões no sistema e ter controle total para adicionar novos usuários e qualquer outro recurso do sistema.
- O sistema deve fornecer meios para o gerenciamento de projetos de pesquisa baseado em cinco pilares: Informações gerais sobre o projeto como datas de início, fim, entrega de relatório, equipe; gerenciador financeiro; controle de atividades; controle de documentos e controle de permissões do projeto.
- O sistema deve fornecer meios para o gerenciamento de pessoas. As pessoas devem poder se vincular a projetos e devem ter acesso as suas informações pessoais, financeira e atividades que realiza nos projetos em que atua.
- O sistema deve fornecer meios para o gerenciamento de atividades dos usuários dentro dos projetos em que atuam.
- O sistema deve fornecer relatórios detalhados sobre projetos e pessoas

### 2.2.2 Requisitos Não Funcionais

A seguir são listadas questões de projetos que não afetam diretamente no funcionamento final do sistema. A lista inclui questões de projeto e tecnologias escolhidas para o desenvolvimento do sistema.

- O sistema deve ser desenvolvido em ambiente web para que o usuário final possa acessar de diferentes sistemas operacionais.
- O sistema deve usar apenas ferramentas gratuitas, não envolvendo nenhum custo.
- O sistema deve ser desenvolvido nas linguagens PHP, HTML, Javascript e CSS por serem linguagens voltadas para a web e com comunidade bastante extensa, provendo uma grande variedade de documentação e frameworks para o auxílio do desenvolvimento.
- O sistema deve utilizar, quando possível, frameworks para o desenvolvimento que auxiliem na padronização do projeto, reuso posterior do projeto pela comunidade e principalmente a utilização de framework que auxiliem na coesão do sistema, de modo a aumentar sua segurança e confiabilidade.
- O sistema deve utilizar o banco de dados PostgreSQL. É um banco de dados gratuito, tem documentação abrangente e possui frameworks de suporte que auxiliam na utilização deste banco de dados com linguagens de programação como PHP.
- O sistema deve possuir curva de aprendizagem de modo que o usuário precise de o mínimo de conhecimentos de informática possível. O sistema deve ser amigável e intuitivo para qualquer público alvo.
- O sistema deve utilizar o framework MVC para PHP Yii por ser livre de custos, ter documentação abrangente, confiabilidade e ser o framework escolhido pelo CPD da UFRGS.

## 3 FERRAMENTAS UTILIZADAS

Nos capítulos anteriores foram discutidos os aspectos relevantes à análise de requisitos do sistema bem como os problemas que o sistema aborda. Neste capítulo serão apresentadas as ferramentas, linguagens e frameworks utilizados no desenvolvimento do sistema. O sistema proposto é formado por diversas ferramentas e frameworks de código aberto que tem como principal vantagem uma comunidade de desenvolvedores extensa, portanto, recebem constantes atualizações, possuem documentação robusta e fácil reuso de código. A partir deste capítulo é esperado que o leitor possua conhecimento em linguagens de programação, modelagem de dados, técnicas de construção de programas e técnicas de engenharia de software, mais especificamente as utilizadas no desenvolvimento ágil.

### 3.1 Aplicação Web

A aplicação foi desenvolvida a partir do modelo cliente-servidor (KUROSE; ROSS, 2010) em que a aplicação fica centralizada em um servidor e é acessada por clientes através de uma rede de computadores. Para o servidor foi usado o Apache (COAR; BOWEN, 2007) e para os clientes pode ser usado qualquer navegador Web.

O sistema proposto foi desenvolvido em ambiente Web seguindo os padrões da W3C<sup>1</sup>. Por ser baseado em Web temos a vantagem de ser nativamente independente de plataforma. Como o sistema roda dentro de um navegador Web, qualquer Sistema Operacional dotado de um navegador Web compatível com as especificações atualizadas da W3C fará uso pleno de todas as funcionalidades do sistema.

### 3.2 Linguagens e Tecnologias Utilizadas

O sistema proposto utiliza as linguagens interpretadas PHP, Javascript, HTML e SQL. O PHP em conjunto com o framework Yii PHP<sup>2</sup> são responsáveis por todo o modelo de negócio do sistema. Este modelo segue o esquema MVC (KRASNER; POPE, 1988) no qual é quebrado em três grandes partes, Modelo, Visão e Controlador, em que cada parte cuida de uma parte específica do sistema, facilitando a manutenção e reuso de código. O SQL é utilizado em conjunto com o banco de dados PostgreSQL (HALPIN; MORGAN, 2010) e é responsável pelo armazenamento de informações do sistema. Por fim, HTML, Javascript e CSS são responsáveis pela interface responsiva (FRAIN, 2012) com o usuário. No sistema foi utilizado o framework Twitter Bootstrap<sup>3</sup> disponibilizado pelo Twitter que possui documentação robusta e possibilita o desenvolvimento rápido de interfaces responsivas e amigáveis.

---

<sup>1</sup> Disponível em: <http://www.w3c.br/Padroes/WebDesignAplicacoes>

<sup>2</sup> Disponível em: <http://www.yiiframework.com/>

<sup>3</sup> Disponível em: <http://getbootstrap.com/>

As linguagens foram escolhidas por serem de código aberto, possuírem desempenho satisfatório e fácil aprendizado. Durante este capítulo será mostrado exemplos de utilização destas linguagens e serão mostradas as facilidades que elas impuseram no desenvolvimento deste trabalho.

Linguagens interpretadas são amplamente utilizadas por proverem uma alta produtividade com uma baixa curva de aprendizagem. Com apenas alguns passos o usuário é capaz de resolver problemas simples de programação. Por oferecerem uma extensa disponibilidade de bibliotecas e grande comunidade de código aberto é que estas linguagens foram escolhidas para o desenvolvimento do sistema.

### 3.2.1 HTML

O utiliza a linguagem de marcação de hipertextos HTML (*HyperText Markup Language*). Esta linguagem interpretada pelos navegadores web possui ampla gama de bibliotecas de código aberto bem como possui rápido aprendizado. Com a união do HTML e linguagens como CSS (*Cascading Style Sheets*) e Javascript podemos rapidamente desenvolver interfaces de usuário intuitivas e robustas, aumentando assim, o nível de usabilidade do usuário. O sistema proposto utiliza o HTML na sua quinta versão, HTML5, o CSS na sua terceira versão, CSS3 e código Javascript compatível com as normas ECMAScript<sup>4</sup> seguindo a especificação ECMA-262. Estas tecnologias são apoiadas pelo framework Bootstrap do Twitter que possui uma gama de componentes que possibilitam o desenvolvimento rápido de interfaces responsivas baseadas nas últimas tecnologias web e também ajudam na compatibilidade entre os navegadores da web.

### 3.2.2 CSS

A linguagem CSS define folhas de estilo para a marcação HTML. Com ele é possível modificar a aparência e comportamento de elementos HTML baseado em classes e identificadores. A seguir, será mostrado um exemplo de uma pequena página HTML que tem seu comportamento alterado pelo CSS.

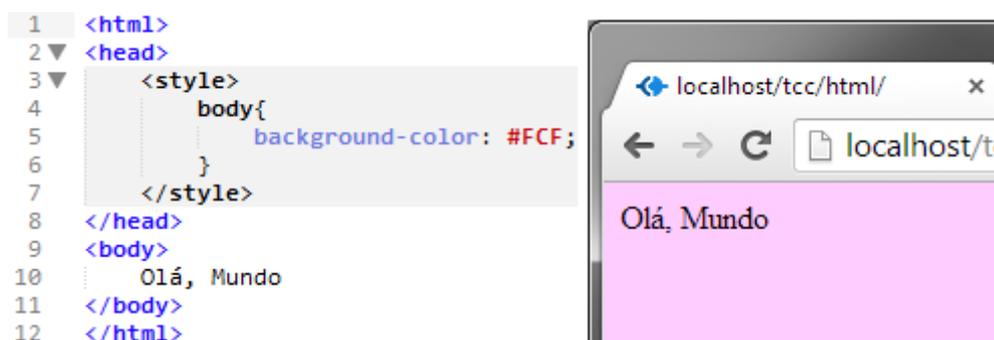


Figura 3.1: Código de exemplo de utilização de HTML em conjunto com CSS

A Figura 3.1 representa o código fonte a esquerda e o resultado exibido pelo navegador a direita. Podemos observar neste exemplo de página HTML que o estilo CSS se encontra embutido no mesmo arquivo HTML, dentro da tag <head> e da tag <style>. Também poderíamos utilizar arquivos externos para a marcação CSS. Importando-os dentro da tag <head>.

<sup>4</sup> Disponível em: <http://www.ecmascript.org/>

A figura anterior demonstra como podemos alterar o comportamento da página. No caso da imagem, alteramos a cor padrão do corpo da página, branco, para a cor definida no código a esquerda na linha 5: Definimos o `background-color` (cor de fundo) para a cor de rosa. A seguir, na Figura 3.2, definiremos uma classe, chamada `fundo-branco`, o qual terá comportamento diferenciado: todos os elementos desta classe terão o seu fundo da cor branca.

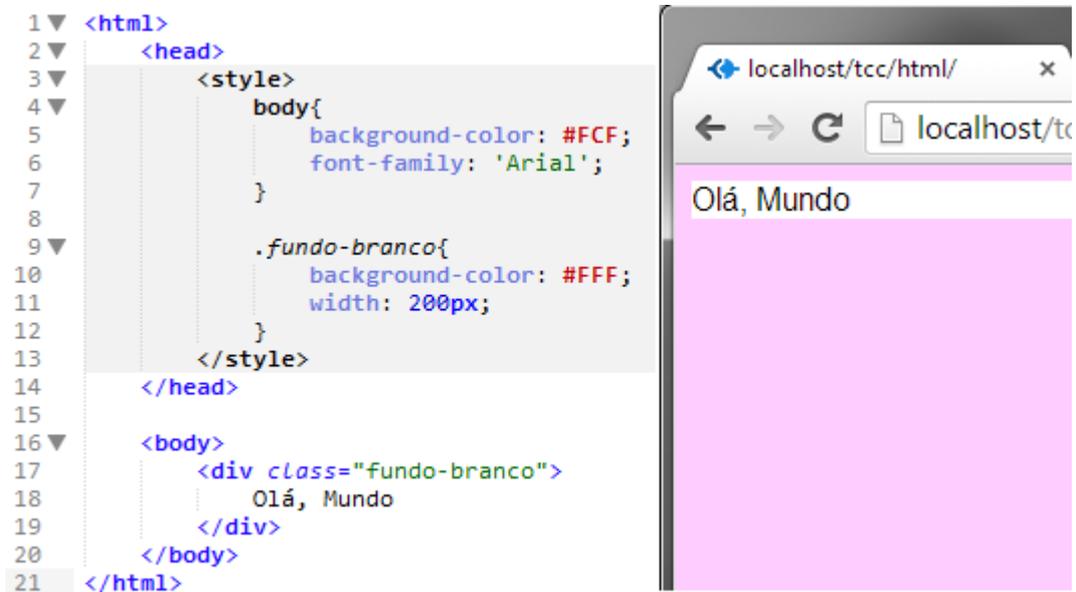


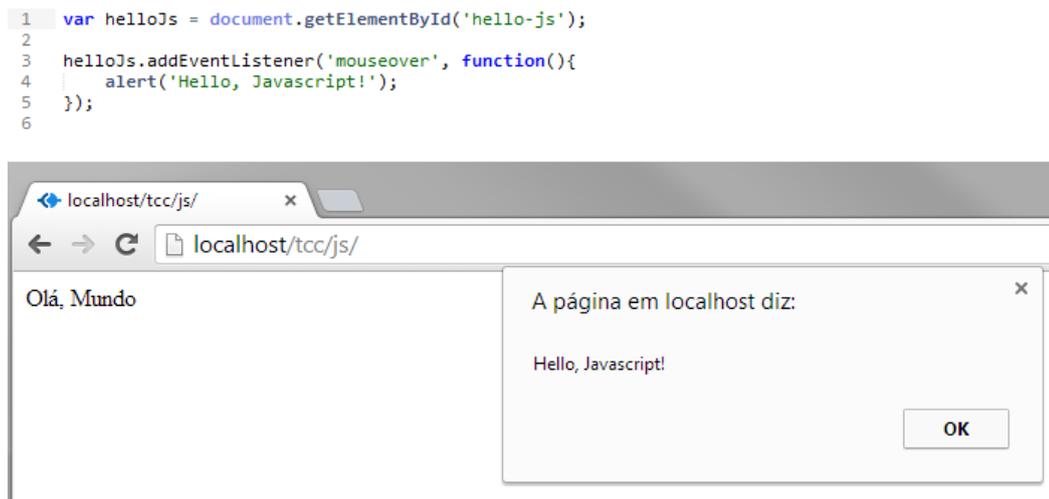
Figura 3.2: Alteração de cor de fundo de elementos da classe “fundo-branco”

É possível observar, a partir da Figura 3.2, que a inclusão da classe `fundo-branco` no elemento `<div>` (linha 17) fez com que o conteúdo dentro desse elemento ficasse com o fundo branco, pois assim foi definida a classe no CSS (linha 10). A classe também define o diâmetro deste elemento, que no caso foi de duzentos pixels.

### 3.2.3 Javascript

Javascript é uma linguagem de *scripting*, orientada a objetos, fracamente tipada que contém grandes aspectos do mundo funcional como *closures* e funções de alta-ordem quanto funcionalidades do mundo procedural. O Javascript é implementado como uma implementação da especificação do padrão de linguagem ECMAScript.

A mais comum utilização do Javascript é a manipulação dinâmica de páginas HTML através do acesso ao *Document Object Model* (DOM) que nada mais é que a representação em forma de objetos modelo de um hiperdocumento, em nosso caso, o HTML. Como o Javascript executa no navegador web do cliente, podemos obter dinamismo em nossa interface de usuário, carregando, alterando e salvando conteúdos dinamicamente sem depender da atualização da página pelo servidor. O Exemplo de código a seguir mostrará como o conteúdo de uma página pode ser alterado dinamicamente.



**Figura 3.3: Exemplo de manipulação do DOM por Javascript**

No exemplo apresentado na Figura 3.3, tem-se o código Javascript acima e a página web abaixo. Nesse exemplo, primeiramente buscamos o objeto do DOM com identificador “hello-js” (linha 1) a seguir adicionamos um evento a ele (linha 3). Neste evento ao passarmos o mouse por cima da palavra “Olá, Mundo” é exibida uma janela com a mensagem “Hello, Javascript!” como mostrado na parte de baixo da imagem. Este exemplo simples nos mostra as facilidades que o Javascript nos dá na manipulação do DOM.

No sistema proposto foi utilizado a famosa biblioteca Javascript chamada jQuery (BIBEAULT e KATS, 2008). Essa biblioteca possui diversas funcionalidades para facilitar a manipulação do DOM. Por exemplo, utilizando jQuery é possível obter o mesmo resultado do exemplo anterior com o código a seguir (Figura 3.4).

```

1 $('#hello-js').hover(function(){
2     alert('Hello, Javascript!');
3 });
4

```

**Figura 3.4: Exemplo de manipulação do DOM por Javascript reescrito com jQuery**

### 3.2.4 PHP

O PHP é uma linguagem de propósito geral *scripting*, imperativa, fracamente tipada, orientada a objetos, originalmente concebida para a criação de páginas dinâmicas web ao permitir a integração de código PHP em conjunto com o código HTML. Linguagens *scripting* vêm sendo amplamente utilizada no desenvolvimento web pela alta produtividade que possibilitam (OUSTERHOUT, 1998). Estas linguagens de programação são interpretadas, diferentemente das linguagens compiladas e isto permite uma significativa redução no tempo e complexidade do desenvolvimento de sistemas de larga escala, como também permite fácil manutenção e integração de componentes. Com a ajuda de compiladores em tempo de execução, o desempenho destas linguagens tem crescido significativamente e a utilização destas linguagens em sistemas de grande porte vem aumentando a cada dia (ZHAO *et al.*, 2012).

A seguir, será mostrado um breve exemplo da utilização de código PHP em conjunto com código HTML para a geração de uma página web simples.

The image shows a code editor on the left and a web browser on the right. The code editor contains PHP code that defines a function `printName`, sets an array `$pessoas` with names 'Jorge', 'Beto', 'Marina', and 'Joana', and generates HTML output. The browser shows the rendered page with the heading '4 Pessoas' and a list of the four names.

```

1 <?php
2     function printName($name){
3         echo $name;
4     }
5
6     $pessoas = array('Jorge', 'Beto', 'Marina', 'Joana');
7
8 <?php
9 <html>
10 <body>
11     <h3><?php echo count($pessoas)?> Pessoas</h3>
12     <ul>
13         <?php foreach($pessoas as $p): ?>
14             <li><?php printName($p) ?> </li>
15         <?php endforeach; ?>
16     </ul>
17 </body>
18 </html>

```

The browser window shows the URL `localhost/tcc/php/` and the rendered page content:

**4 Pessoas**

- Jorge
- Beto
- Marina
- Joana

**Figura 3.5: Exemplo de código PHP para geração de página HTML**

Nota-se na Figura 3.5 a utilização do PHP para a geração de uma página dinamicamente. Da linha 2 a linha 6 definimos uma função para a impressão de um nome e também um vetor de nomes. Da linha 8 até a linha 17 a página HTML é definida com código PHP embutido. O interpretador PHP então interpreta o arquivo para a geração da página resultante, que só conterá código HTML. O resultado gerado pode ser visto na Figura 3.6.

```

1 <html>
2 <body>
3     <h3>4 Pessoas</h3>
4     <ul>
5         <li> Jorge </li>
6         <li> Beto </li>
7         <li> Marina </li>
8         <li> Joana </li>
9     </ul>
10 </body>
11 </html>

```

**Figura 3.6: Código HTML gerado**

### 3.2.5 Framework Yii PHP

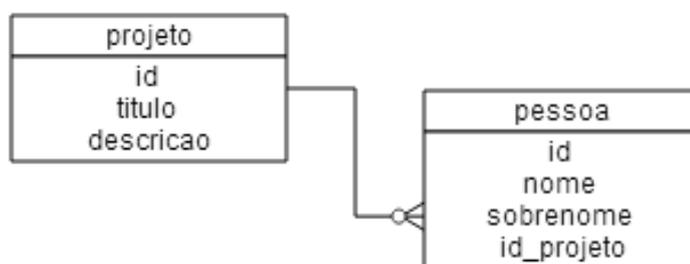
Yii é um framework de desenvolvimento escrito em PHP de alto desempenho e baseado em componentes. Ele facilita a construção de aplicações de larga escala. Possibilita máximo reuso em programação web e pode acelerar significativamente o processo de desenvolvimento (WINESETT, 2010, pág. 8).

Yii é um framework de código aberto escrito em PHP5 totalmente orientado a objetos. Ele é baseado no padrão Model-View-Controller (MVC), o qual quebra a lógica da aplicação em três camadas distintas onde o Model representa as informações da aplicação, geralmente reflete uma tabela de um banco de dados relacional e serve de camada para a manipulação de dados persistentes do sistema; A View é a camada de interface de usuário, ela lida especificamente em como as informações do sistema serão exibidas ao usuário final; Por fim temos o Controller, que cuida da lógica de negócios do sistema e faz a comunicação entre as duas primeiras (KRASNER; POPE, 1988).

No Yii este sistema MVC funciona muito bem graças aos componentes e ferramentas adicionais que o framework disponibiliza para o desenvolvedor. Neste capítulo citaremos algumas das ferramentas e componentes chaves que fazem deste framework um dos mais usados em PHP atualmente. Mostraremos exemplos de código com base em um aplicativo web construído para efeito de demonstração chamado Webapp.

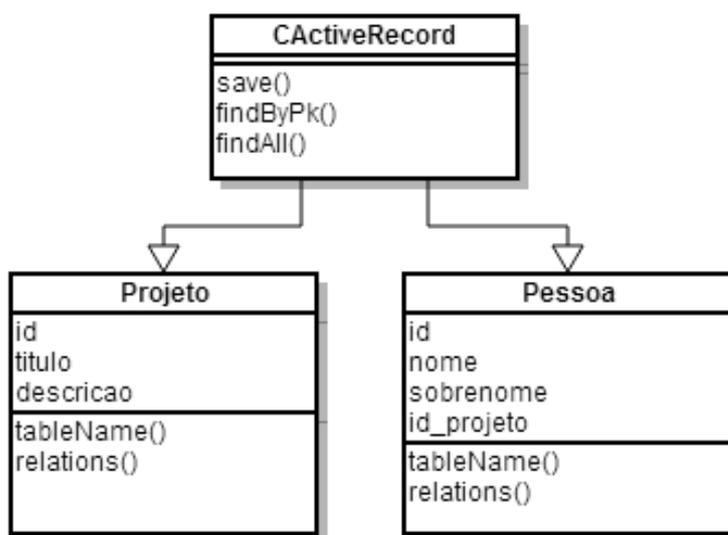
### 3.2.5.1 Active Record

O padrão de design Active Record (AR), no Yii disponibilizado por objetos que estendem a classe `CActiveRecord`, é usado para abstrair a camada de dados persistentes (neste caso um banco de dados relacional) da camada de dados dinâmicos (dados instanciados em memória) na forma orientada a objetos. Isto significa que cada objeto AR tem seus campos correspondentes às linhas de uma tabela do banco de dados relacional e suas relações. A seguir será mostrado um pequeno exemplo de como este padrão funciona. Na Figura 3.7 temos um exemplo de banco de dados que possui duas tabelas e uma relação entre estas tabelas:



**Figura 3.7: Tabelas exemplo de um banco de dados relacional**

Podemos observar a tabela projeto e pessoa com seus respectivos atributos e a relação entre estas tabelas. O campo `id_projeto` na tabela pessoa referencia a tabela projeto configurando uma relação em que uma pessoa obrigatoriamente possui um projeto e um projeto pode ter zero ou mais pessoas. Os modelos deste exemplo, no Yii, serão representados pelo diagrama de classes apresentado na Figura 3.8.



**Figura 3.8: Diagrama de classes**

Esse diagrama mostra o funcionamento a modelagem do AR. Temos as classes Projeto e Pessoa o qual estendem a classe `CActiveRecord`. Ambas as classes possuem os métodos `tableName` e `relations`. Esses métodos definem qual a tabela do banco o objeto representa e quais são suas relações, respectivamente (MAKAROV, 2011).

A partir disso é possível criar classes em PHP para estes objetos como ilustrado nas Figuras 4.9 e 4.10. Os objetos instanciados a partir destas classes terão atributos como ilustrado na Tabela 3.1.

```

1  <?php
2  class Projeto extends CActiveRecord
3  {
4
5      ...
6
7      public function tableName()
8      {
9          return 'projeto';
10     }
11
12     public function rules()
13     {
14         return array(
15             array('titulo, descricao', 'required'),
16         );
17     }
18
19     public function relations()
20     {
21         return array(
22             'pessoas' => array(self::HAS_MANY, 'Pessoa', 'id_projeto'),
23         );
24     }
25
26     ...
27 }

```

Figura 3.9: Código PHP da classe AR Projeto

Note que, nesse código, o retorno do método *relations* define as relações deste objeto, que neste caso só possui uma relação (pessoas). Isto fará com que a instancia destes objetos tenha um vetor de objetos Pessoa que representará todas as pessoas relacionadas ao objeto. Também observamos o método *rules* que define regras para os atributos (neste caso, tornando obrigatório o preenchimento dos campos “titulo” e “descricao”).

```

1  <?php
2  class Pessoa extends CActiveRecord
3  {
4
5      ...
6
7      public function tableName()
8      {
9          return 'pessoa';
10     }
11
12     public function rules()
13     {
14         return array(
15             array('nome, sobrenome, id_projeto', 'required'),
16             array('id_projeto', 'numerical', 'integerOnly'=>true),
17         );
18     }
19
20     public function relations()
21     {
22         return array('projeto' => array(self::BELONGS_TO, 'Projeto', 'id_projeto'));
23     }
24
25     ...
26 }

```

Figura 3.10: Código PHP da classe AR Pessoa

Igualmente como acontece na classe Projeto, na classe Pessoa define-se o código necessário para a definição da tabela do banco de dados, relações e regras para os atributos. A Tabela 3.1 ilustra como ficam os atributos destes objetos:

Tabela 3.1: Descrição dos objetos e seus atributos

Projeto	
Atributo	Tipo de Dado
id	Inteiro
titulo	String
descrição	String
peessoas	Vetor de objetos do tipo Pessoa
Pessoa	
id	Inteiro
nome	String
sobrenome	String
projeto	Objeto do tipo Projeto

Após a criação dessas classes podemos acessar e manipular os dados do banco de forma simples e rápida como mostrado no código da Figura 3.11.

```

1  <?php
2  ...
3
4  $projeto = new Projeto();
5  $projeto->id = 100;
6  $projeto->titulo = "SIPESQ - Sistema de Apoio ao Pesquisador";
7  $projeto->descricao = "Sistema web que auxilia projetos, atividades e pessoas";
8  $model->save(); //Salva no banco de dados
9
10 ...
11
12 $pessoa = new Pessoa();
13 $pessoa->id = 1;
14 $pessoa->id_projeto = 100; //vincula o projeto SIPESQ
15 $pessoa->nome = "Guilherme";
16 $pessoa->sobrenome = "Severo";
17 $pessoa->save(); //Salva no banco de dados
18
19 ...
20
21 $pessoa = Pessoa::model()->findByPk(1);
22 $projeto = Projeto::model()->findByPk(100);
23
24 echo $pessoa->nome; //saída: Guilherme
25 echo $pessoa->projeto->titulo; //saída: SIPESQ - Sistema de Apoio ao Pesquisador
26 echo $projeto->peessoas[0]->nome; //saída: Guilherme

```

Figura 3.11: Utilização do modelo AR para acesso ao banco de dados

### 3.2.5.2 Controllers

Os *controllers* do Yii definem as rotas e o modelo de negócios do aplicativo. Todo *controller* deve estender a classe base do framework CController.

O Yii define as rotas do aplicativo da forma `webapp/controller/action/parâmetros`. Por exemplo, suponha que tenhamos a classe `ProjetoController` que estende a classe base `CController`. Nosso aplicativo então terá rotas do tipo `http://sipesq/projeto`. Ao definirmos métodos com o prefixo *action* na classe `ProjetoController`, por exemplo `actionCreate`, nosso aplicativo terá a rota `http://webapp/projeto/create`. E dentro do método `actionCreate` podemos definir as regras de negócios, como a manipulação de objetos e dados do banco, renderização de páginas, etc. A Figura 3.12 ilustra um trecho de código de um controlador para o nosso aplicativo de exemplo.

```

1  <?php
2  class ProjetoController extends CController
3  {
4      public function actionIndex()
5      {
6          $projects = Projeto::model()->findAll();
7          $this->render('index',array('projects'=>$projects));
8      }
9  }

```

Figura 3.12: Código da classe ProjetoController

Neste código podemos observar no método `actionIndex` a seguinte regra de negócio. O aplicativo busca no banco de dados através da ajuda do modelo AR todos os registros de projeto. Após isto, ele renderiza a View `index` passando como parâmetro todos os objetos encontrados.

```

1  <h1> Projetos </h1>
2  <ul>
3      <?php foreach($projects as $p): ?>
4          <li><?php echo $p->titulo; ?></li>
5      <?php endforeach; ?>
6  </ul>
7

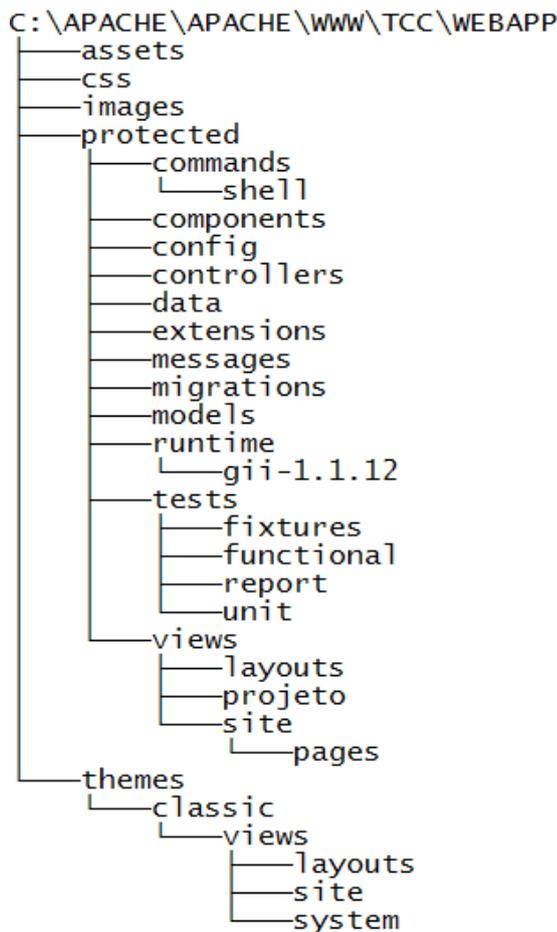
```

Figura 3.13: Código da View index

Na Figura 3.13 é ilustrado como poderíamos exibir os projeto na View `index`. Como passamos o parâmetro `projects` no método `render` de nosso *Controller*, esta variável estará visível na View. No caso do exemplo da figura seria exiba uma lista não ordenada com os títulos de cada projeto encontrado.

### 3.2.5.3 Yii Code Generator (Gii)

Até agora foi comentado como cada componente do framework funciona e como ele pode ser utilizado. O Yii oferece muito mais ferramentas para a utilização destes componentes. A forma mais rápida de construir uma aplicação é utilizando o gerador de código do Yii. Com este gerador é possível criar uma aplicação modelo em poucas linhas de comando. Em nosso exemplo criamos o aplicativo webapp. Para isso utilizamos o comando webapp do Yii “`yiic webapp <app_path> [<VCS>]`”. Este comando cria um webapp na pasta selecionada. O parâmetro VCS é opcional e define o controle de versão do aplicativo. Neste exemplo foi o utilizado o controle de versões Git. Ao final deste comando o Yii gerará uma página web modelo já no padrão MVC com a hierarquia de pastas como mostrada na Figura 3.14.



**Figura 3.14: Hierarquia de pastas do webapp criada pelo Yii**

Com o aplicativo gerado o Yii disponibiliza, além da linha de comando uma interface gráfica que pode ser acessada a partir do webapp. A interface gráfica é robusta e muito fácil de usar. Nesta interface é possível obter geração automatizada de código para a aplicação. As principais ferramentas que o Gii disponibiliza são geração de Controllers, CRUD, Formulários, Modelos e Módulos. A seguir será mostrada a utilização da interface Web do Gii para a criação de um CRUD referente à tabela projeto do banco de dados do sistema exemplo.

The screenshot shows the Yii Code Generator interface. On the left, there is a sidebar with a 'Generators' menu containing links for 'Controller Generator', 'Crud Generator', 'Form Generator', 'Model Generator', and 'Module Generator'. The main area is titled 'Crud Generator' and contains the following information:

- Description:** This generator generates a controller and views that implement CRUD operations for the specified data model.
- Fields with \* are required. Click on the highlighted fields to edit them.**
- Model Class \*:** Projeto
- Controller ID \*:** projeto
- Base Controller Class \*:** Controller
- Code Template \*:** default (C:\Apache\apache\www\yii\framework\yii\generators\crud\templates\default)
- Buttons:** Preview, Generate
- Table of Generated Files:**

Code File	Generate <input checked="" type="checkbox"/>
<a href="#">controllers\ProjetoController.php</a>	new <input checked="" type="checkbox"/>
<a href="#">views\projeto\_form.php</a>	new <input checked="" type="checkbox"/>
<a href="#">views\projeto\_search.php</a>	new <input checked="" type="checkbox"/>
<a href="#">views\projeto\_view.php</a>	new <input checked="" type="checkbox"/>
<a href="#">views\projeto\admin.php</a>	new <input checked="" type="checkbox"/>
<a href="#">views\projeto\create.php</a>	new <input checked="" type="checkbox"/>
<a href="#">views\projeto\index.php</a>	new <input checked="" type="checkbox"/>
<a href="#">views\projeto\update.php</a>	new <input checked="" type="checkbox"/>
<a href="#">views\projeto\view.php</a>	new <input checked="" type="checkbox"/>

**Figura 3.15: Geração de CRUD a partir do Yii Code Generator**

CRUD é acrônimo para Create, Retrive, Update, Delete. Ele define operações básicas feitas em uma tabela de banco de dados como busca, adição, edição e deleção de linhas de uma tabela. No caso do Gii, ele cria, a partir do modelo especificado (Projeto), o controlador e seus métodos e também as *views* relacionadas a este controlador com formulários, página de criação, edição e administração desta tabela. Com isso fica muito fácil construir um aplicativo no Yii, pois ele diminui expressivamente a escrita de código além de fornecer documentação robusta, muitas extensões e exemplos de código na comunidade. Por estes motivos que o Yii foi o framework selecionado para o sistema proposto.

## 4 O SISTEMA DE APOIO AO PESQUISADOR (SIPESQ)

A proposta do sistema é de gerenciar as rotinas administrativas dos projetos tendo como foco os pesquisadores e suas equipes de trabalho. É por esse motivo que o sistema subdivide-se, principalmente, em três grandes áreas ligadas entre si (Pessoas, Atividades e Projetos), uma área de Acervo que inclui subpartes como Gestão de Livros, Agenda de Horários, Contatos, Inscrições e uma Wiki. Por fim, o sistema deverá propor uma parte gerencial onde deverá ser possível a geração de relatórios das grandes partes do sistema como projetos, atividades e pessoas; possibilitar o gerenciamento de permissões dentro do sistema por meio de grupos de usuários; possibilitar o gerenciamento de parâmetros do sistema como rubricas, categorias de atividades e quaisquer outros parâmetros utilizados em outras áreas do sistema.

A figura seguinte demonstra uma árvore com as funcionalidades do sistema proposto:

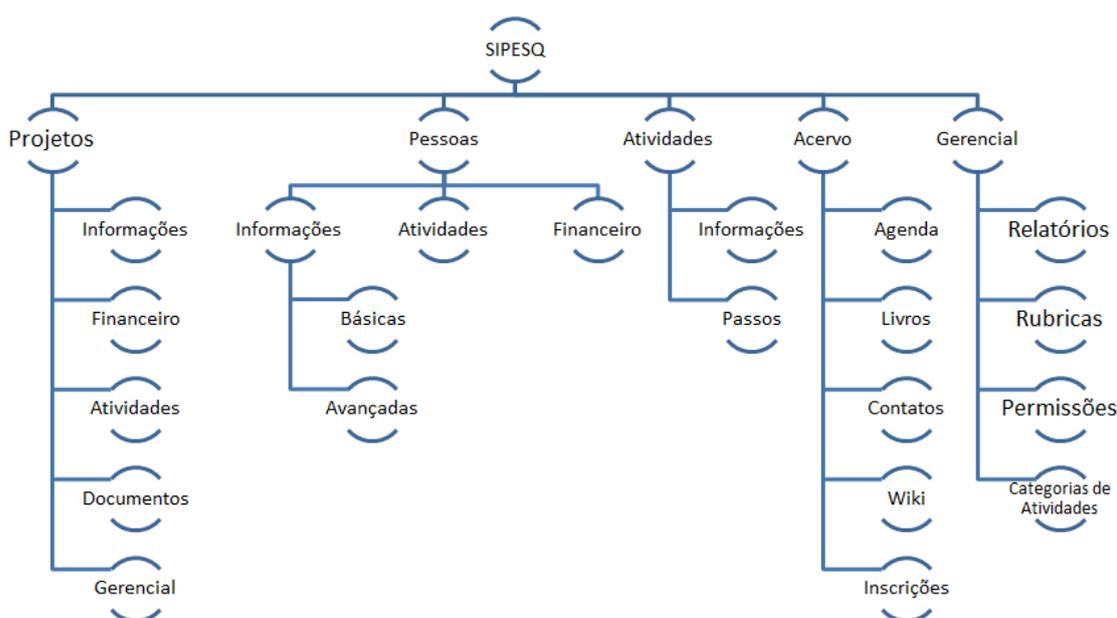


Figura 4.1: Árvore de funcionalidades do sistema

### 4.1 Descrição das funcionalidades

A seguir será apresentada a descrição de cada funcionalidade a integrar o sistema proposto. A partir da análise de requisitos foram listadas cinco funcionalidades essenciais do sistema: Projetos, Pessoas, Atividades, Acervo e Gerencial.

#### **4.1.1 Projetos**

Responsável pela gerência de projetos. O pesquisador deve ser capaz de criar projetos e, dentro de cada projeto, deve ser capaz de determinar suas informações básicas, seus participantes, coordenadores e colaboradores; determinar suas atividades e seus prazos; gerenciar a parte financeira do projeto, seus orçamentos, receitas, despesas e patrimônios; gerenciar o andamento do projeto se está em fase de Elaboração, Tramitação, Prestação de Contas, etc.; gerenciar documentos relativos ao projeto como notas fiscais, documentos assinados, contratos, etc.; deve ser capaz de gerenciar as permissões que cada participante terá dentro projeto; e, por último e mais importante, deverá ser capaz de gerar de relatórios de um projeto. Nestes relatórios deverão constar todas as atividades do projeto, sua equipe, seus patrimônios adquiridos, o detalhamento financeiro do projeto com gráficos e tabelas para fácil entendimento do pesquisador. Contudo, a seção de projetos deverá ser a principal área do sistema, pois ela agregará dados de todas as outras grandes áreas do sistema como Pessoas e Atividades. Sendo assim, é vital que esta área receba maior atenção e esteja de acordo com as especificações apresentando um ambiente amigável, robusto e rápida recuperação de informação.

#### **4.1.2 Pessoas**

Responsável pela gerência de pessoas. O pesquisador deve ser capaz de adicionar pessoas ao sistema. Cada pessoa cadastrada será também um usuário do sistema. O pesquisador deverá ter acesso irrestrito as informações de outros usuários enquanto que um usuário comum deverá ter acesso irrestrito apenas as suas próprias informações e acesso básico a informações de outras pessoas a menos que ganhe permissão para acessar outras áreas do sistema. Cada pessoa poderá gerenciar suas atividades, seus projetos e seus pagamentos. A página de pessoas, então, será dividida em subáreas: Informações, que deverá conter os dados pessoais da pessoa e projetos em que atua; Atividades, que deverá permitir que o usuário gerencie suas atividades; Financeiro, que deverá permitir que o usuário acompanhe seus pagamentos como bolsas, reembolsos, pagamento de passagens, etc.

#### **4.1.3 Atividades**

Responsável pela gerência das atividades do sistema. O pesquisador deve ser capaz de gerenciar as atividades de todos os usuários do sistema. Desta forma, é possível ter controle e dimensionamento da força de trabalho exercida pelo seu grupo de pesquisa. Nesta área do sistema deve ser possível a atribuição de atividades a determinados usuários bem como a geração de relatório de atividades de cada usuário e projeto. Na visão do usuário ele deverá ser capaz de gerenciar as suas próprias atividades e visualizar as atividades de outras pessoas.

#### **4.1.4 Acervo**

Dentro de acervo temos pequenas subáreas de vital importância para o sistema. São elas:

#### **4.1.5 Contatos**

Nesta área o sistema deverá permitir a criação, busca e edição de contatos úteis para o grupo de pesquisa.

#### **4.1.6 Wiki**

Esta área deverá conter um sistema de Wiki, como por exemplo, o Media Wiki. Este sistema deverá ser parecido com o famoso site enciclopédia Wikipedia<sup>5</sup>, onde nele será cadastrado rotinas de trabalho relevantes, permitindo assim o reuso de rotinas pelos usuários. Nestas rotinas poderemos ter, por exemplo, as informações de como tramitar um projeto na FAURGS, como tramitar um projeto no CNPq, manuais e informações sobre o próprio sistema, etc.

#### **4.1.7 Gerencial**

Responsável pela gerência do sistema, seus parâmetros e fácil acesso a geração de relatórios. Nesta área o pesquisador deve ser capaz de controlar diversos parâmetros do sistema. Para começar, ele deve ter fácil acesso a geração dos relatórios disponíveis do sistema. Depois disto, ele deverá ter acesso a criação de grupos de usuários, nestes grupos o pesquisador deve ser capaz de definir permissões para os usuários em determinadas áreas do sistema, assim sendo, ele será capaz de customizar áreas de atuação para grupos específicos de usuários. Por final, ele deve ser capaz de gerenciar as rubricas, que são acessíveis através do financeiro dos projetos e a categoria de atividades, que serão usadas para a customização de atividades dos usuários.

---

<sup>5</sup> <http://pt.wikipedia.org/>.

## 5 O SISTEMA DESENVOLVIDO

No capítulo anterior foram mostradas as tecnologias utilizadas bem como uma breve descrição da utilização de cada tecnologia. Também foi apontado o motivo da escolha de cada tecnologia para sua aplicação no sistema proposto. Este capítulo tem o propósito de mostrar o desenvolvimento do sistema em conjunto com as tecnologias mencionadas no capítulo anterior.

Para tanto, neste capítulo serão mostrados detalhes de desenvolvimento do sistema. Será discutido como cada ferramenta foi utilizada e com que bases o sistema foi desenvolvido, como métodos de engenharia utilizados e decisões de desenvolvimento. Também serão mostradas telas do sistema em funcionamento e uma breve explicação das funcionalidades do sistema tendo em vista tais ilustrações.

### 5.1 Metodologia Utilizada

O desenvolvimento do sistema não se dá por uma metodologia padrão de desenvolvimento, porém foram utilizadas ferramentas que o desenvolvimento ágil oferece e foram seguidos os princípios do manifesto ágil (BECK *et al.*, 2001), como documentação menos extensa e maior preocupação com entregas frequentes de software funcional, colaboração com o cliente e software responsivo a mudanças no planejamento.

Para tanto o sistema foi desenvolvido com iterações focadas em partes específicas do sistema como, por exemplo, cadastro de pessoas, cadastro e gerenciamento de atividades, etc. A cada iteração foram feitas reuniões quinzenais para avaliar o desenvolvimento do sistema, validar os requisitos implementados, especificar mudanças no sistema, novos requisitos e novas funcionalidades.

Nas iterações de desenvolvimento foi utilizado o conceito de *User Stories*. Conceito introduzido em conjunto com as metodologias ágeis facilita o levantamento de requisitos do sistema. Basicamente narramos requisitos de usuários em variados papéis no sistema. Podemos também definir níveis de complexidade de implementação e níveis de importância destas histórias para o sistema.

Na Tabela 5.1 é apresentada uma lista de histórias utilizadas no desenvolvimento do sistema proposto. A tabela tem colunas que descrevem cada linha e serão explicadas a seguir:

- # (numeral): Identifica unicamente a história. Não tem relação com os outros campos. É utilizada apenas para referência.
- V (Valor): Número inteiro entre 1 e 5 que representa a importância desta história para o sistema.

- C (complexidade): Número inteiro entre 1 e 5 que representa a complexidade de desenvolvimento do sistema. Em combinação com a coluna valor é útil para priorizar histórias e até mesmo excluir histórias do desenvolvimento do sistema.
- No papel de: Representa o papel alvo da história no momento da ação. Pode ser, por exemplo, um usuário, um administrador do sistema, etc.
- Quero: Representa o serviço esperado do sistema pelo papel alvo.
- Para: Representa o objetivo do papel alvo ao realizar a ação. Este campo pode ser opcional.

Tabela 5.1: Histórias propostas para o desenvolvimento do sistema

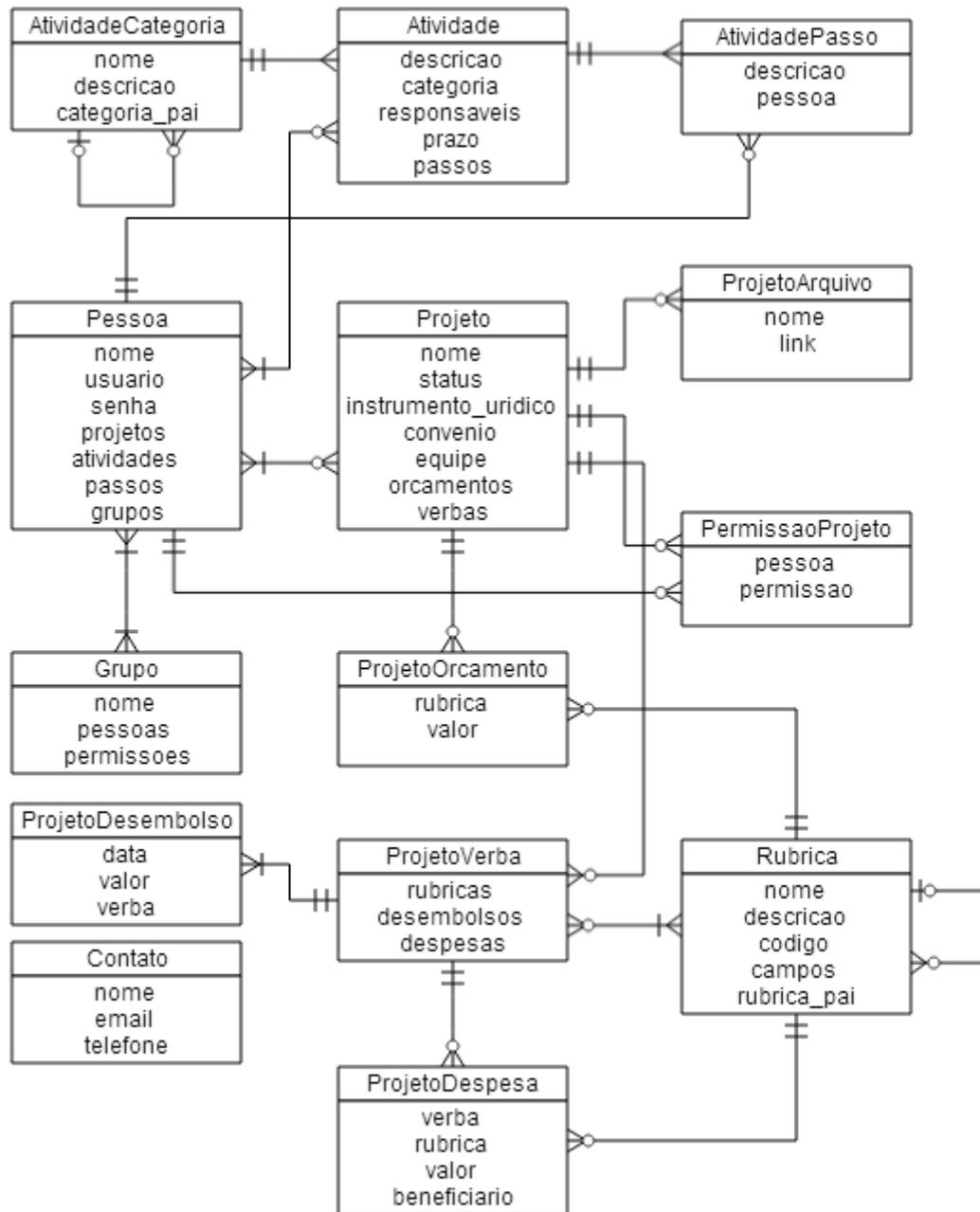
#	V	C	No papel de	...quero	...para
1	5	4	Administrador	Ter acesso privilegiado a grupos do sistema	Gerenciar grupos e permissões dos usuários no sistema
2	4	2	Administrador	Ter acesso a rubricas do sistema	Gerenciar rubricas que serão usadas na área financeira dos projetos
3	4	2	Administrador	Ter acesso às categorias de atividades	Gerenciar categorias de atividades que serão usadas no cadastro de atividades
4	3	2	Administrador	Ter acesso a relatório do sistema	Saber dados numéricos relevantes de projetos, financeiro e atividades cadastradas no sistema.
5	5	1	Usuário	Cadastrar-me no sistema	Entrar no sistema
6	5	1	Usuário	Entrar no sistema	Usar as funcionalidades
7	3	1	Usuário	Visualizar meu perfil	Ter acesso as minhas informações, meus projetos, atividades e pagamentos.
8	3	5	Usuário	Visualizar minhas notificações	Ficar atento as mudanças em minhas atividades e projetos
9	3	2	Usuário	Buscar e ter acesso às informações básicas de outros usuários	
10	4	2	Usuário	Visualizar os projetos cadastrados no sistema	
11	4	2	Usuário	Visualizar as atividades cadastradas no sistema	
12	3	1	Usuário	Visualizar agenda de horários	Saber o horário dos outros usuários
13	4	3	Usuário	Gerenciar agenda de horários	Poder colocar meu horário na agenda

14	2	2	Usuário	Gerenciar minhas férias	Avisar aos outros usuários meu período de férias
15	4	2	Usuário	Acessar acervo de livros	Saber os livros disponíveis e pegar emprestada alguma unidade
16	3	2	Usuário	Visualizar contatos cadastrados no sistema	
17	3	1	Usuário	Visualizar inscrições cadastradas no sistema	
18	5	4	Usuário	Acesso ao MediaWiki	Acessar informações relevantes como pautas de reuniões, rotinas administrativas e ofícios.
19	2	1	Usuário	Visualizar grupos cadastrados no sistema	Verificar membros dos grupos e permissões que os grupos possuem
20	4	3	Usuário	Buscar projetos, contatos e usuários	Ter acesso rápido as principais informações do sistema
21	5	3	Coordenador de Projeto	Gerenciar os projetos que coordeno	Ter acesso as informações avançadas dos projetos como financeiro e gerencial
22	5	5	Coordenador de Projeto	Acessar a seção financeira dos projetos que coordeno	Poder gerenciar os gastos do projeto
23	5	5	Coordenador de Projeto	Acessar a seção gerencial dos projetos que coordeno	Gerenciar permissões de outros usuários
24	5	2	Coordenador de Projeto	Editar projetos que coordeno	Gerenciar informações do projeto como equipe, orçamentos, etc
25	3	1	Coordenador de Projeto	Apagar projetos que coordeno	Poder excluir um projeto que precise ser excluído
26	5	1	Coordenador de Projeto	Ter acesso à equipe dos projetos que coordeno	Gerenciar membros do projeto
27	5	3	Coordenador de Projeto	Ter acesso às atividades dos projetos que coordeno	Gerenciar atividades para os membros do projeto
28	4	3	Coordenador de Projeto	Ter acesso aos documentos dos projetos que coordeno	Gerenciar documentos relativos ao projeto
29	5	5	Coordenador de Projeto	Gerar relatórios completos de projetos	
30	4	2	Membro de Projeto	Acessar informações dos projetos que participo	
31	3	2	Membro de Projeto	Gerenciar atividades dos projetos que participo	
32	3	2	Membro de Projeto	Gerenciar documentos do projeto	Auxiliar os coordenadores do projeto na gerência dos documentos

33	5	2	Responsável de Atividade	Gerenciar informações de atividade	Alterar uma atividade ou criar um passo para a mesma
34	5	2	Responsável de Atividade	Gerenciar participantes da atividade	Gerenciar equipe participante da atividade
35	5	2	Participante de Atividade	Acessar informações da atividade	Poder editar informações da atividade para auxiliar o coordenador da atividade
36	3	2	Participante de Atividade	Cadastrar passos na atividade	Cadastrar passos relativos a conclusão desta atividade
37	5	5	Membro de Grupo	Herdar permissões de grupos que participo	Acessar seções que os grupos que participo têm acesso

## 5.2 Modelagem da Base de Dados

A partir das histórias da Tabela 5.1 é possível elaborar um diagrama de relacionamentos do sistema para posterior planejamento da base de dados e da codificação do sistema em si. Assim sendo, o diagrama Entidade-Relacionamento do sistema é ilustrado na Figura 5.1.



**Figura 5.1: Diagrama Entidade-Relacionamento do sistema proposto**

A Figura 5.1 mostra um diagrama na notação *Crow's Foot* (HALPIN; MORGAN, 2008) onde a cardinalidade se dá por símbolos localizados nas extremidades das relações. Essa cardinalidade é representada por dois símbolos que representam a cardinalidade máxima, símbolo mais extremo, e mínima, símbolo menos extremo. A Figura 5.2 demonstra os símbolos e cardinalidades propostas na notação *Crow's Foot*.



Figura 5.2: Notação Crow's Foot

Ao usar a relação ilustrada na Figura 5.2 para relacionar duas entidades, a entidade da esquerda teria pelo menos uma ou mais unidades da entidade da direita, enquanto que a entidade da direita teria zero ou até uma entidade da esquerda.

### 5.3 Entidades da Base de Dados

A seguir serão explicados as entidades propostas na modelagem feita no diagrama da Figura 5.1 e seus principais atributos.

#### 5.3.1 Pessoa

A entidade Pessoa define uma pessoa no sistema. Em seus atributos essa entidade guarda informações de login, como usuário e senha; informações pessoais, como nome completo, nome da mãe, CPF, RG, endereço, telefone e e-mail; informações acadêmicas, como currículo lattes, número de inscrição, etc.

A entidade também possui relações com outras entidades, dentre elas pode-se obter os projetos de uma pessoa, suas atividades, seus passos e seus grupos.

#### 5.3.2 Grupo

A entidade grupo é formada por uma lista de entidades Pessoa, um nome, uma descrição e um objeto do formato JSON que guarda informações sobre permissões. Nesse objeto constam informações como seção do sistema, subseção e nível de permissão nessa subseção. O sistema usa essa informação para permitir ou negar certas ações dos usuários no sistema. Desse modo, oferece-se um sistema de permissões robusto e flexível no qual pode-se definir grupos de usuário com permissões específicas. Por exemplo, define-se o grupo Administradores, em que os usuários desse grupo herdam permissões máximas em todas as seções do sistema. Pode-se, também, definir o grupo Financeiro, do qual poderiam ter acesso somente a área financeira dos projetos e assim por diante.

#### 5.3.3 Atividade

Usada para controlar as atividades dos usuários nos projetos essa entidade possui atributos como nome, categoria, responsável, participantes, projetos, passos e descrição. Sua funcionalidade é, ao final de um projeto, relatar as atividades exercidas no período de vigência do projeto.

Uma atividade pode ter zero ou mais passos. Estes passos são subatividades com a funcionalidade de auxiliar a organização da atividade. Vale ressaltar que os passos não aparecem no relatório final, por isso, são flexíveis para a organização do trabalho na atividade.

### 5.3.4 AtividadeCategoria

Define a categoria de uma entidade do tipo Atividade. As categorias são definidas em dois níveis e foram inspiradas nos tipos de atividades propostas no currículo lattes. A tabela 5.2 ilustra as categorias disponíveis no sistema.

**Tabela 5.2: Categorias de Atividades**

<b>Categoria Primária</b>	<b>Categoria Secundária</b>
Administrativa	Administrativa
Atuação Profissional	Pesquisa e Desenvolvimento Direção e Administração Estágio Serviço Técnico Especializado Treinamento Outra Atividade Técnico-Científica Conselhos, Comissões e Consultoria Ensino Graduação Ensino Pós-Graduação
Banca Examinadora e Comissão Julgadora	Mestrado Doutorado - Final Doutorado - Qualificação Especialização Graduação Avaliação de Cursos Outras Participações
Eventos	Participação em Eventos Organização de Evento
Orientação	Mestrado Doutorado Iniciação Científica Graduação Orientação de Outra Natureza
Produção Bibliográfica	Livro Capítulo de Livro Texto em Jornais ou Revistas Trabalho completo/resumo publicado em anais de congresso Apresentação de Trabalho Demais tipos de produção bibliográfica Artigo
Produção Técnica	Trabalho técnico Demais tipos de produção técnica Software
Projetos de Pesquisa	Elaboração de Projetos Levantamento de fundos Revisão de Literatura Compilação de Fontes Coleta de Dados Análise de Dados/Fontes Elaboração de Textos de Discussão

Categoria Primária	Categoria Secundária
	Internos Elaboração de Apresentação Elaboração de Dossiê/Clipping Seminários de Discussão do Projeto (Reunião) Elaboração de Relatório Técnico Final Prestação de Contas do Projeto Gestão do Projeto de Pesquisa

### 5.3.5 AtividadePasso

Cada entidade do tipo Atividade pode ter uma lista de passos. Os passos são úteis para auxiliar o andamento da atividade, eles descrevem os passos necessários para a conclusão da atividade. Quando uma atividade tem todos os seus passos concluídos então ela também é marcada como concluída.

### 5.3.6 Projeto

A entidade Projeto define um projeto de pesquisa dentro do sistema. É a principal entidade do sistema. Seus atributos guardam informações como nome do projeto, vigência, instrumento jurídico, convênio, atividades, orçamentos, receitas, despesas, equipe, coordenadores, documentos e permissões.

### 5.3.7 ProjetoArquivo

Define um arquivo dentro de uma entidade projeto.

### 5.3.8 Rubrica

Define a rubrica de um orçamento, receita ou despesa. É a partir dela que ocorre a classificação da parte financeira de um projeto. As rubricas do sistema possuem formato de árvore para melhor manuseio orçamentário de um projeto. Sendo assim, ao adicionar uma receita de rubrica X para um projeto, o projeto poderá ter despesas de qualquer Rubrica Y filha de X. Por exemplo, ao definirmos uma receita com a rubrica Custeio, que tem como filhas as rubricas Material de Consumo, Passagem e Diárias, o projeto poderá ter despesas de rubrica Custeio, Material de Consumo, Passagem ou Diárias.

### 5.3.9 ProjetoOrçamento

Define um orçamento de um projeto. É uma estimativa de quanto o projeto pode receber de verba. Cada orçamento deve indicar a rubrica do qual pertence.

### 5.3.10 ProjetoVerba

Define uma receita de um projeto. Esta receita pode ter desembolsos o qual incrementam o valor disponível para a mesma e pode ter despesas o qual decrementa o valor disponível. Uma verba pode ter uma ou mais rubricas disponíveis, sendo que cada rubrica terá o limite teórico de gasto estipulado pelo valor do que foi orçamentado a ela.

Um projeto pode ter diversas verbas e cada verba pode ter diversas despesas e desembolsos.

### **5.3.11 ProjetoDesembolso**

Define um desembolso orçamentário a uma verba. Seus atributos são valor, descrição e data de desembolso.

### **5.3.12 ProjetoDespesa**

Define uma despesa relacionada a uma verba de projeto. Cada despesa pode ter uma rubrica específica. Vale salientar que, dependendo da rubrica, os atributos de uma despesa mudam, ou seja: a rubrica escolhida para a despesa é que define o conjunto de atributos de uma despesa. Por exemplo, ao incluir uma despesa na rubrica "Auxílio Financeiro a Estudantes" o usuário preencherá os campos: valor, data, vigência, recibos, contrato de bolsa, relatório de atividades e contrato de bolsa assinado, ao incluir uma despesa na rubrica "Diária" o usuário deverá preencher os campos: valor, data, vigência, destino, data de ida, data de volta, relatório de missão cumprida assinado e motivo. O sistema fornece um ambiente flexível no cadastramento de despesas.

### **5.3.13 PermissaoProjeto**

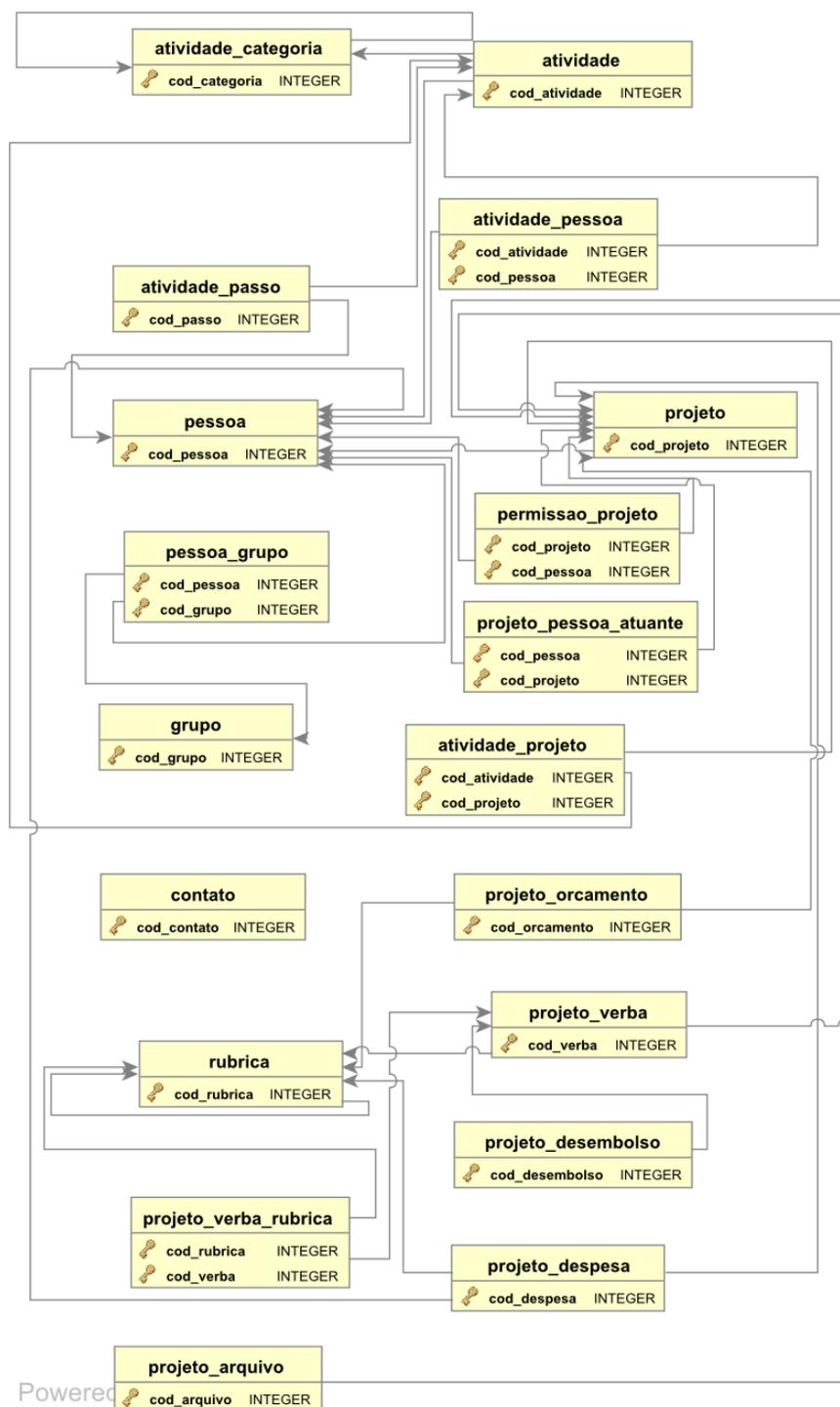
Define permissões customizadas ao projeto. O coordenador do projeto ou os administradores do sistema podem gerenciar tais permissões. Com isso, pessoas de fora da equipe do projeto podem receber permissões para editar informações do projeto.

### **5.3.14 Contato**

A entidade Contato serve para auxiliar os usuários do sistema. Os usuários podem cadastrar contatos relevantes no sistema como o número de telefone de um órgão público do interesse do sistema, por exemplo, gabinete do reitor da UFRGS, FAURGS, etc.

## **5.4 Implementação da Base de Dados e Geração de Código**

A partir da modelagem do banco de dados é possível criar o banco de dados. A figura 5.3 ilustra o mapeamento da modelagem sugerida na figura 5.1 para tabelas na base de dados. Como proposto no capítulo 4, o Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido para o sistema foi o PostgreSQL. A representação em formato de diagrama das tabelas na base de dados PostgreSQL é como segue:



**Figura 5.3: Diagrama Entidade-Relacionamento do banco de dados**

A comunicação entre o SGBD e o sistema, como visto no capítulo 4, é feita a pelo Yii a partir de objetos ActiveRecord (AR) que o framework oferece. Assim sendo, cada tabela ou entidade do banco de dados é mapeada para um objeto AR. Além desse mapeamento, também é feito a criação de CRUDs, no qual o framework cria, a partir do modelo AR, um Controller com ações respectivas do padrão CRUD (Criação, Recuperação, Atualização e Deleção) e também as Views necessárias para a renderização das páginas e interação das ações CRUD com o usuário final. Este processo pode ser observado no fluxograma da figura 5.4 no qual mostra os passos da modelagem da base de dados, criação de modelos

AR, criação de Controllers e páginas CRUD e adaptações desse código automatizado para o sistema proposto.

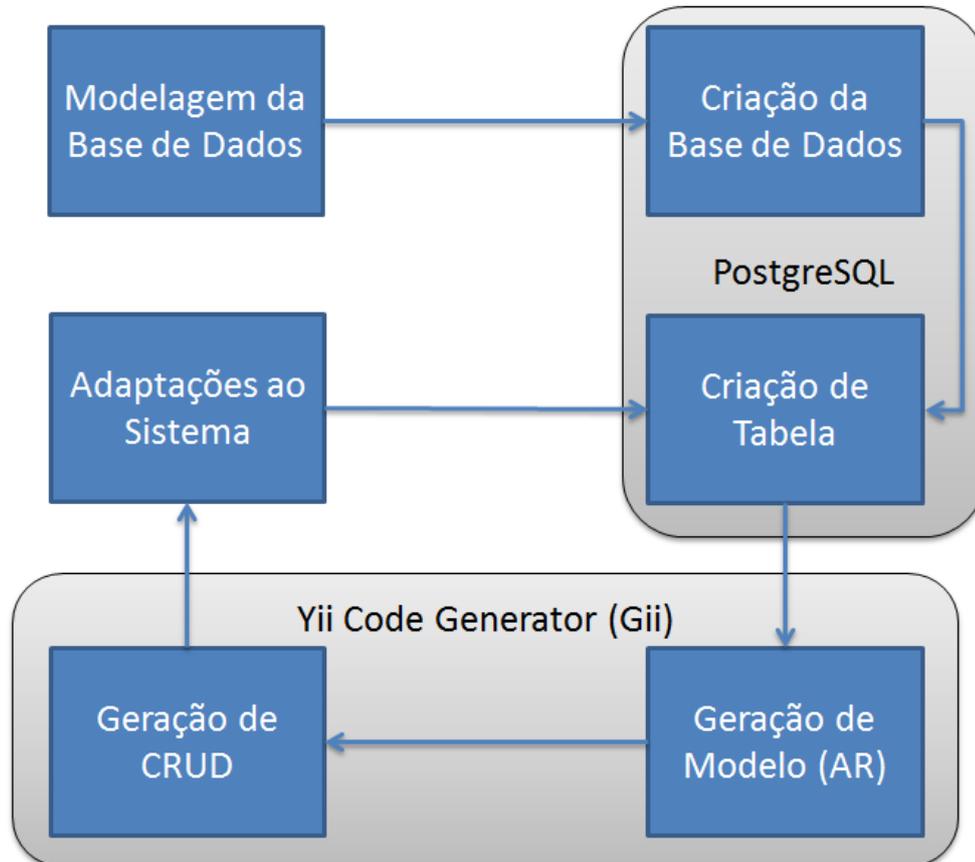


Figura 5.4: Fluxograma de criação de páginas a partir de tabelas da base de dados

## 5.5 Visões do Sistema

A partir da geração automática de código e posterior adaptação do código gerado para os moldes do sistema foi possível satisfazer as histórias propostas no início deste capítulo. A seguir serão apresentadas as principais visões do sistema desenvolvido em sua versão final. O sistema utiliza para seu design o framework de código aberto Twitter Bootstrap que oferece um grande conjunto de componentes reusáveis facilitando o desenvolvimento das visões do sistema.

Para o layout do sistema foi definida uma barra de menu fixa no topo da página. Nela podem ser encontrados o menu de navegação, as notificações, o menu de operações, a busca e o menu de configurações. Abaixo dessa barra de menu é encontrado um menu navegacional que indica ao usuário o caminho que percorreu para chegar na página em que se encontra. Por último é encontrado o conteúdo da página em si. Esse layout pode ser observado na figura 5.5 que mostra detalhadamente cada parte do layout.

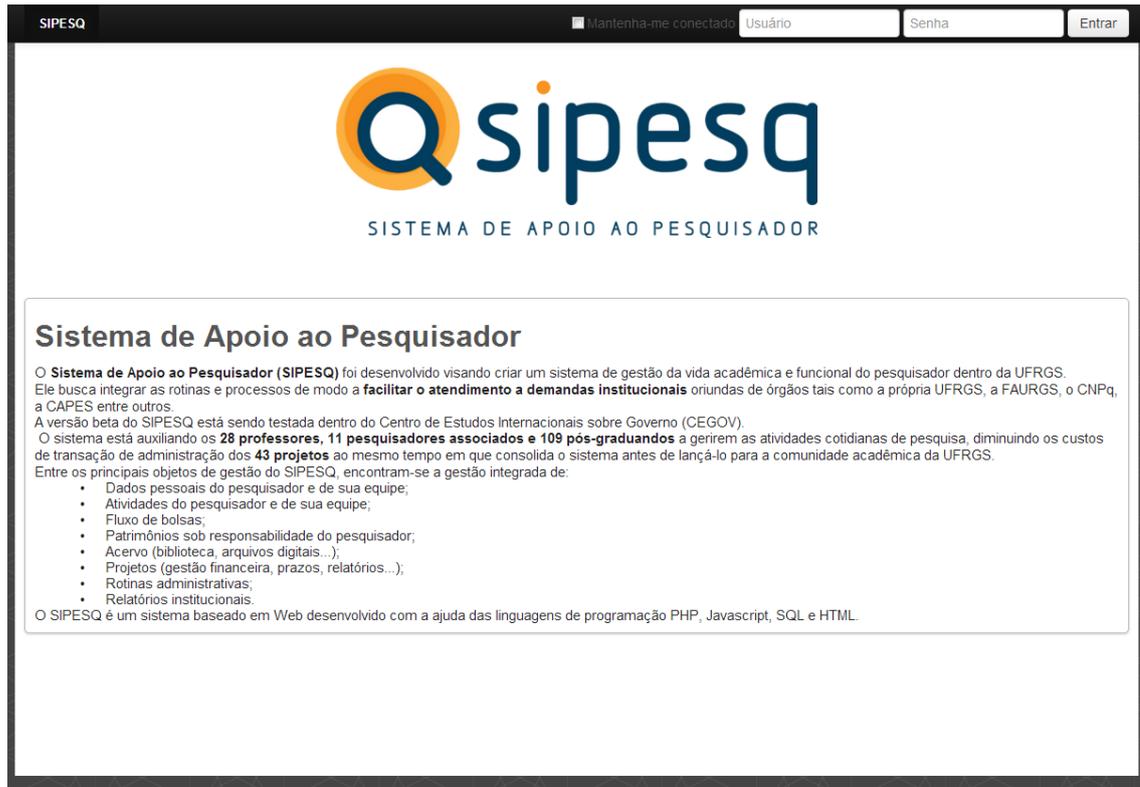


**Figura 5.5: Esquema de layout do sistema**

A seguir serão mostradas as figuras correspondentes às principais funcionalidades do sistema.

### 5.5.1 Tela Inicial do Sistema

A Figura 5.6 mostra a página inicial do sistema que aparece quando o usuário não está conectado ao mesmo. Nesta página o usuário deve digitar seu nome de usuário e sua senha. O sistema não permite nenhum tipo de interação sem o usuário estar previamente cadastrado.



**Figura 5.6: Tela de login do sistema**

Após a digitação do nome de usuário e sua senha o sistema então valida esses dados e redireciona o usuário, caso os dados estejam corretos, para a página inicial do sistema. Nessa página inicial o usuário encontrará os menus navegacionais, menus de opções e configurações e no conteúdo da página o usuário tem sua agenda de atividades do mês e a lista de atividades do mês como ilustrado na Figura 5.7.



**Figura 5.7: Tela inicial do sistema**

Na agenda o usuário encontra, em vermelho, atividades que tem prazo para a data específica e, em azul, projetos que terminam na data específica. A agenda também pode ser visualizada nos modos Ano, que mostra todas as ocorrências do ano corrente e o modo Semana que mostra as ocorrências da semana corrente.

### 5.5.2 Visões de Projeto

Nesta seção serão mostradas as visões referentes às páginas que o usuário dispõe para o gerenciamento de projetos.

As figuras 5.8 e 5.9 ilustram as visões oferecidas para a criação de um projeto. Na figura 5.8 temos o formulário de informações básicas de um projeto; Na figura 5.9 temos o formulário para seleção da equipe técnica do projeto. No formulário para seleção da equipe foi utilizado o componente de código aberto jQuery Tokeninput<sup>6</sup>.

<sup>6</sup> Disponível em: <http://loopj.com/jquery-tokeninput/>

## Criar Projeto

Informações

Instrumento Jurídico

Convênio

Equipe

Financeiro

Campos com \* são obrigatórios.

### Informações Gerais

Nome \*

Abreviatura \*

Finalizado

Situação

Tipo de Projeto \*

Pasta no Skydrive

### Prazos

Início

Término

Data do Relatório

### Descrição

Este projeto tem como objetivo a pesquisa de novas redes...

**Figura 5.8: Formulário de informações de um projeto**

## Criar Projeto

Informações Instrumento Jurídico Convênio Equipe Financeiro

Campos com \* são obrigatórios.

### Responsáveis e Participantes

Professor \*

Pós-Graduando \*

Graduando \*

Pessoas Atuantes

<b>Marcelo</b>	x
<b>Eduardo</b>	x
Marcelo	
<b>Demar</b>	
Marcelo M	
Marcelo N	
Marcelo S	
Marco A	
Marcus V	

**Figura 5.9: Formulário de seleção de equipe técnica de um projeto**

Além desses formulários há também um formulário de Instrumento Jurídico, que guarda informações como tipo de instrumento jurídico (Termo de Cooperação, Convênio, Contrato de Repasse, etc.) do projeto; um formulário de Convênio, que guarda informações como fundação de apoio, protocolos, vigência, código no portal do SICONS e também um formulário financeiro, que guarda informações sobre a verba orçamentada para o projeto.

Após a criação, os projetos podem ser encontrados na página de projetos do sistema como ilustrado na Figura 5.10.

Página inicial » Projetos

## Projetos

Selecione um projeto

Exibindo 1-5 de 6 resultados.

- Elaboração
- Negociação
- Tramitação
- Andamento: Início
- Andamento: Meio
- Andamento: Fim
- Prestação de Contas
- Encerrados

**Diretrizes para Requalificação**

Professor Responsável: Eber  
 Código do Projeto: 230  
 Término: 30/12/2014  
 Situação: **Elaboração**  
 Pasta no Skydrive:

**Mapeamento e Desafios dos Sistemas**

Professor Responsável: Vanessa  
 Código do Projeto:  
 Término: 01/10/2014  
 Situação: **Elaboração**  
 Pasta no Skydrive:

**Arquitetura da Informação**

Professor Responsável: Diego  
 Código do Projeto:  
 Término: 31/12/2013  
 Situação: **Elaboração**  
 Pasta no Skydrive:

**Residência em Sistemas de Informação**

Professor Responsável: Marcelo  
 Código do Projeto:  
 Término: 31/12/2013  
 Situação: **Elaboração**  
 Pasta no Skydrive:

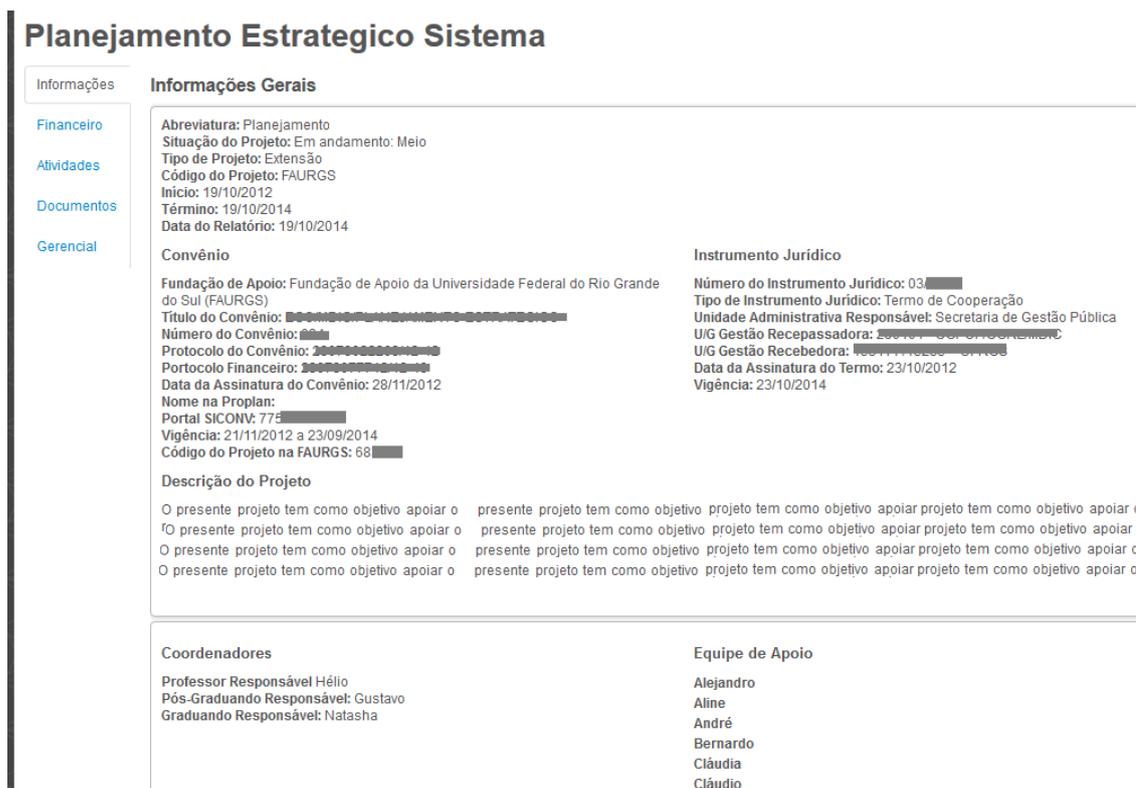
**VI Seminário Nacional**

Professor Responsável: Marília  
 Código do Projeto:  
 Término: 20/12/2014  
 Situação: **Elaboração**  
 Pasta no Skydrive:

**Figura 5.10: Página inicial de projetos**

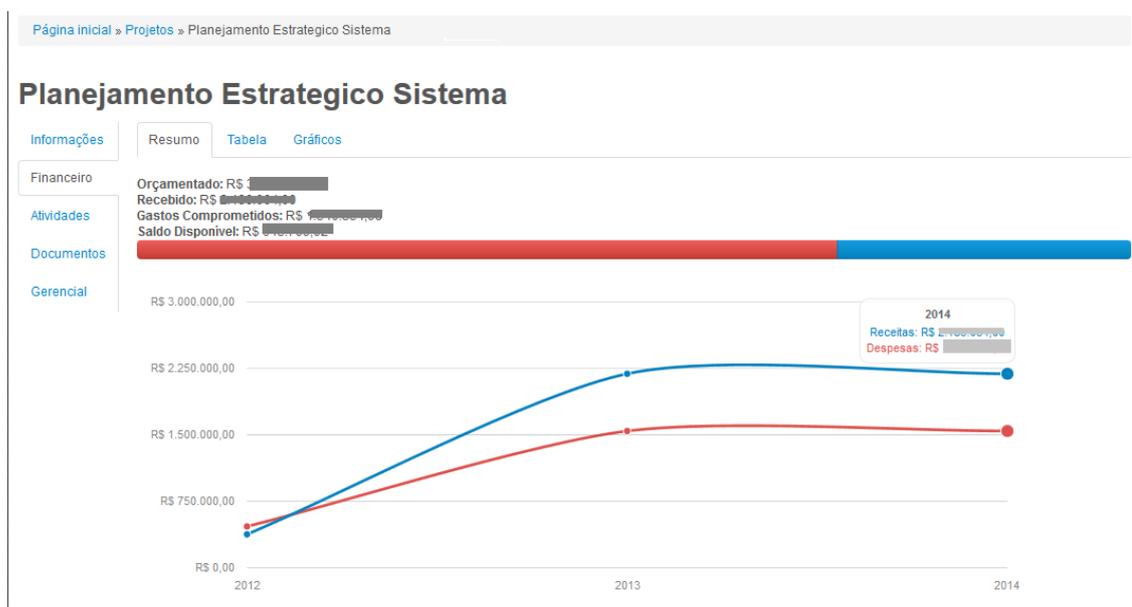
A Figura 5.10 mostra a página inicial de projetos já criados. Nesta página o usuário terá acesso a todos os projetos cadastrados no sistema. Os projetos são apresentados classificados pela sua fase, tornando mais fácil o controle do usuário sobre os projetos.

Ao clicar em um dos projetos o usuário vai para a página específica desse projeto. Nessa página o usuário dispõe de todas as informações do projeto. Para melhor navegação dessas informações o sistema subdivide as páginas em Informações, Financeiro, Atividades, Documentos e Gerencial.



**Figura 5.11: Página de informações de um projeto**

A Figura 5.11 mostra o conteúdo da página de informações de um projeto. Nessa página encontram-se informações gerais do projeto como prazos, situação do projeto, informações de convênio, informações sobre instrumento jurídico, equipe de apoio e coordenadores do projeto.



**Figura 5.12: Página de financeiro de um projeto**

A Figura 5.12 mostra a página inicial do financeiro de um projeto. Como o financeiro de um projeto pode ser extenso esta página está subdividida em três partes: Resumo, mostrada na Figura 5.12, é onde o usuário encontra um pequeno resumo financeiro do projeto durante sua execução; Tabela, nesta parte tem-se tabelas com orçamentos,

recebimentos e despesas do projeto; Gráficos, onde é disponibilizado gráficos gerais e individuais referente a cada rubrica utilizada nas despesas do projeto. Tais partes estão ilustradas nas figuras 5.13, 5.13 e 5.14, respectivamente.

## Planejamento Estrategico Sistema

Informações | Resumo | Tabela | Gráficos

Financieiro | Orçamentos | Recebimentos | **Despesas** | Patrimônios

Atividades

Documentos

Gerencial

Selecione uma Rubrica

Nome do Gasto	Rubrica	Data	Duração (meses)	Valor	+
Bolsa Hélio	Auxílio Financeiro a Pesquisadores	01/12/2012	6	R\$ 200,00	+Q
Bolsa José	Auxílio Financeiro a Pesquisadores	01/03/2013	6	R\$ 200,00	+Q
Bolsa Letícia	Auxílio Financeiro a Pesquisadores	01/12/2012	6	R\$ 200,00	+Q
Bolsa Letícia	Auxílio Financeiro a Pesquisadores	01/06/2013	6	R\$ 200,00	+Q
Bolsa Gustavo	Auxílio Financeiro a Pesquisadores	01/03/2013	6	R\$ 200,00	+Q
Bolsa Marcelo	Auxílio Financeiro a Pesquisadores	01/01/2013	6	R\$ 200,00	+Q

Figura 5.13: Visão de tabelas do financeiro de um projeto



### Balanco Financeiro do Projeto por Rubrica

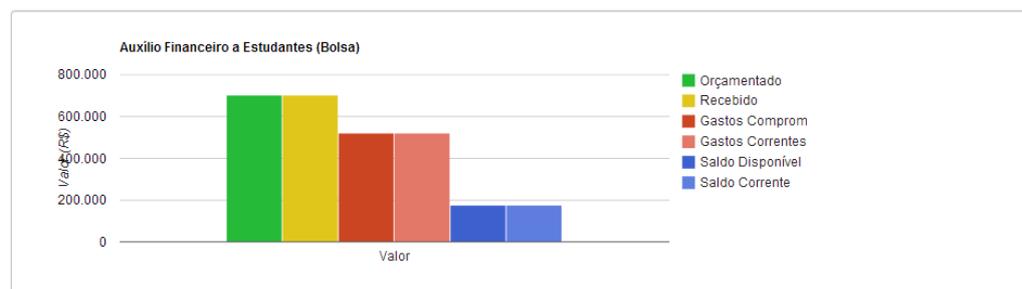


Figura 5.14: Visão de gráficos do financeiro de um projeto

Depois da parte do financeiro de um projeto tem-se a página de atividades. Nessa página o usuário encontra todas as atividades relacionadas ao projeto e também dispõe de menus para a criação de novas atividades. A Figura 5.15 ilustra a página de atividades de um projeto.

## Planejamento Estrategico

Informações

Financeiro

Atividades

Documentos

### Atividades

Adicionar Atividade

---

**Negociação Plano Trabalho** ( 01/10/2012 a 16/10/2012 )

Responsável: Marco

Categoria: Projetos de Pesquisa > Gestão do Projeto de Pesquisa

Participantes: Natasha , Aline

Projetos: Planejamento Estrategico Sistema

Estágio: A Fazer

---

**Tramitação Convênic** ( 18/10/2012 a 31/10/2012 )

Responsável: Aline

Categoria: Projetos de Pesquisa > Gestão do Projeto de Pesquisa

Participantes: Natasha , Gustavo

Projetos: Planejamento Estrategico Sistema

Estágio: A Fazer

---

**Elaboração de plano de trabalho detalhado do projeto "Planejamento Estratégico**  
01/11/2012 )

Responsável: Maria

Categoria: Projetos de Pesquisa > Elaboração de Projetos

Participantes: Natasha , Hélio

Projetos: Planejamento Estrategico Sistema

Estágio: A Fazer

---

**Elaboração do Relatório de "Proposta de Alocação de Analistas**

Responsável: Hélio

Categoria: Produção Técnica > Trabalho técnico

Participantes: Hélio

Projetos: Planejamento Estrategico Sistema

Estágio: A Fazer

**Figura 5.15: Página de atividades de um projeto**

Na página de documentos de um projeto o usuário dispõe encontra documentos relacionados ao projeto e também dispõe de menus para adição, edição e remoção de documentos. Essa página é ilustrada na Figura 5.16.

Informações

Financeiro

Atividades

Documentos

Gerencial

## Documentos

Nome	Data	Link	Ações
Termo de Cooperação	18/12/2012		
DOU - Termo de Cooperação	18/12/2012		
Convênio FAURGS + Termo de Compromisso	18/12/2012		
Plano de Trabalho	19/12/2012		
Termo de aceite Produto A	18/07/2013		
Termo de aceite Produto B	18/07/2013		
Termo de aceite Produto 1	18/07/2013		

[+ Incluir Documento](#)

**Figura 5.16: Página de documentos de um projeto**

Na página gerencial de um projeto o usuário dispõe de mecanismos para o gerenciamento de permissões do projeto. Nessa página o usuário é capaz de adicionar outros usuários com permissões específicas no projeto.

Informações

Financeiro

Atividades

Documentos

Gerencial

## Gerencial

### Permissões do projeto

Nome	Nível de Acesso	Detalhe
Hélio	Admin	Professor Responsável
Gustavo	Admin	Pós-graduando Responsável
Natasha	Admin	Graduando Responsável

[Editar Projeto](#)

### Permissões Individuais

[Adicionar Permissão](#)

Eduardo	Informações: Leitura Atividades: Leitura Financeiro: Acesso Negado Documentos: Leitura Gerencial: Acesso Negado Deleção: Acesso Negado <a href="#">Editar</a> <a href="#">Remover</a>
Marco	Informações: Leitura Atividades: Leitura Financeiro: Acesso Negado Documentos: Leitura Gerencial: Acesso Negado Deleção: Acesso Negado <a href="#">Editar</a> <a href="#">Remover</a>

**Figura 5.17: Página gerencial de um projeto**

### Adicionar Permissão

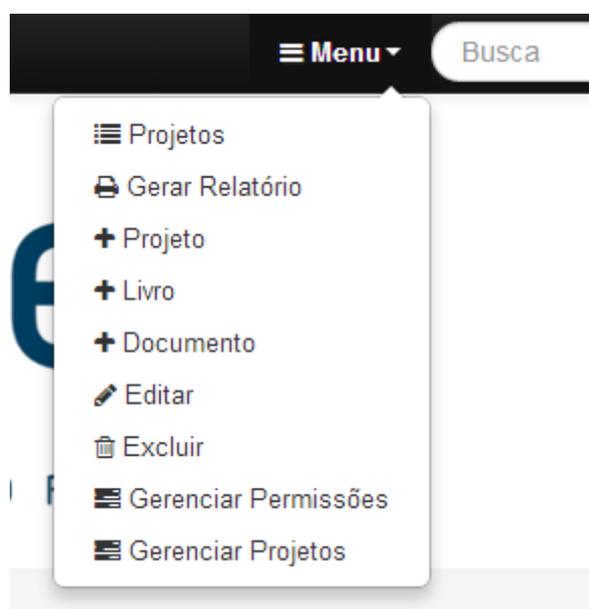
Local	Nível de Informação
Acesso a Informações	Leitura e Edição
Acesso ao Financeiro	Leitura
Acesso aos Documentos	Leitura, Edição e Exclusão
Acesso as Atividades	Leitura e Edição
Acesso ao Gerencial	Permitido
Deleção	Acesso Negado

Adicionar

**Figura 5.18: Formulário de adição de permissão a um projeto**

A Figura 5.17 mostra a página gerencial de um projeto enquanto que a Figura 5.18 ilustra o formulário que o usuário dispõe quando adiciona uma nova permissão no projeto. No formulário o usuário escolhe a pessoa a quem será dada as permissões e em seguida seleciona uma seção do projeto (informações, financeiro, atividades, documentos e gerencial) e o nível de permissão na seção.

Em todas as páginas do projeto é fornecido um menu de operações com as possíveis operações sobre o projeto. A Figura 5.19 ilustra este menu.



**Figura 5.19: Menu de operações de um projeto**

Nesse menu é encontrado as principais operações de um projeto, como a geração de relatório. Geração de relatórios de projetos é uma das principais funcionalidades do sistema proposto. A Figura 5.20 ilustra, parcialmente, um relatório gerado pelo sistema.

## 1 Informações Gerais

**Coordenador:** Hélio  
**Código do Projeto:** FAURGS: 8 88-8  
**Início:** 19/10/2012  
**Término:** 19/10/2014  
**Data do Relatório:** 19/10/2014

## 2 Equipe

Alejandro  
 Aline  
 André  
 Bernardo  
 Cláudia  
 Cláudio  
 Diego  
 Eduardo  
 Fernanda  
 Fernanda  
 Hélio  
 Isadora  
 Jose  
 Leticia  
 Marcelo  
 Márcia  
 Marco  
 Maria  
 Natasha  
 Ricardo  
 Rodolfo

## 3 Descrição

O presente projeto tem como objetivo apoiar o presente projeto tem como objetivo presente projeto tem como objetivo presente projeto tem como objetivo o MDIC e de presente projeto tem como objetivo presente projeto tem como objetivo presente projeto tem como objetivo ante os anos presente projeto tem como objetivo presente projeto tem como objetivo presente projeto tem como objetivo mentais de presente projeto tem como objetivo presente projeto tem como objetivo

## 4 Atividades

### Negociação Plano Trabalho e Termo Cooperação

Categoria	Projetos de Pesquisa > Gestão do Projeto de Pesquisa
Prazo	01/10/2012 a 16/10/2012
Responsável	Marco
Participantes	Natasha, Gustavo, Hélio

**Figura 5.20: Relatório de um projeto**

Neste relatório são exibidos todos os dados de um projeto como informações, atividades, tabelas referentes ao financeiro, gráficos, etc. Este relatório é de suma importância para os coordenadores de projetos no momento da finalização do projeto. Vale destacar que o relatório é dedicado para impressão em papel, assim sendo, não são exibidos menus ou links do sistema, apenas são exibidas informações contidas no relatório.

### 5.5.3 Visões de Pessoas

Na página pessoal de cada usuário é exibida informações relevantes a este usuário. Nessa página o usuário tem acesso as suas informações básicas, projetos, atividades e pagamentos. A Figura 5.21 ilustra a página de um usuário. Nesta página são oferecidos menus para o gerenciamento de atividades, pagamentos, dados pessoais e troca de senha.

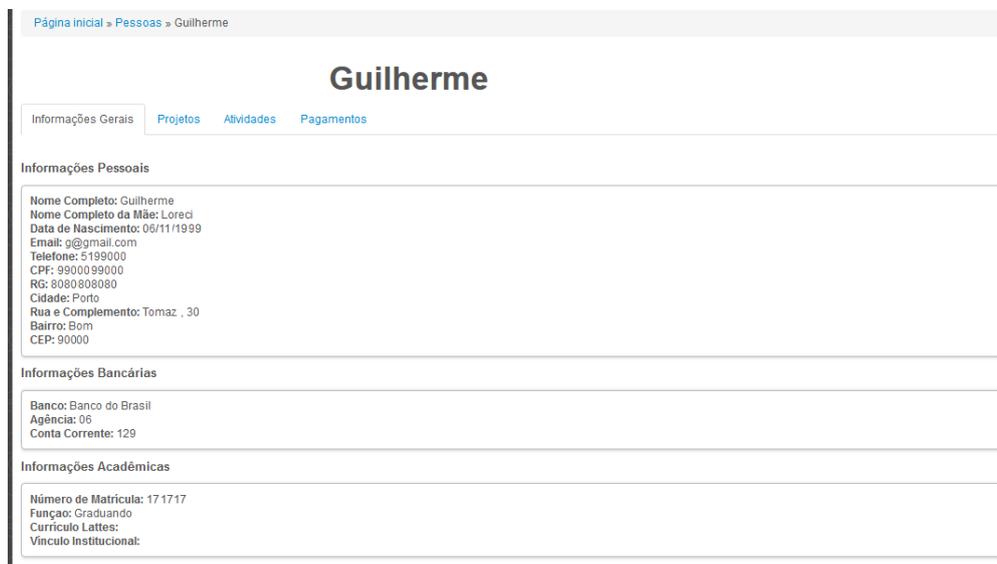


Figura 5.21: Página de um usuário

### 5.5.4 Visões dos menus do sistema

O sistema dispõe, na barra superior, de menus que possibilitam as operações básicas sobre as entidades do sistema. A seguir serão ilustrados os principais menus do sistema.

#### 5.5.4.1 Menu de Operações

O menu de operações aparece ao lado esquerdo da busca. Este menu é sensível ao contexto, ou seja, ele muda de acordo com a página que o usuário está. Se o usuário estiver em uma página de projeto, este menu mostrará opções referentes a projetos, caso o usuário esteja em uma página de atividades o menu mostrará opções referentes a atividades. Na Figura 5.19 foi ilustrado o menu de projetos, já na Figura 5.22 é ilustrado o menu de atividades.



Figura 5.22: Menu de operações sobre atividades

#### 5.5.4.2 Menu de Configurações

O menu de configurações apresenta opções globais referente ao sistema. Nele pode ser encontradas opções para geração de relatórios, gerenciamento de rubricas, gerenciamento de grupos e permissões, perfil do usuário, categorias de atividades e o menu para sair do sistema. A Figura 5.23 ilustra este menu.



**Figura 5.23: Menu de configurações do sistema**

#### 5.5.4.3 Notificações

Neste menu são encontradas notificações ao usuário. Por exemplo, quando um usuário cria ou atualiza uma atividade ou um projeto, todos os envolvidos são notificados. Deste jeito o sistema oferece um jeito rápido de manter os usuários informados sobre alterações no sistema. A Figura 5.24 ilustra notificações correntes a um usuário.



Figura 5.24: Notificações do sistema

#### 5.5.4.4 Busca do sistema

Neste menu é possível realizar buscas inteligentes no sistema. A busca é efetuada em pessoas, projetos e contatos de forma inteligente. Por exemplo, ao digitar o nome de pessoa o sistema retorna as pessoas cadastradas com este nome e os projetos que estas pessoas atuam tornando fácil a busca de entidades no sistema. A Figura 5.25 ilustra a busca do nome Guilherme.



Figura 5.25: Busca do sistema

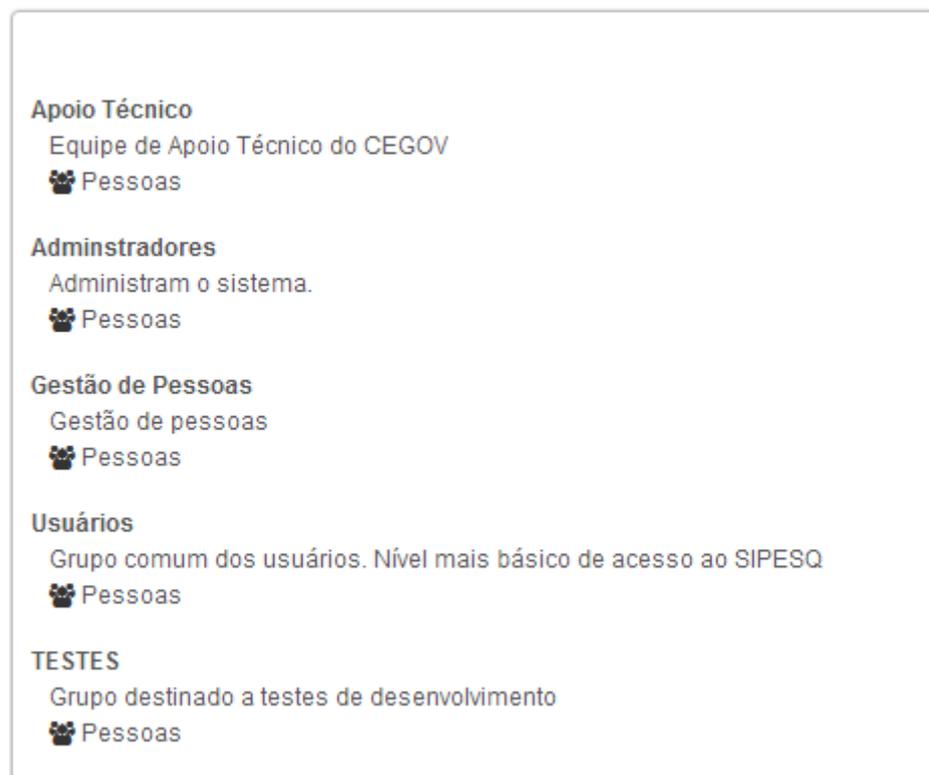
Nota-se que além da pessoa encontrada o sistema retornou projetos que esta pessoa participa.

### 5.5.5 Visões de Grupos

Grupos é uma das funções mais importantes do sistema. São os grupos que definem as permissões globais dos usuários no sistema. Ou seja, pode-se criar grupos que tenham acesso a todas as informações financeira de todos os projetos, mas que não possam acessar informações básicas de atividades ou pessoas, por exemplo. O sistema de grupos prevê flexibilidade para o administrador gerenciar as permissões dos usuários.

Um grupo pode ter um número variável de pessoas e uma pessoa pode pertencer a mais de um grupo, sendo que se dois grupos definem permissões para uma mesma seção do sistema, o usuário herdar a permissão de maior privilégio. Por exemplo, um usuário pertence aos grupos A e B. O grupo A define permissão de apenas leitura para as atividades, enquanto que o grupo B define permissão de leitura e escrita: o usuário terá permissão de leitura e escrita, visto que ele herda sempre o privilégio mais alto de sua intersecção de grupos.

## Grupos



**Figura 5.26: Página de grupos**

A Figura 5.26 ilustra a visão da página de grupos. Nessa página qualquer usuário pode ver todos os grupos do sistema e os usuários cadastrados, porém, somente usuários com privilégios podem criar novos grupos ou alterar um grupo existente. A Figura 5.27 demonstra o formulário de criação/edição de um grupo.

## Grupo: Administradores

Campos com \* são obrigatórios.

Nome do Grupo \*

Administradores

Pessoas do Grupo

**Equipe Atual**

- Aline
- Ana
- Antonio**
- Aragon
- Ariston
- Arthur
- Bruno
- Christiano
- Diego
- Diogo
- Eber
- Edson
- Eduardo
- Eduardo
- Elena
- Felipe
- Fernanda
- Fernando

Descrição do Grupo \*

Administram o sistema.

**Permissões**

- Página de Atividades
- Página de Pessoas
- Página de Projetos
- Página do Acervo
- Página de Gerência

**Permissões da Página de Pessoas**

Acesso as Informações Básicas \*

Permitido

Acesso as Informações Avançadas \*

Leitura, Edição e Exclusão

Acesso as Atividades \*

Leitura, Edição e Exclusão

Acesso ao Financeiro \*

Permitido

Deletar um Usuário \*

Permitido

**Figura 5.27: Formulário de um grupo**

Neste formulário o usuário define as permissões do grupo para as sessões e subseções do sistema.

### 5.5.6 Visões de Atividades

Na página de atividades, assim como na página inicial, o usuário dispõe de uma agenda, baseada no componente de código aberto Bootstrap-Calendar<sup>7</sup> que mostra suas atividades no mês, ano ou semana, dependendo da configuração da agenda. A diferença

<sup>7</sup> Disponível em: <https://github.com/Serhioromano/bootstrap-calendar>

desta página para a página inicial é a possibilidade de filtrar essa agenda por projeto, categoria de atividade ou pessoa. Além disso, o menu de operações oferece ações referente as atividades.

**Atividades**

Projeto  
Selecione um Projeto

Pessoa  
Selecione uma Pessoa

Categoria  
Selecione uma Categoria

Limpar

Dezembro de 2013

<< Anterior Hoje Próximo >> Ano Mês Semana

● Projeto ● Atividade ● Passo

Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

**Lista de Atividades**

- Apresentação de seminário em Paris 22/12/2013
- Avaliação das novas tecnologias pedagógicas 22/12/2013
- Orientação Doutorado Lucas 22/12/2013
- Elaboração do AVA 5/12/2013
- Orientação Doutorado Diego 5/12/2013
- Orientação Doutorado Nazaré 22/12/2013
- Orientação Doutorado Sandra 22/12/2013
- Elaboração de artigo teórico-metodológico 22/12/2013
- Gerência Executiva do projeto Planejamento Estratégico 22/12/2013
- Planejamento Estratégico 22/12/2013
- Rotinas Administrativas do GT Democracia 25/12/2013

**Figura 5.28: Visão da página de atividades**

No menu de operações da página de atividades temos a opção de criar uma nova atividade. A Figura 5.29 mostra o formulário para a criação de uma atividade.

# Adicionar Atividade

Campos com \* são obrigatórios.

**Nome da Atividade \***

**Categoria Primária**

**Categoria Secundária**

**Finalizado**

**Responsável \***

**Descrição \***

**B** *I* U ABC | ↶ ↷ | 🗑️ | ☰ ☷

**Data de Início \***

**Data de Fim \***

**Projetos**

- Edital - Linha A
- Edital - Linha C
- Extensão**
- Avaliação de Projetos Setoriais
- Ciclo de Capacitação
- Contrato UFRGS - Rede
- Desenvolvimento de Metodologia
- Metodologia de avaliação
- Planejamento Estrategico Sistema**
- Programa CEGOV de Capacitação
- Desenvolvimento Institucional**
- Atividades do Pesquisador Érico
- Atividades do Pesquisador Marco
- Desenvolvimento do SIPESQ

Segure a tecla CTRL para selecionar mais de um projeto.

**Pessoas**

×  ×

**Salvar Atividade**

**Figura 5.29: Formulário de criação de atividade**

Após a criação de uma atividade podemos acessar a página referente a esta atividade. Nesta página temos, além das informações cadastradas para a atividade, a opção de criar

passos para essa atividade. Os passos são criados arbitrariamente e tem a vantagem de não serem utilizados em relatórios. Sendo assim, os passos representam uma ótima ferramenta para gerenciar o andamento de uma atividade. Quando todos os passos são marcados como realizados, o sistema inteligentemente marca a atividade como finalizada. A Figura 5.30 ilustra a visão da página de uma atividade.

Programação do SIPESQ

Estágio: A Fazer

Categoria: Produção Técnica > Software

Responsável: Guilherme Rosa Severo

Data de Início: 16/03/2012

Data de Fim: 30/11/2012

Data de criação: 16/03/2012

Última Edição: 04/12/2013

Elaboração do software SIPESQ (Sistema de Apoio ao Pesquisador) por Guilherme Severo, voltado para a gestão da vida de um professor pesquisador e de seus projetos de pesquisa.

---

**Participantes**

Guilherme Rosa Severo

**Projetos**

CEGOV - Desenvolvimento do SIPESQ

---

**Adicionar Passo**

Prazo \*  x ↕ ▼

Pessoa \*

Descrição \*

**Passos em Aberto**

Área de orçamentos de projetos - Exportar para excel □

Guilherme Rosa Severo  
28/02/2013

Criação de Área de documentos - Templates de Arquivos, etc □

Guilherme Rosa Severo  
28/02/2013

Desenvolver sistema de autocompletar na busca do SIPESQ □

Gui Severo  
05/11/2013

**Passos Finalizados**

Adaptação do design - Integração com o Bootstrap do Twitter. ☑

Guilherme Rosa Severo  
30/10/2013

Desenvolvimento da Gerencia de Atividades (kanban) ☑

Guilherme Rosa Severo  
30/10/2013

PRIORIDADE DE 2012: Gerenciamento de permissões de acesso ☑

Guilherme Rosa Severo  
28/11/2012

Figura 5.30: Visão da página de uma atividade

### 5.5.7 Visão do Relatório Geral

O sistema fornece gráficos gerais representando todos os projetos, o financeiro de todos os projetos e as atividades cadastradas no sistema. Estes gráficos estão ilustrados na Figura 5.31.

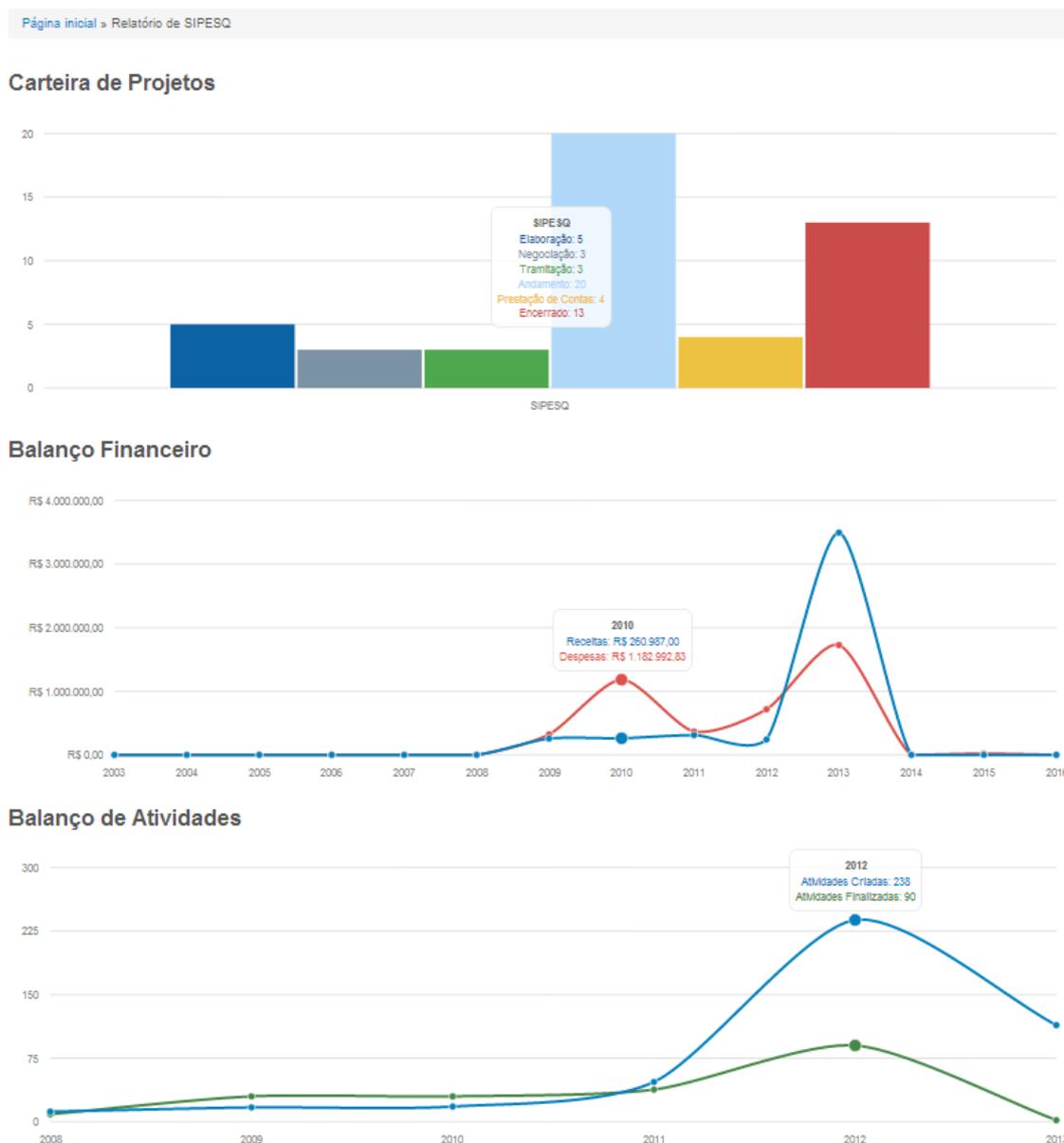


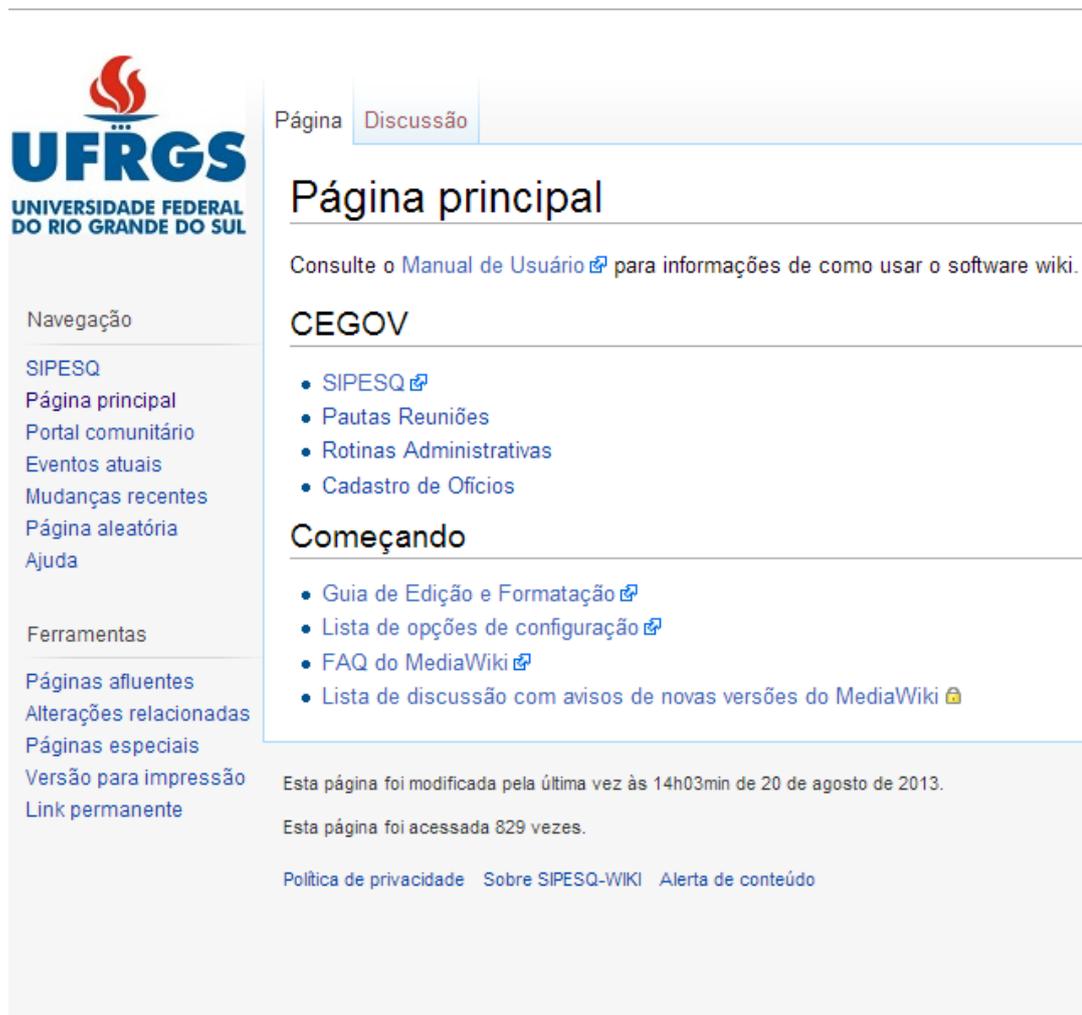
Figura 5.31: Relatório geral do sistema

### 5.5.8 Visão da Wiki

Como especificado no capítulo 2, o sistema proposto deveria ter o recurso de Wiki. Este recurso é importante para o cadastro de rotinas administrativas do sistema como pautas de reuniões, rotinas administrativas e cadastro de ofícios. No sistema proposto, para o sistema de Wiki foi utilizada a ferramenta de código aberto MediaWiki<sup>8</sup>. O MediaWiki oferece um sistema completo escrito em PHP com suporte a banco de dados PostgreSQL, tecnologias já presentes para o sistema.

A Figura 5.32 demonstra a página inicial do MediaWiki em conjunto com o sistema proposto.

<sup>8</sup> <http://www.mediawiki.org/wiki/MediaWiki>



**UFRGS**  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Navegação

- SIPESQ
- Página principal
- Portal comunitário
- Eventos atuais
- Mudanças recentes
- Página aleatória
- Ajuda

Ferramentas

- Páginas afluentes
- Alterações relacionadas
- Páginas especiais
- Versão para impressão
- Link permanente

Página **Discussão**

## Página principal

Consulte o [Manual de Usuário](#) para informações de como usar o software wiki.

### CEGOV

- [SIPESQ](#)
- [Pautas Reuniões](#)
- [Rotinas Administrativas](#)
- [Cadastro de Ofícios](#)

### Começando

- [Guia de Edição e Formatação](#)
- [Lista de opções de configuração](#)
- [FAQ do MediaWiki](#)
- [Lista de discussão com avisos de novas versões do MediaWiki](#)

Esta página foi modificada pela última vez às 14h03min de 20 de agosto de 2013.

Esta página foi acessada 829 vezes.

[Política de privacidade](#) [Sobre SIPESQ-WIKI](#) [Alerta de conteúdo](#)

**Figura 5.32: Página inicial da Wiki**

A Figura 5.33 ilustra a página de Rotinas Administrativas da Wiki. Nesta página são cadastradas rotinas acadêmicas comuns para diminuir a curva de aprendizagem nessas rotinas. Com essas rotinas previamente documentadas fica mais fácil para um usuário novo do sistema realizar essas rotinas.

**UFRGS**  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Navegação

- SIPESQ
- Página principal
- Portal comunitário
- Eventos atuais
- Mudanças recentes
- Página aleatória
- Ajuda

Ferramentas

- Páginas afluentes
- Alterações relacionadas
- Páginas especiais
- Versão para impressão
- Link permanente

Página [Discussão](#)

## Rotinas Administrativas

Esta página foi criada para orientar os membros do CEGOV no processo de administração. A elaboração de rotinas, lista de atividades e competências permite uma melhor interação dos bolsistas, permitindo, desta maneira, ampliarmos o tempo dedicado à atividade apontado por Barabási, pois cria um resumo de demandas que devem ser tratadas.

**Tabela de conteúdo** [\[esconder\]](#)

- 1 PESSOAL
  - 1.1 BOLSAS
    - 1.1.1 Bolsas Institucionais
      - 1.1.1.1 Solicitação
      - 1.1.1.2 Implementação
    - 1.1.2 Bolsas Voluntárias
  - 1.2 RELATÓRIOS
  - 1.3 ORIENTAÇÕES
    - 1.3.1 Agendamento de Apresentações
    - 1.3.2 Monitoramento das datas de defesa dos orientandos
    - 1.3.3 Encaminhamento de Ofícios e Solicitação de Atas e Atestados
    - 1.3.4 Reserva de Salas
    - 1.3.5 Atualização do Lattes e do SIPESQ
    - 1.3.6 Atualização das Pastas
  - 1.4 DESCRIÇÃO DAS FUNÇÕES
    - 1.4.1 DIRETOR DO CEGOV
    - 1.4.2 COORDENADOR EXECUTIVO
    - 1.4.3 ASSISTENTE EXECUTIVO
    - 1.4.4 COORDENADOR DE PROJETOS

Figura 5.33: Página de Rotinas Administrativas da Wiki

## 5.6 Parâmetros do Sistema

O sistema dispõe de parâmetros configuráveis, Grupos, Categorias de Atividades, Rubricas. Esses parâmetros são baseados em sistemas existentes como o Lattes (Categoria de Atividades) e Rubricas (Auditoria Interna da UFRGS). Todos os parâmetros utilizados são totalmente customizáveis. Isso garante ao sistema a readequação necessária caso haja alguma mudança nesses parâmetros, fornecendo assim, maior flexibilidade para o sistema.

## 5.7 Análise do Sistema

Nesta seção será discutido se o sistema proposto oferece as funcionalidades propostas na análise de requisitos e como essas funcionalidades são oferecidas no sistema. A tabela 5.3 mostra a relação das histórias propostas na análise de requisitos e se são cobertas ou não pelo sistema e de que forma são representadas.

Tabela 5.3: Análise da cobertura de requisitos

<b>História</b>	<b>Oferecida</b>	<b>Representação no sistema</b>
No papel de administrador quero ter acesso privilegiado a grupos do sistema para Gerenciar grupos e permissões dos usuários no sistema	SIM	Página de grupos no menu gerencial e permissão para cadastrar/editar grupos no sistema.
No papel de administrador quero ter acesso a rubricas do sistema para Gerenciar rubricas que serão usadas na área financeira dos projetos	SIM	Página de rubricas no menu gerencial e permissão para alterar rubricas.
No papel de administrador quero ter acesso às categorias de atividades para Gerenciar categorias de atividades que serão usadas no cadastro de atividades	SIM	Página de categorias de atividades no menu gerencial.
No papel de administrador quero ter acesso a relatório do sistema para saber dados numéricos relevantes de projetos, financeiro e atividades cadastradas no sistema.	SIM	Página de relatórios no menu gerencial do sistema e página de relatório de projeto no menu de operações dos projetos.
No papel de usuário quero cadastrar-me no sistema.	SIM	Formulário de cadastro de novos membros do sistema.
No papel de usuário quero entrar no sistema para usar as funcionalidades.	SIM	Formulário de login no topo da página.
No papel de usuário quero visualizar meu perfil para ter acesso as minhas informações	SIM	Página de perfil disponível no menu Gerencial do sistema.
No papel de usuário quero visualizar minhas notificações para saber as mudanças em minhas atividades e projetos	SIM	Menu Notificações na barra superior da página.
No papel de usuário quero buscar outros usuários.	SIM	Formulário de busca na barra superior da página.
No papel de usuário quero visualizar os projetos cadastrados no sistema.	SIM	Menu Projetos na barra superior da página.
No papel de usuário quero	SIM	Menu Atividades na barra

<b>História</b>	<b>Oferecida</b>	<b>Representação no sistema</b>
visualizar as atividades cadastradas no sistema.		superior do sistema.
No papel de usuário quero ter acesso à agenda para saber o horário dos outros usuários.	SIM	Página Agenda disponível através do menu Acervo.
No papel de usuário quero gerenciar a agenda de horários.	SIM	Página Agenda disponível através do menu acervo.
No papel de usuário quero gerenciar minhas férias para avisar outros usuários.	SIM	Página Agenda disponível através do menu Acervo.
No papel de usuário quero acessar acervo de livros para registrar possíveis empréstimos	SIM	Página Livros disponível através do menu Acervo.
No papel de usuário quero visualizar contatos cadastrados no sistema	SIM	Página Contatos disponível através do menu Acervo.
No papel de usuário quero gerenciar inscrições cadastradas no sistema.	SIM	Página subscriptions disponível através do menu Acervo.
No papel de usuário quero ter acesso ao MediaWiki para acessar informações relevantes como pautas de reuniões, rotinas administrativas e ofícios.	SIM	Página MediaWiki disponível através do menu Acervo.
No papel de usuário quero visualizar grupos cadastrados no sistema para verificar membros dos grupos e permissões que os grupos possuem.	SIM	Página Grupos disponível através do menu gerencial.
No papel de usuário quero buscar projetos, contatos e usuários para ter acesso rápido aos mesmos.	SIM	Formulário de busca na barra superior do sistema.
No papel de coordenador de projeto quero gerenciar projetos que coordeno.	SIM	Permissão para acessar a pagina dos projetos.
No papel de coordenador de projeto quero gerenciar o financeiro dos projetos que coordeno.	SIM	Permissão para acessar a página de financeiro dos projetos.
No papel de coordenador de projeto quero gerenciar permissões dos projetos	SIM	Permissão para página Gerencial dos projetos.

<b>História</b>	<b>Oferecida</b>	<b>Representação no sistema</b>
que coordeno. No papel de coordenador de projeto quero editar projetos que coordeno.	SIM	Permissão para editar os projetos através do menu de operações.
No papel de coordenador de projeto quero apagar para projetos que coordeno.	SIM	Permissão para deletar um projetos através do menu de operações.
No papel de coordenador de projeto quero gerenciar as atividades dos projetos que coordeno.	SIM	Permissão para ver/editar as atividades dos projetos.
No papel de coordenador de projeto quero gerenciar documentos dos projetos que coordeno.	SIM	Permissão de leitura e escrita na página de documentos dos projetos.
No papel de coordenador de projeto quero gerar relatórios dos projetos que coordeno.	SIM	Permissão para gerar relatório a partir do menu de operações dos projetos.
No papel de membro de projeto quero gerenciar atividades dos projetos que participo.	SIM	Permissão de cadastrar e gerenciar as atividades cadastradas nos projetos que atua.
No papel de membro de projeto quero gerenciar documentos dos projetos que participo.	SIM	Permissão de leitura na página de documentos dos projetos.
No papel de responsável de atividade quero gerenciar informações da atividade.	SIM	Permissão para editar as atividades que são responsáveis
No papel de responsável de atividade quero gerenciar participantes da atividade.	SIM	Permissão para editar a equipe da atividade.
No papel de participante de atividade quero acessar e editar informações da atividade.	SIM	Permissões para editar informações das atividades que participa.
No papel de participante de atividade quero cadastrar passos na atividade.	SIM	Formulário de cadastramento de passos nas atividades.
No papel de membro de grupo quero herdar permissões dos meus grupos.	SIM	Usuário herda a maior permissão disponível de seus grupos.

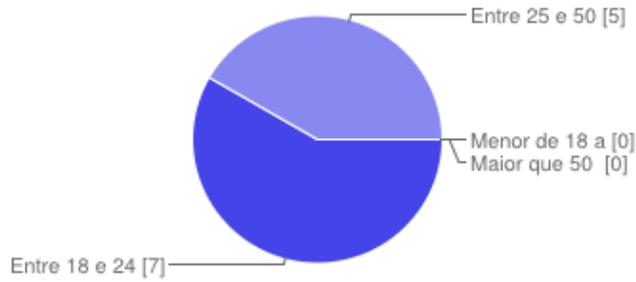
## **5.8 Usabilidade do Sistema**

Para avaliar a usabilidade do sistema foi feito um formulário de usabilidade. O formulário cobria questões como corretude do sistema, interface, facilidade de uso e também áreas que o usuário desejava que fossem aprimoradas. Os resultados do estudo são de fundamental importância para avaliar trabalhos futuros: quais são as maiores necessidades do sistema e onde ele deverá receber foco para atender melhor seus usuários.

A partir de resultados parciais, o estudo de usabilidade apontou seções do sistema que devem ganhar foco em trabalhos futuros. O estudo de usabilidade apontou a necessidade de melhorias na interface que o usuário dispõe para visualizar suas atividades. Também foi apontado o desejo do usuário de o sistema dispor de meios para exportação de dados formatados para outras plataformas, como exemplo a plataforma Lattes. Também foi visto que o sistema precisa de melhorias na área de administração dos projetos.

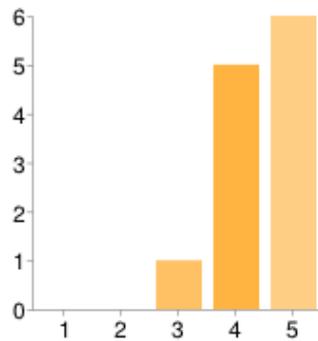
A seguir seguem imagens dos resultados desse estudo. O estudo contou com a participação de 12 pessoas, porém o formulário continuará aberto para mais usuários avaliar o sistema.

### Qual a sua idade?



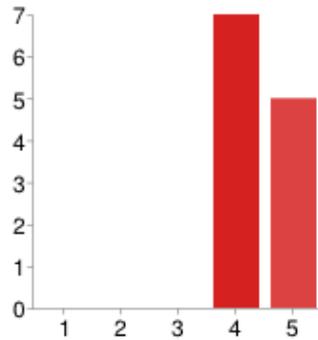
Menor de 18 anos.	0	0%
Entre 18 e 24 anos.	7	58%
Entre 25 e 50 anos.	5	42%
Maior que 50 anos.	0	0%

### Qual a sua experiência no uso de computadores e internet?



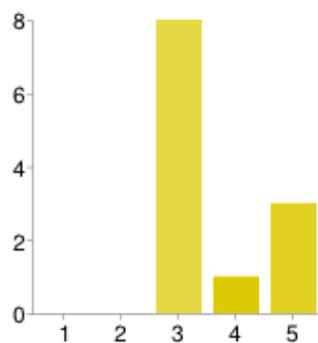
1	0	0%
2	0	0%
3	1	8%
4	5	42%
5	6	50%

### Em uma escala de 1 a 5, quão fácil foi entrar no sistema?



1	0	0%
2	0	0%
3	0	0%
4	7	58%
5	5	42%

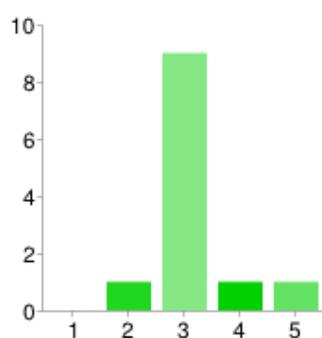
### Em uma escala de 1 a 5, quão fácil foi encontrar suas atividades?



1	0	0%
2	0	0%
3	8	67%
4	1	8%
5	3	25%

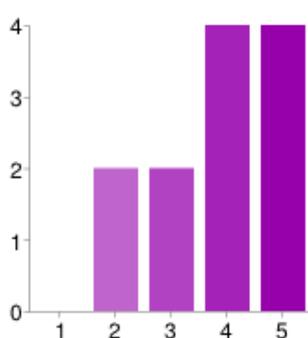
Figura 5.34: Resultados do estudo sobre usabilidade. Primeira parte

**Em uma escala de 1 a 5, quão fácil foi administrar suas atividades?**



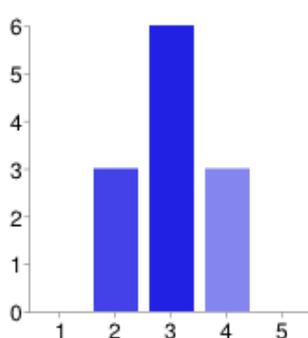
1	0	0%
2	1	8%
3	9	75%
4	1	8%
5	1	8%

**Em uma escala de 1 a 5, quão fácil foi encontrar seus projetos?**



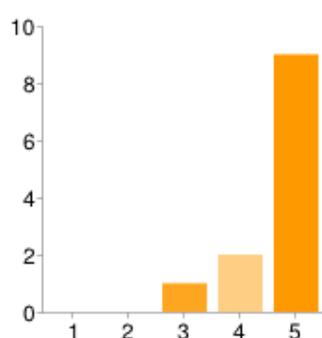
1	0	0%
2	2	17%
3	2	17%
4	4	33%
5	4	33%

**Em uma escala de 1 a 5, quão fácil foi administrar seus projetos?**



1	0	0%
2	3	25%
3	6	50%
4	3	25%
5	0	0%

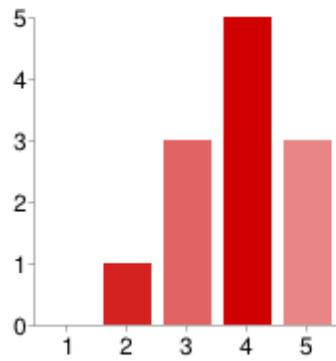
**Em uma escala de 1 a 5, quão fácil foi encontrar outros usuários?**



1	0	0%
2	0	0%
3	1	8%
4	2	17%
5	9	75%

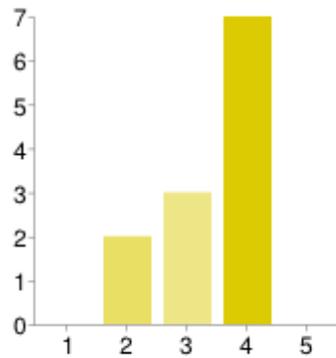
**Figura 5.35: Resultados do estudo sobre usabilidade. Segunda parte**

**Em uma escala de 1 a 5, quão satisfeito você está com a interface do sistema?**



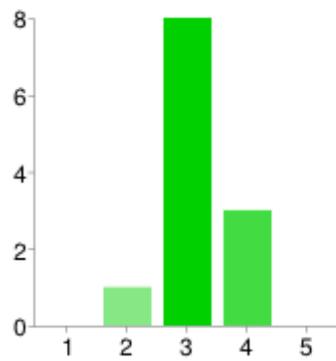
1	0	0%
2	1	8%
3	3	25%
4	5	42%
5	3	25%

**Em uma escala de 1 a 5, quão satisfeito você está com a corretude do sistema?**



1	0	0%
2	2	17%
3	3	25%
4	7	58%
5	0	0%

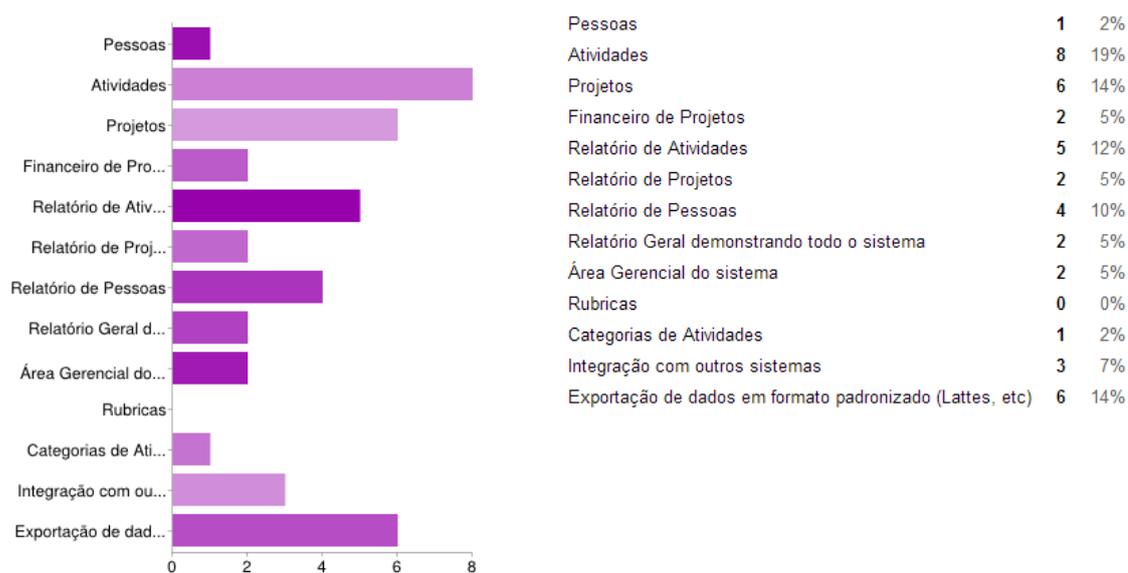
**Em sua opinião o quão completo o sistema é para você?**



1	0	0%
2	1	8%
3	8	67%
4	3	25%
5	0	0%

**Figura 5.36: Resultados do estudo sobre usabilidade. Terceira parte**

### Selecione áreas do sistema que você priorizaria mais desenvolvimento e aprimoramentos



### Em sua opinião o quão importante é o sistema para o gerenciamento de projetos de pesquisa?



### Possui alguma crítica, sugestão ou elogio?

Grande trabalho! É o "sistema nervoso" do CEGOV, como disse o Diretor. Parabéns.

**Figura 5.37: Resultados do estudo sobre usabilidade. Quarta parte**

## **6 CONCLUSÃO**

O presente trabalho apresentou o desenvolvimento de um sistema web para gerenciamento de rotinas administrativas cujo foco são os pesquisadores da UFRGS, se tornando, assim, em um sistema de apoio ao pesquisador.

O objetivo deste trabalho foi propor um sistema web que suprisse as necessidades dos pesquisadores em organizar seus projetos de pesquisa, visto o crescimento do investimento do governo federal na área de pesquisa e tecnologia e consequente crescimento no número de projetos e complexidade envolvida na gestão das rotinas administrativas desses projetos.

O resultado do trabalho se mostrou satisfatório no ponto de vista dos pesquisadores. Atualmente o sistema está sendo utilizado no CEGOV em uma versão beta. Essa versão gerencia cerca de 40 projetos, é utilizada diariamente por uma equipe de mais de 30 pesquisadores e tem se mostrado de fundamental importância no controle das rotinas administrativas do CEGOV.

### **6.1 Limitações do Sistema**

O sistema proposto possui um grande conjunto de funcionalidades, porém, ainda há muitas limitações. A principal limitação do sistema é a duplicação de dados. O sistema não possui integração com sistemas do governo como SICONV, Plataforma Lattes e sistemas da UFRGS como o Portal do Aluno. Dados pessoais, sistema de login dos pesquisadores deveria ser integrado ao sistema ao Portal do Aluno para evitar duplicação de dados, bem como, os projetos de pesquisa e suas atualizações deveriam ser integradas a sistemas como o SICONV.

### **6.2 Trabalhos Futuros**

Primeiramente é esperada a diminuição das limitações apresentadas pela interface no estudo sobre a análise de usabilidade do sistema. O estudo foi de suma importância para o apontamento de limitações do sistema.

Também é esperada a diminuição das limitações do sistema vistas anteriormente com a integração do sistema com outros sistemas de governo. Sendo assim, o sistema funcionaria como um portal de integração de sistemas de governo e da UFRGS com foco no pesquisador, onde o usuário utilizaria seus dados cadastrais já disponíveis no sistema do Portal do Aluno para entrar no sistema e atualizaria seus projetos com dados disponíveis no SICONV e outras plataformas. Também seria de interesse do usuário a geração relatórios de atividades com suporte a exportação para a plataforma Lattes. Para a integração dos sistemas, é importante salientar que o sistema desenvolvido utiliza o mesmo framework

PHP já utilizado pelos sistemas do CPD, o Yii. Isso facilitaria a integração dos sistemas em trabalhos futuros.

Apesar da análise de usabilidade feita ter indicado bons resultados e pontos cruciais a serem melhorados no sistema, ela foi inicial e não teve uma abordagem mais formal. Em trabalhos futuros é preciso de uma análise mais completa, multiplataforma, para a inclusão de dispositivos móveis no estudo, e pautada em critérios formais.

Quanto ao código do sistema, em trabalhos futuros é necessária a inclusão de mecanismos de testes. É necessária a inclusão de uma suíte de testes unitários, funcionais e de integração para garantir a corretude do sistema e prolongar sua vida útil.

Outro ponto interessante é a padronização da geração de relatórios bem como a ampliação dos relatórios gerados. Possibilitar a geração de diferentes tipos de relatórios com diferentes padrões para facilitar a integração com outros sistemas e a disponibilidade do produto de dados formatados no interesse do pesquisador.

## REFERÊNCIAS

- BECK, K. *et al.* **Manifesto for Agile Software Development**. Disponível em: <<http://agilemanifesto.org/>>.
- BIBEAULT, B.; KATS, Y. **jQuery in Action**. 2. ed. Stamford: Manning Publications Co., 2008. p. 452
- CAPES. **EVOLUÇÃO ORÇAMENTÁRIA – 2004 a 2011**. Disponível em: <[http://www.capes.gov.br/images/stories/download/sobre/OR%C3%87AMENTO\\_2004-2011\\_tabela.pdf](http://www.capes.gov.br/images/stories/download/sobre/OR%C3%87AMENTO_2004-2011_tabela.pdf)>. Acesso em: 6 nov. 2013.
- CNPQ. **Investimentos do CNPq em CT&I**. Disponível em: <[http://fomentonacional.cnpq.br/dmfomento/home/fmtvisualizador.jsp?UA=301&Facil=S&Visualizar=S&Corte1=017&Filtro1=Apoio Projeto de Pesquisa&Corte2=021&Filtro2=2012&Corte3=048](http://fomentonacional.cnpq.br/dmfomento/home/fmtvisualizador.jsp?UA=301&Facil=S&Visualizar=S&Corte1=017&Filtro1=Apoio%20Projeto%20de%20Pesquisa&Corte2=021&Filtro2=2012&Corte3=048)>. Acesso em: 2 nov. 2013.
- COAR, K.; BOWEN, R. **Apache Cookbook**. 2. ed. Sebastopol: O`REILLY, 2007. p. 285
- DUNCAN, W. A guide to the project management body of knowledge. **Management**, v. 1, n. 11, p. 459, 1996.
- FRAIN, B. **Responsive Web Design with Html5 and Css3**. [s.l.] Packt Publishing, 2012. p. 305
- HALPIN, T.; MORGAN, T. **Information modeling and relational databases**. [s.l.] Morgan Kaufmann, 2010. p. 475–525
- KRASNER, G. E.; POPE, S. T. A Cookbook for Using the Model- View-Controller User Interface Paradigm in Smalltalk-80. 1988.
- KUROSE, J.; ROSS, K. Computer networks: A top down approach featuring the internet. **Peorsoim Addison Wesley**, v. 1, p. 712, 2010.
- MAKAROV, A. **Yii 1.1 Application Development Cookbook**. Birmingham: Packt Publishing, 2011. p. 371
- OUSTERHOUT, J. Scripting: Higher level programming for the 21st century. **Computer**, n. March, p. 23–30, 1998.
- WINESETT, J. **Agile Web Application Development with Yii 1 . 1 and PHP5**. [s.l.] Packt Publishing, 2010. p. 368

ZHAO, H. *et al.* The HipHop compiler for PHP. **Proceedings of the ACM international conference on Object oriented programming systems languages and applications - OOPSLA '12**, p. 575, 2012.