

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO**

**SISTEMA DE INFORMAÇÕES PARA CONTROLE DO  
GERENCIAMENTO DE PROJETOS BASEADO NO PMBOK**

**GUILHERME FERNANDO GIELOW**

**BLUMENAU**  
**2008**

**2008/1-16**

**GUILHERME FERNANDO GIELOW**

**SISTEMA DE INFORMAÇÕES PARA CONTROLE DO  
GERENCIAMENTO DE PROJETOS BASEADO NO PMBOK**

Trabalho de Conclusão de Curso submetido à  
Universidade Regional de Blumenau para a  
obtenção dos créditos na disciplina Trabalho  
de Conclusão de Curso II do curso de Ciências  
da Computação — Bacharelado.

Prof. Evaristo Baptista - Orientador

**BLUMENAU  
2008**

**2008/1-16**

# **SISTEMA DE INFORMAÇÕES PARA CONTROLE DO GERENCIAMENTO DE PROJETOS BASEADO NO PMBOK**

Por

**GUILHERME FERNANDO GIELOW**

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Evaristo Baptista – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Paulo Roberto Dias – FURB

Membro: \_\_\_\_\_  
Prof. Everaldo Artur Grahl – FURB

Blumenau, 8 de julho de 2008

Dedico este trabalho especialmente à Simone e  
à minha família.

## **AGRADECIMENTOS**

A Deus, pelo seu imenso amor e graça.

À minha família que sempre esteve presente.

Aos meus amigos, pelos empurrões e cobranças.

À Simone que me apoiou neste semestre.

Ao professor Elton Minetto pela ajuda com o Cake PHP.

Ao meu orientador, Evaristo Baptista, por ter acreditado na conclusão deste trabalho.

Se você quer ser bem sucedido, precisa ter dedicação total, buscar seu último limite e dar o melhor de si mesmo.

Ayrton Senna

## RESUMO

Este trabalho apresenta um sistema de informações para controle do gerenciamento de projetos na web, fluxo controlado por workflow e baseado no PMBOK. O sistema desenvolvido tem como objetivo automatizar os processos e atividades inerentes ao controle e administração de projetos, buscando proporcionar ganhos de qualidade, produtividade e padronização em projetos. Utilizando a web como ambiente cliente usou-se o Cake PHP que é baseado nos padrões de projeto: MVC, *active record*, *association data mapping*, *front controller*.

Palavras-chave: PMBOK. *Workflow*. Cake PHP

## **ABSTRACT**

This work presents a system of information for control of project management in web, flow controlled by workflow and based on the PMBOK. The system developed is intended to automate the processes and activities related to the control and administration of projects, particularly related to the area of IT, seeking provide gains in quality, productivity and standardization in projects. Using the web as a customer environment used to the Cake PHP which is based on standards of design: MVC, active record, association data mapping, front controller.

Key-words: PMBOK. *Workflow*. Cake PHP.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Áreas de conhecimento e os processos de gerenciamento de projetos .....	20
Figura 2 - Fluxo de eventos de informações em uma arquitetura MVC .....	25
Figura 3 - Diagrama de caso de uso do administrador .....	33
Figura 4 - Diagrama do caso de uso do gerente e do analista do projeto .....	33
Figura 5 - Diagrama de caso de uso do patrocinador .....	34
Figura 6 - Diagrama de caso de uso do analista contábil .....	34
Figura 7 - Diagrama de estados de um projeto .....	45
Figura 8 - Diagrama de atividades do workflow .....	46
Figura 9 - MER com as tabelas principais do sistema.....	47
Figura 10 - Interface do phpMyAdmin .....	48
Figura 11 - Interface do bake do Cake PHP .....	49
Figura 12 - Funcionamento do MVC do Cake PHP .....	50
Figura 13 - Codificação dos campos obrigatórios na camada modelo .....	51
Figura 14 - Codificação que a classe pertence a classe estado .....	51
Figura 15 - Codificação que a classe possui nenhuma ou mais classes anexo.....	51
Figura 16 - Codificação que a classe acao pode ter e pertencer a muitas classes usuário.....	51
Figura 17 - Codificação da função que grava a auditoria dos usuários.....	52
Figura 18 - Codificação usando as funções query e execute do cake php.....	52
Figura 19 - Codificação da classe consistindo campos obrigatórios.....	53
Figura 20 - Codificação da função index de uma classe .....	53
Figura 21 - Codificação da função view de uma classe .....	53
Figura 22 - Codificação da função add de uma classe .....	53
Figura 23 - Codificação da função edit de uma classe .....	54
Figura 24 - Codificação da função delete de uma classe.....	54
Figura 25 - Codificação da função enviarcontabilidade da classe Estudosviabilidades.....	54
Figura 26 - Codificação da tela index de uma classe .....	55
Figura 27 - Codificação da tela view de uma classe.....	55
Figura 28 - Codificação da função add de uma classe .....	56
Figura 29 - Codificação da função edit de uma classe .....	56
Figura 30 - Tela de login do sistema .....	57
Figura 31 - Tela de mapa do gerente de projetos .....	57

Figura 32 - Tela de cadastro de projetos do sistema.....	58
Figura 33 - Tela de cadastro do estudo de viabilidade .....	59
Figura 34 – Tela para enviar estudo de viabilidade para contabilidade .....	60
Figura 35 – Tela para classificar contabilmente o estudo de viabilidade.....	60
Figura 36 – Tela para enviar estudo de viabilidade para patrocinador.....	61
Figura 37 - Tela do correio eletrônico recebido pelo patrocinador .....	61
Figura 38 – Tela para aprovar estudo de viabilidade.....	61
Figura 39 – Tela para atualizar elemento da EAP.....	63
Figura 40 - Tela de envio de processo de aquisição para o patrocinador.....	64
Figura 41 - Tela de aprovação do processo de aquisição .....	64
Figura 42 - Tela de cadastro de controle de mudança .....	66
Figura 43 - Tela para enviar controle de mudança para patrocinador .....	66
Figura 44 - Tela para aprovar o controle de mudança.....	67
Figura 45 - Tela para enviar termo de aceite de projeto para o patrocinador.....	67
Figura 46 - Tela para aprovar o projeto.....	68
Figura 47 - relatório de custos por departamento do projeto.....	68

## LISTA DE QUADROS

Quadro 1 - Detalhamento do caso de uso cadastrar projeto .....	37
Quadro 2 - Detalhamento do caso de uso cadastrar estudo de viabilidade.....	37
Quadro 3 - Detalhamento do caso de uso classificar contabilmente o estudo de viabilidade ..	38
Quadro 4 - Detalhamento do caso de uso aprovar o estudo de viabilidade.....	38
Quadro 5 - Detalhamento do caso de uso cadastrar atividade .....	39
Quadro 6 - Detalhamento do caso de uso cadastrar elemento EAP .....	39
Quadro 7 - Detalhamento do caso de uso atualizar elemento EAP .....	40
Quadro 8 - Detalhamento do caso de uso cadastrar plano de aquisição.....	40
Quadro 9 - Detalhamento do caso de uso cadastrar processo de aquisição.....	40
Quadro 10 - Detalhamento do caso de uso aprovar processo de aquisição.....	41
Quadro 11 - Detalhamento do caso de uso cadastrar orçamento.....	41
Quadro 12 - Detalhamento do caso de uso cadastrar relatório de desempenho .....	42
Quadro 13 - Detalhamento do caso de uso cadastrar controle de mudança .....	42
Quadro 14 - Detalhamento do caso de uso aprovar controle de mudança .....	43
Quadro 15 - Detalhamento do caso de uso cadastrar termo de aceite de projeto .....	43
Quadro 16 - Detalhamento do caso de uso aprovar projeto .....	44
Quadro 17 - Detalhamento do caso de uso encerrar projeto.....	44
Quadro 18 - Comparação entre os trabalhos correlatos e o sistema desenvolvido.....	69
Quadro 19 - Detalhamento da tabela projetos .....	76
Quadro 20 - Detalhamento da tabela licoesaprendidas .....	76
Quadro 21 - Detalhamento da tabela historicos.....	76
Quadro 22 - Detalhamento da tabela atasreunioes .....	77
Quadro 23 - Detalhamento da tabela estudosviabilidades.....	77
Quadro 24 - Detalhamento da tabela cronogramas .....	77
Quadro 25 - Detalhamento da tabela controlemudancas .....	78
Quadro 26 - Detalhamento da tabela termosaceitesprojetos .....	78
Quadro 27 - Detalhamento da tabela mensagens.....	79
Quadro 28 - Detalhamento da tabela passosworkflows.....	79
Quadro 29 - Detalhamento da tabela relatoriosdesempenhos .....	79
Quadro 30 - Detalhamento da tabela mapascomunicacoes .....	80
Quadro 31 - Detalhamento da tabela planosprojetos.....	80

Quadro 32 - Detalhamento da tabela planosquisicoes .....	80
Quadro 33 - Detalhamento da tabela planoscomunicacoes .....	81
Quadro 34 - Detalhamento da tabela planoscustos.....	81
Quadro 35 - Detalhamento da tabela orcamentos.....	81
Quadro 36 - Detalhamento da tabela declaracoesescopos.....	81
Quadro 37 - Detalhamento da tabela anexos .....	82
Quadro 38 - Detalhamento da tabela dicionarioseaps .....	82
Quadro 39 - Detalhamento da tabela atividades.....	82

## LISTA DE SIGLAS

API – *Application Programming Interface*

AJAX - *Asynchronous JavaScript and XML*

CRUD – *Create Retrieve Update Delete*

EAP – *Estrutura Analítica do Projeto*

ERP - *Enterprise Resource Planning*

HTML - *HyperText Markup Language*

MIT – *Massachusetts Institute Technology*

MVC – *Model View Controller*

PMBOK - *Project Management Body of Knowledge*

PMI - *Project Management Institute*

SESC - *Serviço Social do Comércio*

SICGP - *Sistema de Informações para Controle de Gerenciamento de Projetos*

XML - *eXtensible Markup Language*

WAMP – *Windows Apache MySQL PHP*

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>16</b>
1.1 OBJETIVOS DO TRABALHO .....	17
1.2 ESTRUTURA DO TRABALHO .....	17
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>18</b>
2.1 GERENCIAMENTO DE PROJETOS .....	18
2.1.1 Gerente de projetos .....	18
2.2 PMBOK.....	19
2.2.1 Áreas de conhecimento aplicadas ao trabalho .....	20
2.3 CAKE PHP.....	22
2.3.1 Padrões de projeto .....	23
2.3.1.1 MVC .....	24
2.3.1.2 Active record .....	25
2.3.1.3 Association data mapping.....	26
2.3.1.4 Front controller .....	26
2.3.2 AJAX .....	26
2.4 WORKFLOW.....	27
2.4.1 Uso de <i>workflow</i> nas organizações .....	27
2.5 SISTEMAS WEB.....	28
2.6 TRABALHOS CORRELATOS .....	28
2.6.1 Sistema de gerenciamento de projetos .....	29
2.6.2 Projeto dotproject.....	29
2.6.3 Ferramenta de gerenciamento de tempo de projetos.....	29
2.6.4 Aplicativo de gerência de projetos.....	30
<b>3 DESENVOLVIMENTO DO TRABALHO .....</b>	<b>31</b>
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	31
3.2 ESPECIFICAÇÃO .....	31
3.2.1 UML.....	32
3.2.1.1 Diagramas de casos de uso do sistema .....	32
3.2.1.2 Detalhamento dos casos de uso do sistema .....	34
3.2.1.3 Diagrama de estados .....	44
3.2.1.4 Diagrama de atividades do <i>workflow</i> .....	45

3.2.2 Modelo entidade-relacionamento.....	47
3.3 IMPLEMENTAÇÃO .....	48
3.3.1 Ferramentas utilizadas.....	48
3.3.1.1 Phpmysqladmin .....	48
3.3.1.2 Bake .....	49
3.3.2 Implementação do SICGP.....	49
3.3.2.1 Camadas MVC no sistema.....	49
3.3.2.1.1 Camada modelo.....	50
3.3.2.1.2 Camada controle .....	52
3.3.2.1.3 Camada visão .....	55
3.3.3 Operacionalidade da implementação .....	57
3.4 RESULTADOS E DISCUSSÃO .....	69
<b>4 CONCLUSÕES.....</b>	<b>71</b>
4.1 EXTENSÕES .....	72
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>73</b>
<b>APÊNDICE A – Dicionário de dados do SICGP .....</b>	<b>76</b>

## 1 INTRODUÇÃO

As empresas vêm reconhecendo a importância do gerenciamento de projetos para o sucesso de seus negócios. Utilizando as melhores práticas do gerenciamento de projeto as empresas terão projetos gerenciados de forma mais eficaz e produzirão melhores resultados.

Segundo o Project Management Institute (2004, p. 8), “o gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos”. O objetivo do gerenciamento de projetos é assegurar que processos particulares sejam seguidos, coordenando e monitorando as atividades da engenharia do produto. Um processo de gerenciamento de projeto deve identificar, estabelecer, coordenar e monitorar as atividades, as tarefas e os recursos necessários para um projeto produzir um produto e/ou serviço de acordo com seus requisitos. (MARTINS, 2002, p. 4)

Segundo o Project Management Institute (2004, p. 3) o PMBOK é um conjunto de conhecimentos em gerenciamento de projetos que é a soma dos conhecimentos básicos da profissão de gerenciamento de projetos. O PMBOK completo inclui práticas tradicionais comprovadas amplamente aplicadas, além de práticas inovadoras que estão surgindo na profissão. Como resultado disso, o conjunto de conhecimentos em gerenciamento de projetos está em constante evolução. O PMBOK é dividido em 9 áreas do conhecimento que seguem: gerência de integração, gerência de escopo, gerência de tempo, gerência de custos, gerência de comunicação, gerência de aquisições, gerência de recursos humanos, gerência de qualidade e gerência de riscos.

A atividade de gerenciamento de projetos exige uma comunicação efetiva para todos os envolvidos no projeto, de acordo com a responsabilidade de cada membro. Desta forma entra a comunicação por *workflow*, que distribui as tarefas e comunica o andamento do projeto para os interessados.

Segundo Moro (2007), *workflow* é definido como uma coleção de tarefas organizadas para realizar um processo, quase sempre de negócio. Essas tarefas podem ser executadas por sistemas de computador, agentes humanos, ou então por uma combinação destes. A ordem de execução e as condições pelas quais cada tarefa é iniciada também estão definidas no *workflow*, sendo que o mesmo é capaz ainda de representar a sincronização das tarefas e o fluxo de informações.

Aliando as funcionalidades do *workflow* com as melhores técnicas e práticas

recomendadas pelo PMI através do guia PMBOK, é possível construir as soluções de sistemas que tornam o trabalho de gerenciamento rápido e contínuo.

Este trabalho objetiva automatizar os processos e atividades inerentes ao controle e administração de projetos, buscando proporcionar ganhos de qualidade e produtividade em projetos.

Explorando os recursos da web associados à integração dos fluxos de trabalhos e utilizando as melhores práticas do PMBOK, o sistema desenvolvido objetiva desburocratizar o controle do gerenciamento de projetos, aumentar a velocidade do fluxo de trabalho e disponibilizar acesso ao sistema de qualquer lugar com internet.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi desenvolver um sistema de informações para controle do gerenciamento de projetos na web, fluxo controlado por workflow e baseado no PMBOK. Os objetivos específicos do trabalho são:

- a) organizar as informações do projeto baseado nas melhores práticas do PMBOK;
- b) disponibilizar o sistema para acesso de qualquer lugar com internet para os responsáveis pelo projeto;
- c) quantificar o custo e o tempo gasto em cada área envolvida de cada projeto;
- d) controlar o fluxo de gerenciamento através das funcionalidades do *workflow*;

## 1.2 ESTRUTURA DO TRABALHO

O capítulo 2 apresenta alguns conceitos, tecnologias e ferramentas utilizadas no trabalho. O capítulo 3 apresenta a especificação e desenvolvimento do trabalho, onde os resultados obtidos no desenvolvimento são avaliados. Por último, o capítulo 4 apresenta as conclusões para o trabalho, bem como propostas de extensões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os aspectos teóricos sobre o trabalho tais como: gerenciamento de projetos, PMBOK, Cake PHP, *workflow* e sistemas web. A última seção apresenta os trabalhos correlatos que possuem características comuns ao sistema proposto.

### 2.1 GERENCIAMENTO DE PROJETOS

O gerenciamento de projetos, segundo Possi (2005, p. 8), é a realização de um projeto através de três elementos básicos: habilidades técnicas, habilidades interpessoais e habilidades administrativas. Desta forma a garantia de sucesso do projeto pressupõe o domínio básico destas habilidades por parte do gerente de projetos.

Segundo Project Management Institute (2004, p. 8), o gerenciamento de projetos é realizado através da aplicação e da integração dos seguintes grupos de processos ou fases de gerenciamento de projetos: iniciação, planejamento, execução, monitoramento e controle e encerramento. Cada grupo ou fase engloba tarefas específicas com o objetivo de atender as necessidades do projeto, incluindo entre as principais responsabilidades do gerente do projeto as atividades que seguem:

- a) identificar as necessidades;
- b) estabelecer objetivos claros e alcançáveis;
- c) balancear as demandas conflitantes de qualidade, escopo, tempo e custo;
- d) adaptar as especificações dos planos e da abordagem às diferentes preocupações e expectativas dos clientes.

#### 2.1.1 Gerente de projetos

O gerente de projetos é o principal membro para um gerenciamento de projetos eficaz, pois é ele que gerencia pessoas e a progressão do projeto como um todo.

Conforme Mendes (2006, p. 53), o gerente de projetos possui três paradoxos:

- a) o projeto deve atender as expectativas de todos os envolvidos para que ele seja

considerado bem sucedido;

- b) espera-se que o projeto atinja o objetivo, mesmo que não possua todos os recursos disponíveis;
- c) o gerente não pode prever o futuro, portanto sempre há variações de custo e tempo, mas estas nunca podem surpreender os clientes do projeto.

Um bom gerente de projetos deve possuir motivação, energia e iniciativa para atingir os objetivos, deve também ser portador de boa comunicação, credibilidade administrativa, liderança, credibilidade técnica e saber gerenciar conflitos, lidar com estresse, negociar e resolver problemas e apresentar soluções. Para Boente (2003, p. 32), o gerente de projetos é o grande chefe da equipe de projetos e todas as ações do projeto são conseqüências de suas decisões.

## 2.2 PMBOK

O PMBOK tem como objetivo identificar o conjunto de conhecimentos em gerenciamento de projetos que é amplamente reconhecido, mas não suficiente de forma isolada para garantir um gerenciamento de projeto eficaz.

Segundo Project Management Institute (2004, p. 12), um gerenciamento de projetos eficaz exige que a equipe de gerenciamento de projetos entenda e use o conhecimento e as habilidades de pelo menos cinco áreas de especialização:

- a) conjunto de conhecimentos em gerenciamento de projetos;
- b) conhecimento, normas e regulamentos da área de aplicação;
- c) entendimento do ambiente do projeto;
- d) conhecimento e habilidades de gerenciamento geral;
- e) habilidades interpessoais.

De forma particular o conhecimento de gerenciamento de projetos consiste no estudo e aplicação das etapas que seguem:

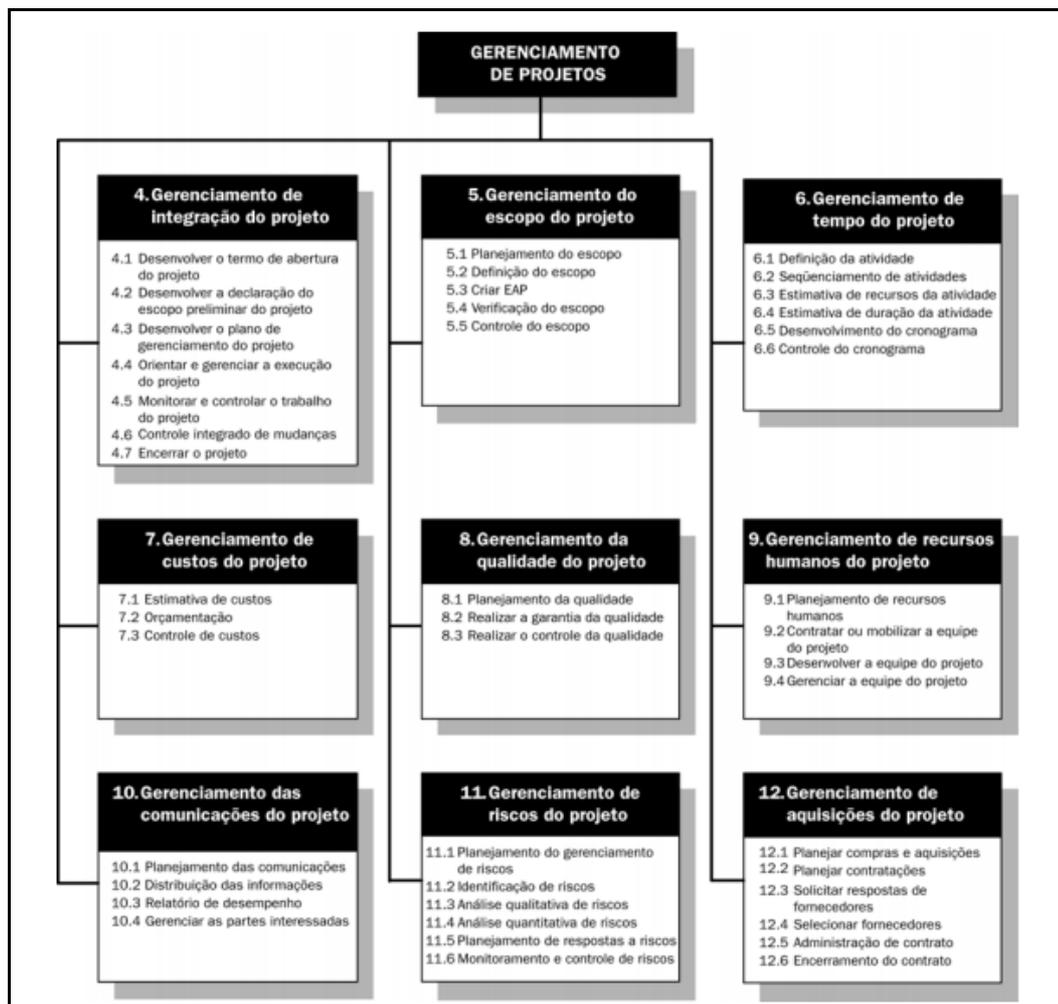
- a) definição do ciclo de vida do projeto;
- b) agrupamento dos processos de gerenciamento de projetos em cinco fases, a saber: iniciação, planejamento, execução, controle e monitoramento e encerramento;
- c) orientação através das nove áreas de conhecimento em gerência de projetos, ou seja: integração, escopo, tempo, custos, qualidade, recursos humanos,

comunicações, riscos e aquisições.

Para efeitos didáticos e melhor estruturação das informações, o guia PMBOK está organizado em três seções, a saber:

- estrutura do gerenciamento de projetos;
- norma de gerenciamento de projetos de um projeto;
- áreas de conhecimento em gerenciamento de projetos.

O PMBOK é dividido em nove áreas do conhecimento e quarenta e quatro processos de gerenciamento de projetos como demonstrado na Figura 1.



Fonte: Project Management Institute (2004).

Figura 1 - Áreas de conhecimento e os processos de gerenciamento de projetos

### 2.2.1 Áreas de conhecimento aplicadas ao trabalho

Foram escolhidas seis das nove áreas de conhecimento do PMBOK para aplicação neste trabalho que são: gerência de integração, gerência de escopo, gerência de tempo,

gerência de custos, gerência de comunicações e gerência de aquisições.

Segundo Vargas (2007), a gerência de integração tem o objetivo de garantir que todas as demais áreas estejam integradas.

A integração é dividida em sete processos:

- a) desenvolver o termo de abertura do projeto;
- b) desenvolver a declaração do escopo preliminar do projeto;
- c) desenvolver o plano de gerenciamento do projeto;
- d) orientar e gerenciar a execução do projeto;
- e) monitorar e controlar o trabalho de projeto;
- f) controlar integração de mudanças;
- g) encerrar o projeto;

A gerência de escopo tem o objetivo de definir e controlar os trabalhos executados pelo projeto para garantir que todas as premissas serão cumpridas, através da menor quantidade de trabalho possível.

O escopo é dividido em cinco processos:

- a) planejar o escopo;
- b) definir o escopo;
- c) criar EAP;
- d) verificar o escopo;
- e) controlar o escopo.

A gerência de tempo tem como objetivo inicial controlar prazos e confeccionar cronogramas.

O tempo é dividido em seis processos:

- a) definir as atividades;
- b) seqüenciar as atividades;
- c) estimar os recursos da atividade;
- d) estimar a duração de atividade;
- e) desenvolver o cronograma;
- f) controlar o cronograma.

A gerência de custos tem como objetivo realizar os trabalhos do projeto e possuir os recursos suficientes.

O custo é dividido em três processos:

- a) estimar os custos;
- b) orçamentar;

c) controlar os custos.

A gerência de comunicações tem como objetivo fazer com que as informações cheguem às pessoas envolvidas e no tempo certo da maneira mais econômica possível.

A comunicação é dividida em quatro processos:

- a) planejar as comunicações;
- b) distribuir as informações;
- c) criar relatório de informações;
- d) gerenciar as pessoas interessadas.

A gerência de aquisições tem como objetivo garantir que todos os elementos externos do projeto sejam fornecidos de forma eficaz.

A aquisição é dividida em seis processos:

- a) planejar compras e aquisições;
- b) planejar contratações;
- c) solicitar respostas aos fornecedores;
- d) selecionar fornecedores;
- e) administrar os contratos;
- f) encerrar o contrato.

### 2.3 CAKE PHP

Conforme Nascimento (2007, p. 241), o Cake PHP teve seu desenvolvimento iniciado em 2005 por Michal Tartarynowicz e é indicado para aqueles que desejam desenvolver aplicativos rápidos e consistentes.

Segundo o Cake PHP Brasil (2007), o Cake PHP é um *framework* escrito em PHP que tem como principais objetivos oferecer uma estrutura que possibilite aos programadores de PHP de todos os níveis desenvolverem aplicações robustas rapidamente, sem perder a flexibilidade. Desta forma o desenvolvedor irá focar seu trabalho nas regras de negócio do sistema.

O Cake PHP é baseado no *framework Ruby on Rails*, que é um *framework* de código aberto para desenvolvimento de aplicações web escrito em *Ruby*<sup>1</sup>, e utiliza padrões de

---

<sup>1</sup> Ruby é uma linguagem de programação interpretada, com tipagem dinâmica e forte, orientada a objetos.

desenvolvimento como:

- a) *active record*, que permite relacionar classes de modelo com o banco de dados;
- b) *association data mapping*, que permite representar as tabelas do banco de dados através de classes e os registros de cada tabela são representados como instâncias das classes correspondentes;
- c) *front controller*, que fornece um controlador centralizado para manter a lógica de processamento que ocorre na camada de apresentação;
- d) MVC, que permite separar a lógica da aplicação, da interface do usuário e do fluxo da aplicação.

Segundo Minetto (2007, p. 23), o Cake PHP possui as seguintes vantagens:

- a) licença flexível do MIT;
- b) compatibilidade com PHP4 e PHP5;
- c) CRUD para interação com o banco de dados;
- d) arquitetura MVC;
- e) validações de estrutura de dados;
- f) *templates*;
- g) ferramentas que auxiliam gerar Javascript, AJAX e formulários em HTML;
- h) comunidade ativa;
- i) funcionalidade em qualquer subdiretório web, com pouca configuração do servidor de aplicação Apache.

Segundo Maseto (2006, p. 85), para que as facilidades da geração do CRUD do Cake PHP funcionem corretamente recomenda-se seguir regras como:

- a) todas as tabelas devem estar no plural;
- b) a chave primária para todas deve ser o campo *id*;
- c) para utilizar chave estrangeira, deve possuir nome da tabela no singular acrescido do campo *id*. Como exemplo, escreve-se `usuario_id`;
- d) o relacionamento “n – n” deve ser organizado em ordem alfabética, e o primeiro nome da tabela no singular acrescido de um *underline* mais o nome da segunda tabela no plural. Como exemplo, escreve-se `usuario_livros`.

### 2.3.1 Padrões de projeto

Conforme Gamma (2000, p. 20), os padrões de projetos são descrições de objetos que

se comunicam e classes que são customizadas para resolver um problema genérico de *design* em um contexto específico, ou seja, uma solução padrão para problemas comuns.

Para Melo (2007), os benefícios de utilizar padrões de projetos são:

- a) melhorar suas habilidades em programação;
- b) desenvolver software sem duplicação de código;
- c) facilitar a documentação e aprendizagem sobre orientação a objetos.

No total existem 23 padrões clássicos que são divididos conforme o tipo de tarefa para a qual se destinam. São as tarefas:

- a) criação (*factory method, abstract factory, builder, prototype, singleton*);
- b) estrutura (*class adapter, object adapter, bridge, composite, decorator, facade, flyweight, proxy*);
- c) comportamento (*interpreter, template method, chain of responsibility, command, iterator, mediator, memento, observer, state, strategy, visitor*).

Na seqüência serão explicados padrões de projetos utilizados no Cake PHP que são:

- a) MVC;
- b) *active record*;
- c) *association data mapping*;
- d) *front controller*.

### 2.3.1.1 MVC

A arquitetura MVC divide as aplicações em 3 camadas sendo elas modelo, visão e controle e tem por objetivo separar regras de negócio e apresentação.

As principais vantagens de usar arquitetura MVC são:

- a) reaproveitamento de código;
- b) facilidade de manutenção;
- c) integração de equipes e/ou divisão de tarefas;
- d) camada de persistência independente;
- e) implementação de segurança;
- f) facilidade na alteração da interface da aplicação;
- g) escalabilidade da aplicação.

Segundo Neto (2007), as características da camada modelo são:

- a) possuir regras de comportamentos;

- b) armazenar, manipular e gerar dados;
- c) encapsular dados.

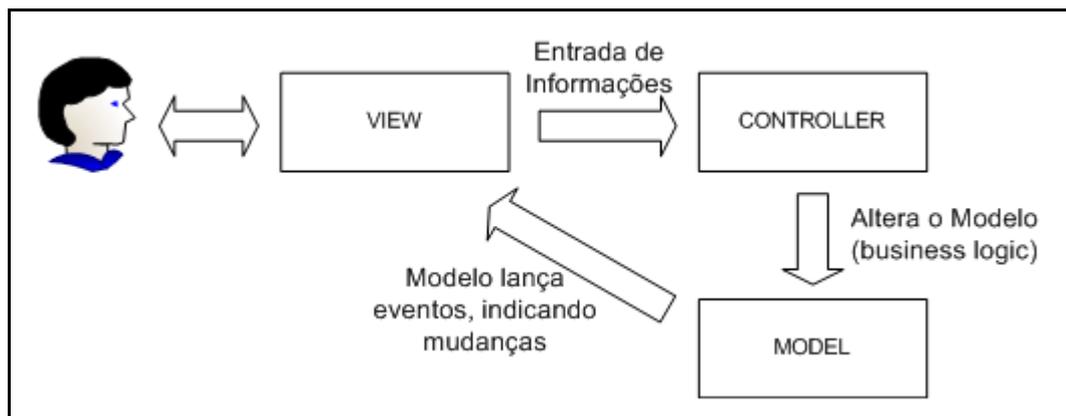
As características da camada controle são:

- a) controlar o fluxo da aplicação;
- b) mapear as ações do usuário;
- c) começar e terminar o fluxo da aplicação.

As características da camada visão são:

- a) apresentar os dados para o usuário;
- b) ser utilizada para entrada e saída de dados.

Segundo Dall'Oglio (2007, p. 478), o fluxo de informações do padrão MVC da seguinte forma: as camadas visão e controle dependem do modelo, o modelo é independente, a camada visão chama a camada controle e aguarda pelo retorno da mesma. Para exemplificar pode-se visualizar a Figura 2.



Fonte: Almeida (2007).

Figura 2 - Fluxo de eventos de informações em uma arquitetura MVC

### 2.3.1.2 Active record

O padrão de projeto *active record* designa cada registro da tabela como um objeto na aplicação e provê métodos de acesso à base de dados. Nesta classe também estão características do modelo de negócio. Para Dall'Oglio (2007, p. 257), esse padrão pode ter grande êxito quando o modelo de dados é semelhante ao modelo de negócio.

O que diferencia o *active record* dos outros padrões *gateway*<sup>2</sup> é o fato dele possuir métodos relativos à lógica de negócios da aplicação.

<sup>2</sup> *gateway* é uma interface que se comunica com um recurso externo escondendo seus detalhes (DALL'OGGIO (2007)).

### 2.3.1.3 Association data mapping

Segundo Dall'Oglio (2007, p. 261) o padrão de projeto *association data mapping* é uma camada de aplicação que separa os objetos das tabelas em si, permitindo transferir dados entre camadas transparentemente, sem a necessidade de métodos de persistência. Dessa forma os objetos é que devem oferecer métodos para a classe verificar o valor de suas propriedades. A classe deverá fornecer métodos de persistência, a busca dos objetos de negócio e também diferentes chaves para a busca destes objetos.

### 2.3.1.4 Front controller

O padrão de projeto *front controller* é o elemento de controle centralizado de uma aplicação MVC que intercepta as requisições dos clientes, seleciona e executa a lógica de negócios apropriada e redireciona a requisição para o próximo passo no fluxo da aplicação.

Para Dall'Oglio (2007, p. 489), *front controller* é um tipo de *script* centralizador, ou seja um *script* que serve para todas as outras páginas. Ou ele pode servir para cada tabela para executar a função de mostrar, incluir ou excluir os dados.

## 2.3.2 AJAX

Segundo Gonçalves (2006, p. 1), AJAX utiliza entre as principais tecnologias o JavaScript e o XML, dessa forma é capaz de tornar o navegador mais interativo, utilizando-se de solicitações assíncronas de informações. Com o AJAX não há necessidade de recarregar a página inteira, mantendo o visual da aplicação e atualizando somente o conteúdo. Em um aplicativo contendo AJAX, uma parte da lógica é movida para o navegador. Dessa forma, no primeiro acesso à página o servidor recebe uma carga maior de dados e depois a carga diminui, pois só atualiza os dados da tela.

## 2.4 WORKFLOW

Segundo Moro (2007), pode-se definir *workflow* como: qualquer tarefa executada em série ou em paralelo por dois ou mais membros de um grupo de trabalho (*workgroup*) visando um objetivo comum.

Conforme Santana (2006, p. 5) especifica, com um sistema *workflow* automatizado, os seguintes benefícios podem ser obtidos:

- a) o trabalho não enguiça nem perde o foco – raramente os requisitantes são chamados para resolver erros ou falhas na administração do processo;
- b) os gerentes podem coordenar a equipe e as tarefas de negócio, tais como desempenhos individuais, otimização de processos e casos especiais, ao invés de se preocupar com a rotina das tarefas. Um exército de secretárias não é mais necessário para entregar e acompanhar um trabalho;
- c) os processos são oficialmente documentados e seguidos exatamente da maneira que foram planejados, garantindo que o trabalho seja executado como previsto, atendendo assim todos os requisitos do negócio;
- d) a melhor pessoa (ou máquina) é alocada para fazer cada caso, sendo que os casos mais importantes são alocados primeiro. Usuários não perdem tempo escolhendo em qual item trabalhar;
- e) processamento paralelo, onde duas ou mais tarefas são executadas ao mesmo tempo, é de longe mais prático que em um *workflow* tradicional (manual).

### 2.4.1 Uso de *workflow* nas organizações

Segundo Araujo (2002), os benefícios de usar o *workflow* nas organizações são:

- a) melhoria no atendimento do cliente;
- b) melhor circulação de documentos;
- c) interoperabilidade e distribuição;
- d) participação e cooperação;
- e) padronização de processos;
- f) memória da organização.

Fatores que devem ser levados em conta na hora de colocar um processo em *workflow*:

- a) valor do processo para organização;
- b) processo diretamente ligado aos objetivos do negócio;
- c) processos que diferenciam a organização;
- d) retorno de investimento.

## 2.5 SISTEMAS WEB

Segundo Angulare (2007), um sistema web nada mais é do que um programa personalizado que funciona exclusivamente na internet, com informações centralizadas em um só banco de dados. A vantagem desse método de trabalho é economizar tempo em atualizações de versões e facilitar a inserção ou edição das informações, deixando-as disponíveis por níveis hierárquicos.

O desenvolvimento de aplicações para web obrigou os desenvolvedores a utilizar uma nova arquitetura de sistemas, abandonando a arquitetura monolítica para se empregar uma arquitetura em camadas. O modelo MVC divide os sistemas web em 3 camadas: modelo, visão e controle. Desta forma, tornam o sistema mais flexível, de modo que partes podem ser alteradas independentemente. Com o emprego desta arquitetura, qualquer alteração em uma determinada camada não influi nas demais, desde que o mecanismo de comunicação entre elas permaneça inalterado.

Os sistemas web possuem diversas vantagens como: manutenções em tempo real, independência de sistema operacional e abrangência mundial. Desta forma, o usuário pode estar trabalhando no seu computador através do acesso a internet e dando continuidade aos processos de gerenciamento de projeto que ele participa na sua empresa, mantendo o fluxo de trabalho ativo.

## 2.6 TRABALHOS CORRELATOS

A seguir são apresentados quatro sistemas relacionados com o sistema de informações para controle de gerenciamento de projetos: gerenciamento de projetos da empresa Iplanner, projeto Dotproject, ferramenta de gerenciamento de tempo de projeto e o aplicativo de

gerenciamento de projetos do SESC.

#### 2.6.1 Sistema de gerenciamento de projetos

O sistema de gerenciamento de projetos desenvolvido pela empresa Iplanner (2007) é flexível, pois, pode ser parametrizado e utilizado em projetos de qualquer área de negócio. É cem por cento web, pois, pode ser acessado de qualquer local e não necessita da instalação de outros programas. Customizado, pois permite personalizar processos, telas e relatórios e pode ser integrado com outros sistemas gerenciais da empresa de acordo com a necessidade do seu negócio. É seguro, pois tem vários níveis de controle de acesso das informações disponibilizadas para garantir segurança dos dados da organização. É gerencial de forma que permite análise gerencial de múltiplos projetos em todas as fases de um projeto. É organizado, pois utiliza banco de dados facilitando a integração com outros sistemas gerenciais da empresa, extração de dados para tomada de decisões e contempla as 9 áreas do conhecimento do PMBOK.

#### 2.6.2 Projeto dotproject

O sistema de gerenciamento de projetos desenvolvido pelos colaboradores do projeto Dotproject-br (2007) é cem por cento web e foi escrito em PHP e utiliza o banco de dados MySQL e o servidor web é o Apache. É multivalentes, sendo desta forma, voltado para ambientes corporativos, pois atende as diversas necessidades de gerentes e escritórios de projetos. As tarefas são comunicadas por correio eletrônico para os envolvidos nos projetos e contempla as 9 áreas do conhecimento do PMBOK.

#### 2.6.3 Ferramenta de gerenciamento de tempo de projetos

Soethe (2004) desenvolveu uma ferramenta de gerenciamento de tempo de projetos utilizando a ferramenta Power Designer para especificação, Interbase 6 para gerenciador de banco de dados e o Delphi 7 como ambiente de programação. A ferramenta foi baseada no gerenciamento de tempo de projetos do manual PMBOK, segunda edição de 2000. Pode ser

utilizado em projetos de qualquer área de negócio

#### 2.6.4 Aplicativo de gerência de projetos

Goetz (2003) desenvolveu um aplicativo de gerência de projetos utilizando uma ferramenta de *workflow* para atender diferentes áreas de negócio, mas com foco na área de informática da empresa SESC do Rio Grande do Sul. Para validação desta integração entre gerência de projetos e o *workflow*, foi desenvolvido um protótipo na ferramenta de *workflow* Lotus Domino da International Business Machines (IBM). Foi utilizada uma metodologia própria de gerenciamento de projetos da empresa.

### 3 DESENVOLVIMENTO DO TRABALHO

O presente capítulo descreve a especificação e implementação do projeto, detalhando aspectos importantes acerca da forma como os padrões de projetos, *frameworks*, componentes e ferramentas foram utilizadas no desenvolvimento do projeto.

#### 3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O sistema de informações para controle do gerenciamento de projetos deverá:

- a) manter informações de projeto como objetivo, visão geral, pessoas envolvidas, orçamento, prazo, cronograma, pré-requisitos e gestor (Requisito Funcional – RF);
- b) comunicar as tarefas de projeto via *workflow*. A cada estágio iniciado será encaminhado um correio eletrônico para as pessoas responsáveis por este estágio (RF);
- c) manter os usuários do sistema de acordo o perfil de acesso (RF);
- d) utilizar o banco de dados MySQL para armazenar os dados gerados pelo usuário (Requisito Não Funcional - RNF);
- e) ser implementado utilizando o *framework* Cake PHP (RNF);
- f) estar disponível para qualquer usuário do sistema através de qualquer navegador de internet (RNF);
- g) ser baseado nas melhores práticas do guia PMBOK (RNF).

#### 3.2 ESPECIFICAÇÃO

Nas subseções seguintes são abordadas as especificações do sistema de informações para controle do gerenciamento de projetos desenvolvido, que se divide em: UML e o modelo entidade-relacionamento. Para a diagramação do projeto foi utilizada a ferramenta JUDE UML pela maior familiaridade.

### 3.2.1 UML

Segundo Bezerra (2007, p. 15), UML é uma linguagem visual para modelar sistemas orientados a objetos. É uma linguagem que define elementos gráficos que podem ser utilizados na modelagem de sistemas. A seguir serão apresentados os diagramas do sistema desenvolvido:

- a) diagramas de caso de uso do sistema;
- b) detalhamento do caso de uso do sistema;
- c) diagrama de estados;
- d) diagrama de atividades do workflow;

#### 3.2.1.1 Diagramas de casos de uso do sistema

Segundo Bezerra (2007, p. 70), o diagrama de caso de uso tem como finalidade ilustrar em nível alto de abstração os elementos externos que interagem com as funcionalidades do sistema.

A seguir serão apresentados os diagramas para cada ator do sistema:

- a) administrador;
- b) gerente de projeto;
- c) analista de projeto;
- d) patrocinador;
- e) analista contábil.

O administrador do sistema tem como função fazer os cadastros básicos do sistema como: usuários, permissões, acessos, departamentos, estados e fornecedores. A Figura 3 apresenta o diagrama do caso de uso do administrador.

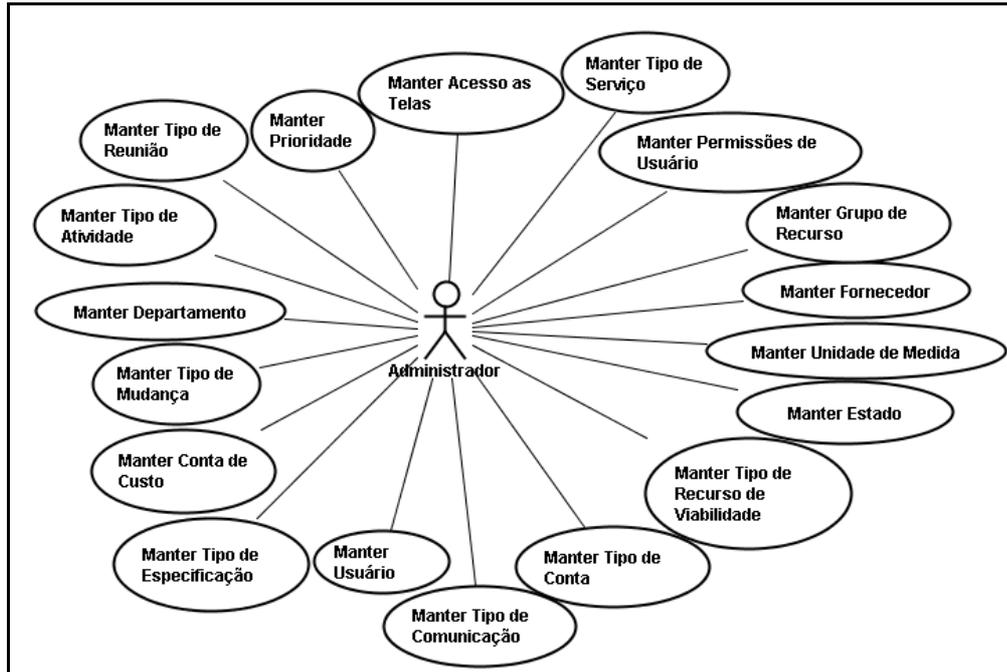


Figura 3 - Diagrama de caso de uso do administrador

O gerente de projeto tem como função cadastrar todas as informações relacionadas ao projeto, enviar informações de projeto para aprovação do patrocinador, designar elementos EAP aos membros do projeto e enviar o estudo de viabilidade para controladoria. O analista de projeto tem como função atualizar seus elementos EAP. A Figura 4 apresenta o diagrama do caso de uso do gerente de projeto e do analista de projeto.

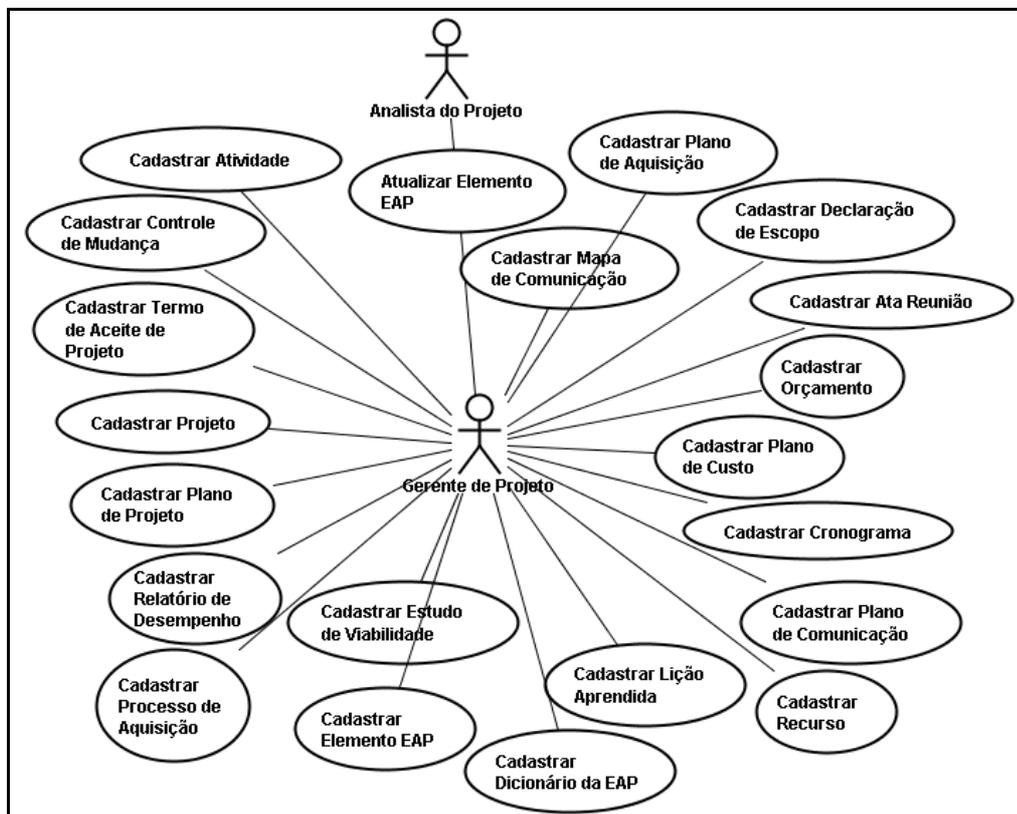


Figura 4 - Diagrama do caso de uso do gerente e do analista do projeto

O patrocinador tem como função aprovar ou reprovar informações do projeto como: processo de aquisição, estudo de viabilidade, projeto e controle de mudança. A Figura 5 apresenta o diagrama de caso de uso do patrocinador.

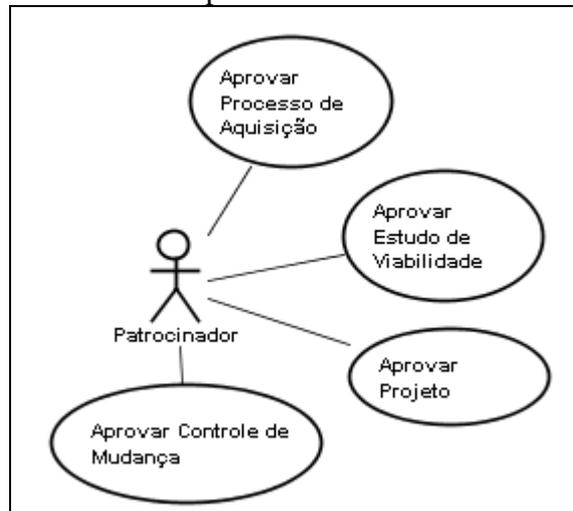


Figura 5 - Diagrama de caso de uso do patrocinador

O analista contábil tem como função classificar contabilmente o estudo de viabilidade. A Figura 6 apresenta o diagrama de caso de uso do analista contábil.

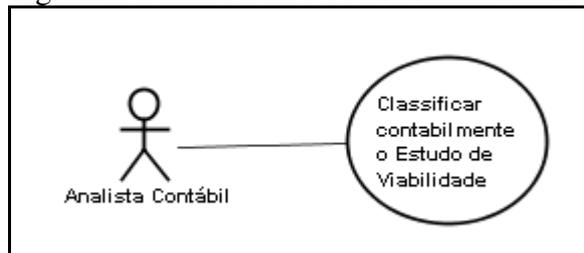


Figura 6 - Diagrama de caso de uso do analista contábil

### 3.2.1.2 Detalhamento dos casos de uso do sistema

Os casos de usos secundários do sistema para o administrador serão apresentados por meio de uma breve descrição que segue:

- a) manter prioridade: permite incluir, alterar, excluir e detalhar a prioridade de um controle de mudança no projeto, por exemplo, baixa, média e alta;
- b) manter acesso as telas: permite incluir, alterar, excluir e detalhar a permissão a um grupo de usuários a uma tela do sistema, por exemplo, grupo administrador acessa a tela de cadastro de prioridade;
- c) manter tipo de serviço: permite incluir, alterar, excluir e detalhar o tipo de serviço que será executado no estudo de viabilidade, por exemplo, análise e programação;
- d) manter permissões de usuário: permite incluir, alterar, excluir e detalhar a permissão de um usuário a um grupo de usuários, por exemplo, usuário agielow

faz parte do grupo administrador;

- e) manter grupo de recurso: permite incluir, alterar, excluir e detalhar o grupo de recurso do projeto, por exemplo, por exemplo, hardware e software;
- f) manter fornecedor: permite incluir, alterar, excluir e detalhar um fornecedor de um projeto, por exemplo, a compra de um computador é feita de um fornecedor;
- g) manter unidade de medida: permite incluir, alterar, excluir e detalhar a unidade medida para um recurso do projeto, por exemplo, kilo, litro e metro;
- h) manter estado: permite incluir, alterar, excluir e detalhar o estado de um projeto, por exemplo, iniciado, planejando, executando e encerrado;
- i) manter tipo de recurso de viabilidade: permite incluir, alterar, excluir e detalhar o tipo de recurso utilizado em um estudo de viabilidade, por exemplo, software e hardware;
- j) manter tipo de conta: permite incluir, alterar, excluir e detalhar o tipo de conta de custo de um departamento, por exemplo, investimento e despesa;
- k) manter tipo de comunicação: permite incluir, alterar, excluir e detalhar o tipo de comunicação do mapa de comunicação do projeto, por exemplo, geral e gerencial;
- l) manter usuário: permite incluir, alterar, excluir e detalhar um usuário do sistema, por exemplo agielow e ggielow;
- m) manter tipo de especificação: permite incluir, alterar, excluir e detalhar o tipo de especificação utilizado no dicionário EAP, por exemplo, geral e detalhada;
- n) manter conta de custo: permite incluir, alterar, excluir e detalhar a conta de custo no qual é contabilizado os custos do projeto, por exemplo, conta de custo de investimento do financeiro;
- o) manter tipo de mudança: permite incluir, alterar, excluir e detalhar o tipo de mudança do controle de mudança do projeto, por exemplo, pessoal e de escopo;
- p) manter departamento: permite incluir, alterar, excluir e detalhar o departamento de usuário cadastrado no sistema, um usuário trabalha em um departamento da empresa;
- q) manter tipo de atividade: permite incluir, alterar, excluir e detalhar o tipo da atividade do projeto, por exemplo, administrativa e de análise;
- r) manter tipo de reunião: permite incluir, alterar, excluir e detalhar o tipo de reunião da ata de reunião do projeto, por exemplo, geral e gerencial.

Os casos de usos secundários do sistema para o gerente de projeto serão apresentados por meio de uma breve descrição que segue:

- a) cadastrar cronograma: permite incluir, alterar, excluir e detalhar o cronograma do projeto que possui as atividades do projeto com a data de entrega para cada atividade;
- b) cadastrar dicionário da EAP: permite incluir, alterar, excluir e detalhar o dicionário EAP, que detalha mais a atividade do projeto;
- c) cadastrar lição aprendida: permite incluir, alterar, excluir e detalhar as lições aprendidas, que são todas as experiências boas e ruins vividas no ambiente de trabalho do projeto e relatadas como lições aprendidas;
- d) cadastrar recurso: permite incluir, alterar, excluir e detalhar os recursos utilizados no projeto, que são os meios para atender as atividades do projeto;
- e) cadastrar plano de comunicação: permite incluir, alterar, excluir e detalhar o plano de comunicação de um projeto, que é determinar as necessidades de informação e de comunicação das partes interessadas do projeto;
- f) cadastrar plano de custo: permite incluir, alterar, excluir e detalhar o plano de custo de um projeto, que é determinar como os custos serão estimados e controlados;
- g) cadastrar ata de reunião: permite incluir, alterar, excluir e detalhar a ata de reunião, assuntos e as ações a serem tomadas que a compõem;
- h) cadastrar declaração de escopo: permite incluir, alterar, excluir e detalhar a declaração de escopo do projeto, que descreve a necessidade de negócio, a justificativa, os requisitos e os limites atuais do projeto;
- i) cadastrar plano de projeto: permite incluir, alterar, excluir e detalhar o planejamento do projeto, que descreve como o projeto deve ser desenvolvido;
- j) cadastrar mapa de comunicação: permite incluir, alterar, excluir e detalhar o mapa de comunicação, permite mapear quais informações serão enviadas para os membros do projeto;

A seguir são apresentados os detalhamentos dos casos de usos principais do sistema, mostrados através de quadros explicativos.

O Quadro 1 apresenta o detalhamento do caso de uso cadastrar projeto.

<b>Cadastrar Projeto</b>	
<b>Ator</b>	Gerente de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir cadastrar um Projeto.
<b>Pré-Condições</b>	1. O estado deve estar cadastrado. 2. O patrocinador deve estar cadastrado como usuário.
<b>Pós-Condições</b>	O projeto está cadastrado no sistema.
<b>Cenário Principal</b>	1. Usuário seleciona o estado do projeto. 2. Usuário informa o título do projeto. 3. Usuário seleciona o patrocinador do projeto. 4. Usuário seleciona a data de prazo do projeto. 5. Usuário seleciona a opção Adicionar. 6. Sistema carrega os passos do <i>workflow</i> para o projeto. 7. Sistema grava o projeto.
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 1 - Detalhamento do caso de uso cadastrar projeto

O Quadro 2 apresenta o detalhamento do caso de uso cadastrar estudo de viabilidade.

<b>Cadastrar Estudo de Viabilidade</b>	
<b>Ator</b>	Gerente de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir cadastrar um Estudo de Viabilidade.
<b>Pré-Condições</b>	1. O tipo de serviço deve estar cadastrado. 2. O projeto deve estar cadastrado.
<b>Pós-Condições</b>	1. O estudo de viabilidade está cadastrado no sistema. 2. O estudo de viabilidade deve ser enviado ao analista contábil.
<b>Cenário Principal</b>	1. Usuário seleciona o tipo de serviço. 2. Usuário seleciona o projeto. 3. Usuário seleciona a data de validade inicial. 4. Usuário seleciona a data de validade final. 5. Usuário informa o texto de encerramento. 6. Usuário seleciona um ou mais usuários. 7. Usuário seleciona a opção Adicionar. 8. Sistema grava o estudo de viabilidade.
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 2 - Detalhamento do caso de uso cadastrar estudo de viabilidade

O Quadro 3 apresenta o detalhamento do caso de uso classificar contabilmente o estudo de viabilidade.

<b>Classificar contabilmente o Estudo de Viabilidade</b>	
<b>Ator</b>	Analista Contábil
<b>Descrição</b>	Este caso de uso tem por objetivo permitir classificar contabilmente o Estudo de Viabilidade.
<b>Pré-Condições</b>	1. O estudo de viabilidade deve estar cadastrado. 2. Receber o estudo de viabilidade do gerente de projeto. 3. A conta de custo deve estar cadastrada.
<b>Pós-Condições</b>	O estudo de viabilidade está classificado no sistema.
<b>Cenário Principal</b>	1. Usuário seleciona o projeto. 2. Usuário seleciona uma ou mais contas de custo. 3. Usuário seleciona a opção Adicionar. 4. Sistema grava o estudo de viabilidade. 5. Sistema envia correio eletrônico para gerente de projeto.
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 3 - Detalhamento do caso de uso classificar contabilmente o estudo de viabilidade

O Quadro 4 apresenta o detalhamento do caso de uso aprovar estudo de viabilidade.

<b>Aprovar o Estudo de Viabilidade</b>	
<b>Ator</b>	Patrocinador
<b>Descrição</b>	Este caso de uso tem por objetivo permitir aprovar o Estudo de Viabilidade.
<b>Pré-Condições</b>	1. O estudo de viabilidade deve estar cadastrado. 2. Receber o estudo de viabilidade do gerente de projeto.
<b>Pós-Condições</b>	O estudo de viabilidade está aprovado ou reprovado no sistema.
<b>Cenário Principal</b>	1. Usuário seleciona a opção Aprovar Estudo. 2. Sistema grava o estudo de viabilidade. 3. Sistema envia correio eletrônico para gerente de projetos.
<b>Cenário Alternativo</b>	1. Usuário seleciona a opção Reprovar Estudo. 2. Sistema grava o estudo de viabilidade. 3. Sistema envia correio eletrônico para gerente de projetos.
<b>Exceções</b>	Não há.

Quadro 4 - Detalhamento do caso de uso aprovar o estudo de viabilidade

O Quadro 5 apresenta o detalhamento do caso de uso cadastrar atividade.

<b>Cadastrar Atividade</b>	
<b>Ator</b>	Gerente de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir cadastrar Atividade.
<b>Pré-Condições</b>	O projeto deve estar cadastrado.
<b>Pós-Condições</b>	A atividade está cadastrada no sistema.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona o projeto.</li> <li>2. Usuário informa a atividade pai.</li> <li>3. Usuário informa o valor de custo da atividade</li> <li>4. Usuário informa os dias de prazo.</li> <li>5. Usuário informa o valor realizado.</li> <li>6. Usuário seleciona um ou mais orçamentos.</li> <li>7. Usuário seleciona um ou mais cronogramas.</li> <li>8. Usuário seleciona a opção Adicionar.</li> <li>9. Sistema grava a atividade.</li> </ol>
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 5 - Detalhamento do caso de uso cadastrar atividade

O Quadro 6 apresenta o detalhamento do caso de uso cadastrar elemento EAP.

<b>Cadastrar Elemento EAP</b>	
<b>Ator</b>	Gerente de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir cadastrar Elemento EAP.
<b>Pré-Condições</b>	A atividade deve estar cadastrada.
<b>Pós-Condições</b>	O elemento EAP está cadastrado no sistema.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona a atividade.</li> <li>2. Usuário informa o valor planejado.</li> <li>3. Usuário informa o valor agregado.</li> <li>4. Usuário informa o valor real.</li> <li>5. Usuário informa o valor percentual concluído.</li> <li>6. Usuário seleciona a data planejada.</li> <li>7. Usuário seleciona a data efetiva.</li> <li>8. Usuário informa os dias de atraso.</li> <li>9. Usuário informa o solicitante.</li> <li>10. Usuário informa o responsável.</li> <li>11. Usuário seleciona a opção Adicionar.</li> <li>12. Sistema grava o elemento EAP.</li> <li>13. Sistema envia correio eletrônico para o solicitante e para o responsável.</li> </ol>
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 6 - Detalhamento do caso de uso cadastrar elemento EAP

O Quadro 7 apresenta o detalhamento do caso de uso atualizar elemento EAP.

<b>Atualizar Elemento EAP</b>	
<b>Ator</b>	Analista de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir atualizar Elemento EAP.
<b>Pré-Condições</b>	Receber o Elemento EAP do gerente de projeto.
<b>Pós-Condições</b>	O elemento EAP está atualizado no sistema.
<b>Cenário Principal</b>	1. Usuário informa o valor de percentual concluído. 2. Usuário seleciona a opção Atualizar. 3. Sistema grava o elemento EAP.
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 7 - Detalhamento do caso de uso atualizar elemento EAP

O Quadro 8 apresenta o detalhamento do caso de uso cadastrar plano de aquisição.

<b>Cadastrar Plano de Aquisição</b>	
<b>Ator</b>	Gerente de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir cadastrar Plano de Aquisição.
<b>Pré-Condições</b>	O projeto deve estar cadastrado.
<b>Pós-Condições</b>	O plano de aquisição está cadastrado no sistema.
<b>Cenário Principal</b>	1. Usuário seleciona o projeto. 2. Usuário informa a descrição do plano de aquisição. 3. Usuário informa o documento de referência. 4. Usuário informa a descrição de avaliação de fornecedores. 5. Usuário seleciona a opção Adicionar. 6. Sistema grava o plano de aquisição.
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 8 - Detalhamento do caso de uso cadastrar plano de aquisição

O Quadro 9 apresenta o detalhamento do caso de uso cadastrar processo de aquisição.

<b>Cadastrar Processo de Aquisição</b>	
<b>Ator</b>	Gerente de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir cadastrar Processo de Aquisição.
<b>Pré-Condições</b>	1. O plano de aquisição deve estar cadastrado. 2. O fornecedor deve estar cadastrado.
<b>Pós-Condições</b>	O processo de aquisição está cadastrado no sistema.
<b>Cenário Principal</b>	1. Usuário seleciona o plano de aquisição. 2. Usuário seleciona o fornecedor. 3. Usuário informa o valor de orçamento. 4. Usuário informa os dias de prazo. 5. Usuário informa a descrição do processo aquisição. 6. Usuário seleciona a opção Adicionar. 7. Sistema grava processo de aquisição.
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 9 - Detalhamento do caso de uso cadastrar processo de aquisição

O Quadro 10 apresenta o detalhamento do caso de uso aprovar processo de aquisição.

<b>Aprovar Processo de Aquisição</b>	
<b>Ator</b>	Patrocinador
<b>Descrição</b>	Este caso de uso tem por objetivo aprovar Processo de Aquisição.
<b>Pré-Condições</b>	O processo de aquisição deve estar cadastrado.
<b>Pós-Condições</b>	O processo de aquisição está aprovado ou reprovado no sistema.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona a opção Aprovar Processo de Aquisição.</li> <li>2. Sistema envia um correio eletrônico para o gerente de projeto.</li> <li>3. Sistema grava o processo de aquisição.</li> </ol>
<b>Cenário Alternativo</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona a opção Reprovar Processo de Aquisição.</li> <li>2. Sistema envia um correio eletrônico para o gerente de projeto.</li> <li>3. Sistema grava o processo de aquisição.</li> </ol>
<b>Exceções</b>	Não há.

Quadro 10 - Detalhamento do caso de uso aprovar processo de aquisição

O Quadro 11 apresenta o detalhamento do caso de uso cadastrar orçamento.

<b>Cadastrar Orçamento</b>	
<b>Ator</b>	Gerente de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir cadastrar Orçamento.
<b>Pré-Condições</b>	O projeto deve estar cadastrado.
<b>Pós-Condições</b>	O orçamento está cadastrado no sistema.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona o projeto.</li> <li>2. Usuário informa o valor de reserva.</li> <li>3. Usuário informa o valor de administração.</li> <li>4. Usuário seleciona uma ou mais atividades.</li> <li>5. Usuário seleciona um ou mais recursos.</li> <li>6. Usuário seleciona a opção Adicionar.</li> <li>7. Sistema calcula o valor do projeto.</li> <li>8. Sistema calcula o valor total.</li> <li>9. Sistema grava o orçamento.</li> </ol>
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 11 - Detalhamento do caso de uso cadastrar orçamento

O Quadro 12 apresenta o detalhamento do caso de uso cadastrar relatório de desempenho.

<b>Cadastrar Relatório de Desempenho</b>	
<b>Ator</b>	Gerente de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir cadastrar Relatório de Desempenho.
<b>Pré-Condições</b>	O projeto deve estar cadastrado.
<b>Pós-Condições</b>	O relatório de desempenho está cadastrado no sistema.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona o projeto.</li> <li>2. Usuário informa a descrição do relatório de desempenho.</li> <li>3. Usuário seleciona a opção Adicionar.</li> <li>4. Sistema busca todos os elementos EAPs atrasados e anexos ao relatório de desempenho.</li> <li>5. Sistema grava o relatório de desempenho.</li> </ol>
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 12 - Detalhamento do caso de uso cadastrar relatório de desempenho

O Quadro 13 apresenta o detalhamento do caso de uso cadastrar controle de mudança.

<b>Cadastrar Controle de Mudança</b>	
<b>Ator</b>	Gerente de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir cadastrar Controle de Mudança.
<b>Pré-Condições</b>	<ol style="list-style-type: none"> <li>1. O projeto deve estar cadastrado.</li> <li>2. O tipo de mudança deve estar cadastrado.</li> <li>3. A prioridade deve estar cadastrada.</li> </ol>
<b>Pós-Condições</b>	O controle de mudança está cadastrado no sistema.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona o projeto.</li> <li>2. Usuário seleciona o tipo de mudança.</li> <li>3. Usuário seleciona a prioridade</li> <li>4. Usuário informa descrição do controle de mudança.</li> <li>5. Usuário informa a justificativa.</li> <li>6. Usuário seleciona o prazo da mudança.</li> <li>7. Usuário informa a descrição do impacto.</li> <li>8. Usuário seleciona a data de comunicação.</li> <li>9. Usuário seleciona a data de envio de recursos.</li> <li>10. Usuário seleciona a opção Adicionar.</li> <li>11. Sistema grava o controle de mudança.</li> </ol>
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 13 - Detalhamento do caso de uso cadastrar controle de mudança

O Quadro 14 apresenta o detalhamento do caso de uso aprovar controle de mudança.

<b>Aprovar Controle de Mudança</b>	
<b>Ator</b>	Patrocinador
<b>Descrição</b>	Este caso de uso tem por objetivo permitir aprovar Controle de Mudança.
<b>Pré-Condições</b>	1. O controle de mudança deve estar cadastrado. 2. Receber o controle de mudança do gerente de projeto.
<b>Pós-Condições</b>	O controle de mudança está aprovado ou reprovado no sistema.
<b>Cenário Principal</b>	1. Usuário seleciona a opção Aprovar Controle de Mudança. 2. Sistema grava o controle de mudança. 3. Sistema envia correio eletrônico para gerente de projeto.
<b>Cenário Alternativo</b>	1. Usuário seleciona a opção Reprovar Controle de Mudança. 2. Sistema grava o controle de mudança. 3. Sistema envia correio eletrônico para gerente de projeto.
<b>Exceções</b>	Não há.

Quadro 14 - Detalhamento do caso de uso aprovar controle de mudança

O Quadro 15 apresenta o detalhamento do caso de uso cadastrar termo de aceite de projeto.

<b>Cadastrar Termo de Aceite de Projeto</b>	
<b>Ator</b>	Gerente de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir cadastrar Termo de Aceite de Projeto.
<b>Pré-Condições</b>	O projeto deve estar cadastrado.
<b>Pós-Condições</b>	O termo de aceite está cadastrado no sistema.
<b>Cenário Principal</b>	1. Usuário seleciona o projeto. 2. Usuário informa o local. 3. Usuário informa o resumo do projeto 4. Usuário seleciona nenhum ou mais usuários. 5. Usuário seleciona a opção Adicionar. 6. Sistema grava o termo de aceite de projeto.
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 15 - Detalhamento do caso de uso cadastrar termo de aceite de projeto

O Quadro 16 apresenta o detalhamento do caso de uso aprovar projeto.

<b>Aprovar Projeto</b>	
<b>Ator</b>	Patrocinador
<b>Descrição</b>	Este caso de uso tem por objetivo aprovar Projeto.
<b>Pré-Condições</b>	O termo de aceite do projeto deve estar cadastrado.
<b>Pós-Condições</b>	O projeto está aprovado ou reprovado no sistema.
<b>Cenário Principal</b>	1. Usuário seleciona a opção Aprovar Projeto. 2. Sistema grava o termo de aceite de projeto. 3. Sistema envia correio eletrônico para gerente de projeto.
<b>Cenário Alternativo</b>	1. Usuário seleciona a opção Reprovar Projeto. 2. Sistema grava o termo de aceite de projeto. 3. Sistema envia correio eletrônico para gerente de projeto.
<b>Exceções</b>	Não há.

Quadro 16 - Detalhamento do caso de uso aprovar projeto

O Quadro 17 apresenta o detalhamento do caso de uso encerrar projeto.

<b>Encerrar Projeto</b>	
<b>Ator</b>	Gerente de Projeto
<b>Descrição</b>	Este caso de uso tem por objetivo permitir encerrar Projeto.
<b>Pré-Condições</b>	1. O projeto deve estar cadastrado. 2. O projeto deve estar aprovado no sistema.
<b>Pós-Condições</b>	O projeto está encerrado no sistema.
<b>Cenário Principal</b>	1. Usuário seleciona o projeto. 2. Usuário selecionado o estado do projeto como encerrado. 3. Usuário seleciona a opção Salvar. 4. Sistema grava o projeto.
<b>Cenário Alternativo</b>	Não há.
<b>Exceções</b>	Não há.

Quadro 17 - Detalhamento do caso de uso encerrar projeto

### 3.2.1.3 Diagrama de estados

Segundo Bezerra (2007, p. 287), o diagrama de estados é útil para analisar as transições entre estados de objetos de um sistema, desta forma prevê todas as operações possíveis em função de eventos que podem ocorrer.

Para um projeto cadastrado há 4 estados que são:

- a) iniciado, quando o gerente de projeto cadastra um novo projeto;
- b) planejando, quando o estudo de viabilidade foi aprovado pelo patrocinador;
- c) executando, quando o projeto já teve o planejamento finalizado;

d) encerrado, quando o termo de aceite de projeto foi aprovado.

A Figura 7 apresenta o diagrama de estados de um projeto no sistema.

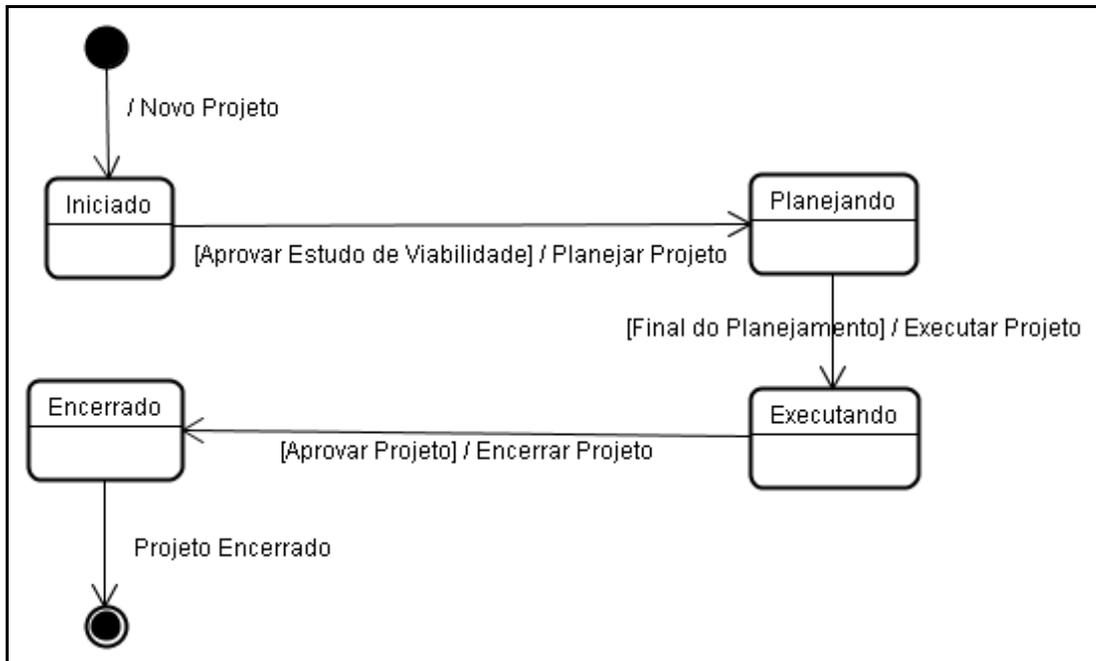


Figura 7 - Diagrama de estados de um projeto

#### 3.2.1.4 Diagrama de atividades do *workflow*

O diagrama de atividades do *workflow* apresenta o fluxo de trabalho do gerenciamento de projeto no sistema SICGP (Figura 8).

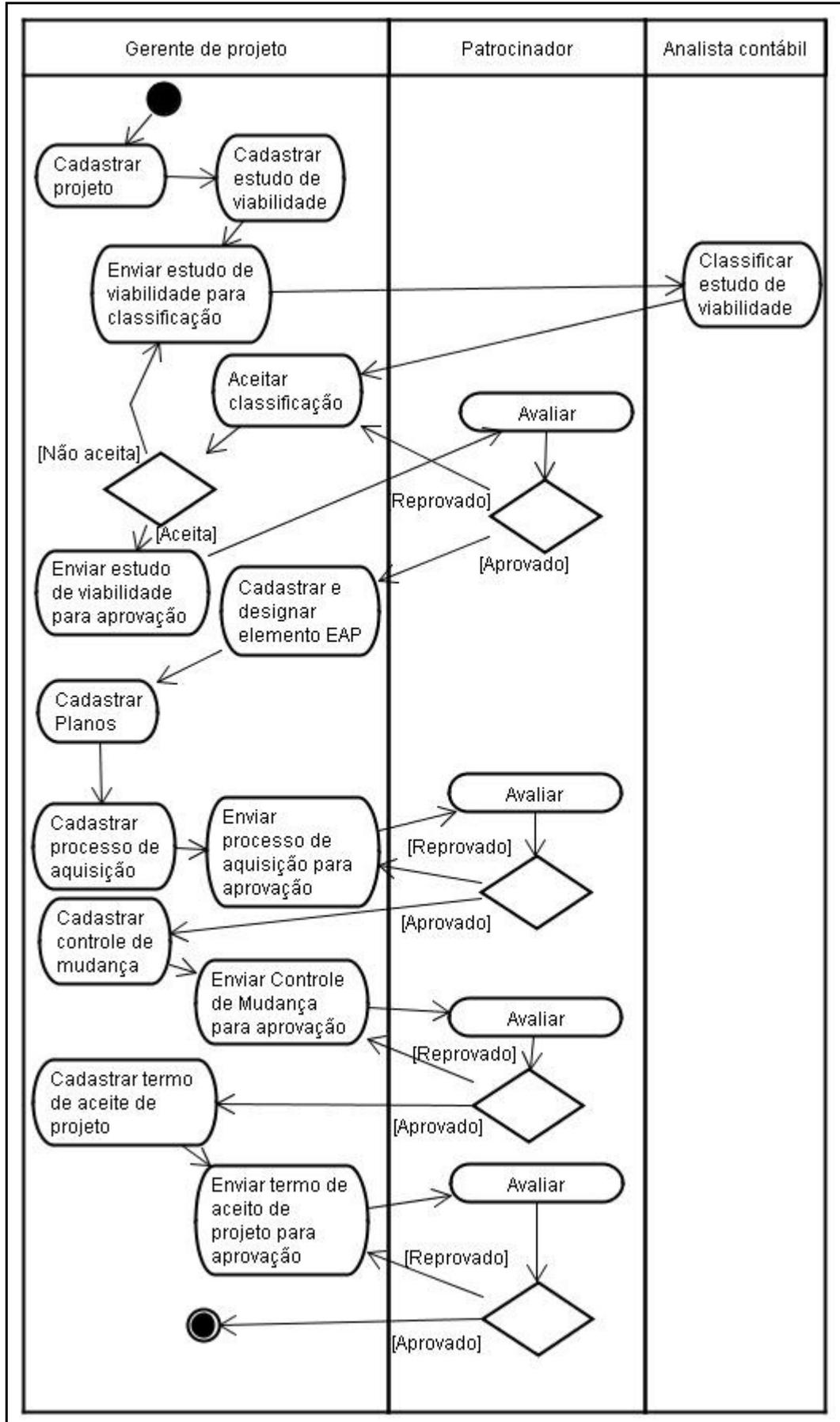


Figura 8 - Diagrama de atividades do workflow



### 3.3 IMPLEMENTAÇÃO

A seguir são mostradas as ferramentas utilizadas no desenvolvimento do sistema, bem como a implementação do sistema de informações para controle do gerenciamento de projetos.

#### 3.3.1 Ferramentas utilizadas

O sistema de informações para controle de gerenciamento de projetos foi desenvolvido em PHP utilizando como banco de dados o MySQL através do *framework* Cake PHP. Para administração de base de dados MySQL foi utilizado o phpmyadmin que vem no pacote WAMP. Para geração das telas foi utilizado o script PHP bake do Cake PHP.

##### 3.3.1.1 Phpmyadmin

Segundo o Phpmyadmin (2008), o phpmyadmin é uma ferramenta desenvolvida em PHP para gerenciamento do MySQL. Com esta ferramenta é possível executar várias funções com o MySQL, como criar e deletar tabelas, renomear, editar campos, trabalhar com chaves primárias, rodar *dumps*, exportar dados.

A Figura 10 mostra a interface do phpmyadmin.

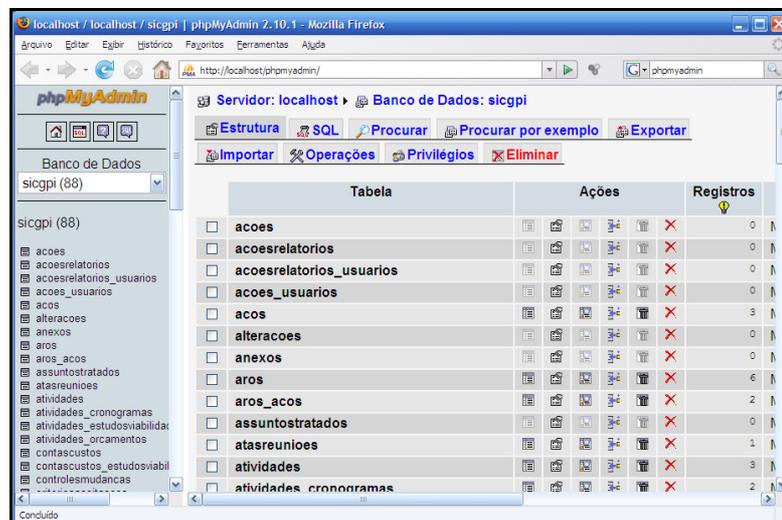


Figura 10 - Interface do phpMyAdmin

### 3.3.1.2 Bake

Para Minetto (2007, p. 31), o bake é um script PHP executável via linha de comando que roda no Windows e no Linux, e foi utilizado para gerar as telas CRUD do sistema. O bake foi essencial para o desenvolvimento do trabalho, pois ele gerou todas as classes do sistema.

A seguir na Figura 11, visualiza-se a interface de geração de tela do sistema no bake.



```

C:\WINDOWS\system32\cmd.exe - php cake/scripts/bake.php -app sicgpi
C:\wamp\www\cake1.1>php cake/scripts/bake.php -app sicgpi

CAKEPHP BAKE
-----
Baking...
-----
Name: sicgpi
Path: C:\wamp\www\cake1.1\sicgpi
-----
[M]odel
[C]ontroller
[U]iew
What would you like to Bake? <M/U/C>
> -
  
```

Figura 11 - Interface do bake do Cake PHP

### 3.3.2 Implementação do SICGP

A seguir são mostradas as camadas MVC do sistema e as propriedades mais utilizadas em cada camada.

#### 3.3.2.1 Camadas MVC no sistema

O funcionamento MVC do Cake PHP começa pelo usuário fazendo um pedido nos métodos da camada controle, que pede os dados para camada modelo, o modelo envia os dados para camada controle que mostra os dados na camada visão conforme o *template*, como demonstrado na Figura 12.

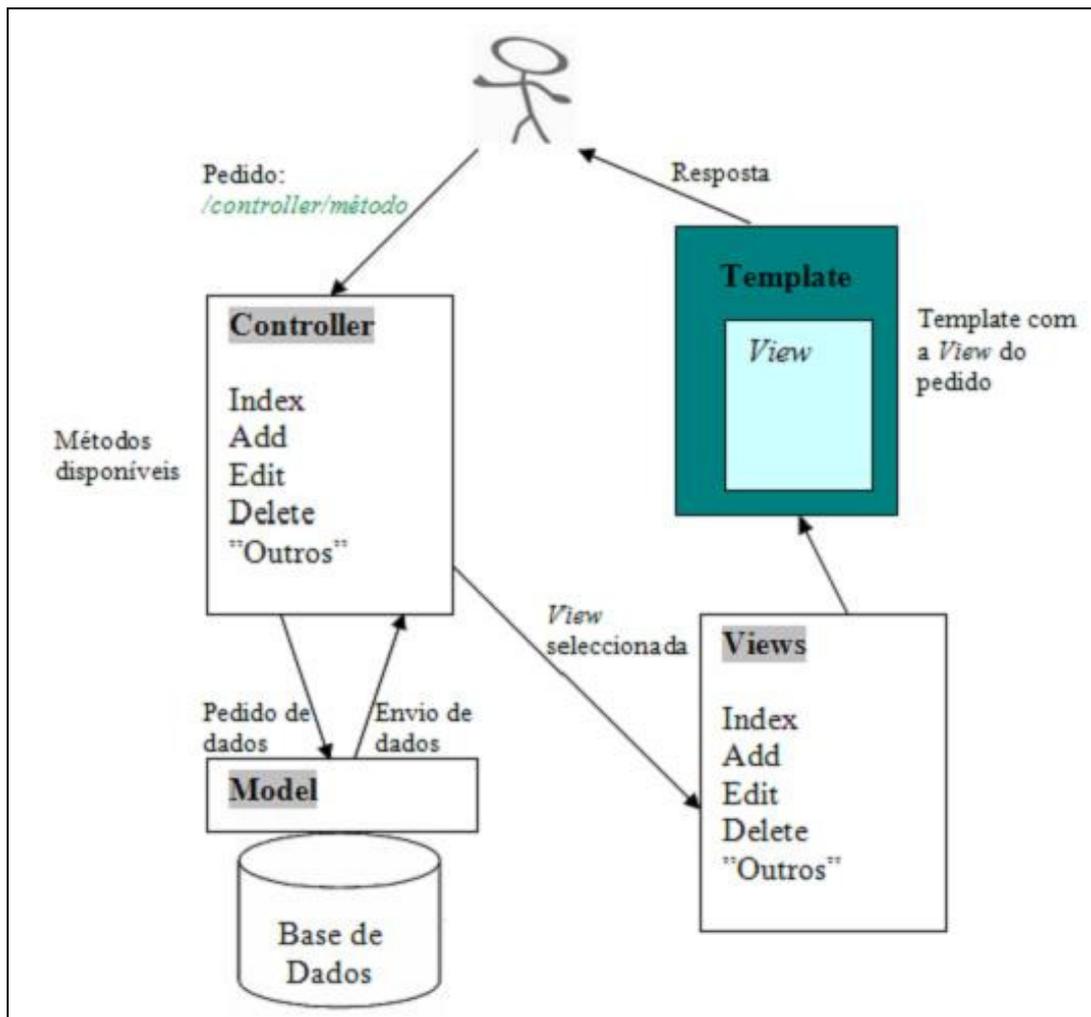


Figura 12 - Funcionamento do MVC do Cake PHP

### 3.3.2.1.1 Camada modelo

Na camada modelo que são definidos quais campos são obrigatórios, como apresenta o código na Figura 13.

Na camada modelo são definidas também quais associações esta classe tem com outras classes do sistema, como na Figura 14, Figura 15 e Figura 16.

Para fins de auditoria de usuário foi complementada a parte básica do *framework* que grava a data de inclusão no campo `created` e a data de modificação de um registro no campo `modified`, utilizando a função `beforeSave()` para gravar também o usuário destas duas operações de auditoria nos campos na mesma ordem `user_created` e `user_modified`. Na Figura 17 está a codificação exemplo do uso da auditoria de usuário.

```

0001 <?php
0002 class Projeto extends AppModel {
0003
0004     var $name = 'Projeto';
0005     var $validate = array(
0006         'estado_id' => VALID_NOT_EMPTY, // campos consistidos pela tela
0007         'ds_titulo' => VALID_NOT_EMPTY, // campos consistidos pela tela
0008         'dt_abertura' => VALID_NOT_EMPTY, // campos consistidos pela tela
0009         'dt_prazo' => VALID_NOT_EMPTY, // campos consistidos pela tela
0010     );
0011

```

Figura 13 - Codificação dos campos obrigatórios na camada modelo

```

0013     var $belongsTo = array( //pertence a classe Estado
0014         'Estado' =>
0015             array('className' => 'Estado',
0016                 'foreignKey' => 'estado_id',
0017                 'conditions' => '',
0018                 'fields' => '',
0019                 'order' => '',
0020                 'counterCache' => ''
0021             ),

```

Figura 14 - Codificação que a classe pertence a classe estado

```

0033     var $hasMany = array( // possui nenhuma ou mais classes Anexo
0034         'Anexo' =>
0035             array('className' => 'Anexo',
0036                 'foreignKey' => 'projeto_id',
0037                 'conditions' => '',
0038                 'fields' => '',
0039                 'order' => '',
0040                 'limit' => '',
0041                 'offset' => '',
0042                 'dependent' => '',
0043                 'exclusive' => '',
0044                 'finderQuery' => '',
0045                 'counterQuery' => ''
0046             ),

```

Figura 15 - Codificação que a classe possui nenhuma ou mais classes anexo

Na Figura 18 é apresentado o uso das funções `query()` e `execute()`, estas possibilitam o uso de comandos SQL para inserir, alterar, excluir e selecionar informações das tabelas do sistema.

```

0048     $hasAndBelongsToMany = array( // a classe Acao pode ter e
0049         'Acao' => // pertencer a muitas classes Usuario
0050             array('className' => 'Acao',
0051                 'joinTable' => 'acoes_usuarios',
0052                 'foreignKey' => 'usuario_id',
0053                 'associationForeignKey' => 'acao_id',
0054                 'conditions' => '',
0055                 'fields' => '',
0056                 'order' => '',
0057                 'limit' => '',
0058                 'offset' => '',
0059                 'unique' => '',
0060                 'finderQuery' => '',
0061                 'deleteQuery' => '',
0062                 'insertQuery' => ''
0063             ),

```

Figura 16 - Codificação que a classe acao pode ter e pertencer a muitas classes usuário

```

0190     }
0191     function beforeSave(){ // antes de salvar os dados
0192         if (!isset($this->data["Usuario"]["id"])){
0193             $this->data["Usuario"]["user_created"] = $_SESSION['Usuario']; //usuario logado
0194         }
0195         $this->data["Usuario"]["user_modified"] = $_SESSION['Usuario']; //usuario logado
0196         return true;
0197     }
0198 }
0199 ?>

```

Figura 17 - Codificação da função que grava a auditoria dos usuários

```

0240 // pega no passo workflow a mensagem, a pessoa responsavel e coloca pra enviar mensagem
0241 function enviaMensagem($sit, $id, $usuario) {
0242     $workflow = $this->query("SELECT user_responsavel, mensagem_inicial FROM passosworkflows WHERE
0243     projeto_id = $id and nr_passo = $sit");
0244     // atualiza a data e hora de expiracao
0245     $this->execute("UPDATE passosworkflows SET dt_ultima_expiracao = NOW() WHERE
0246     projeto_id = $id and nr_passo = $sit");
0247     // atualiza a data de fechamento do passo anterior quando estiver no passo 2
0248     if ($sit > 1){
0249         $this->execute("UPDATE passosworkflows SET dt_termino = NOW()
0250         WHERE projeto_id = $id and nr_passo = ($sit -1)");
0251     }
0252     $user_responsavel = $workflow [0]['passosworkflows']['user_responsavel'];
0253     $mensagem_inicial = $workflow [0]['passosworkflows']['mensagem_inicial'];
0254     // 1 classificar contabilmente
0255     if ($sit == 1) {
0256         $this->execute("INSERT INTO mensagens (projeto_id, ds_mensagem, dt_mensagem, nm_envolvidos) VALUES
0257         ($id, '$mensagem_inicial', NOW(), '$user_responsavel' )");
0258     }
0259     else if ($sit == 2){
0260         $this->execute("INSERT INTO mensagens (projeto_id, ds_mensagem, dt_mensagem, nm_envolvidos) VALUES
0261         ($id, '$mensagem_inicial', NOW(), '$user_responsavel' )");
0262     }
0263     else if ($sit == 3){
0264         $this->execute("INSERT INTO mensagens (projeto_id, ds_mensagem, dt_mensagem, nm_envolvidos) VALUES
0265         ($id, '$mensagem_inicial', NOW(), '$user_responsavel' )");
0266     }
0267     else if ($sit == 4){
0268         $this->execute("INSERT INTO mensagens (projeto_id, ds_mensagem, dt_mensagem, nm_envolvidos) VALUES
0269         ($id, '$mensagem_inicial', NOW(), '$user_responsavel' )");
0270     }
0271 }

```

Figura 18 - Codificação usando as funções query e execute do cake php

### 3.3.2.1.2 Camada controle

Na camada controle sempre existem, no mínimo, cinco funções padrões de toda classe que são:

- index, que permite visualizar todos os registros de uma tabela;
- view, que permite visualizar no detalhe o registro de uma tabela;
- add, que permite adicionar um novo registro da tabela;
- edit, que permite alterar um registro da tabela;
- delete, que permite excluir um registro da tabela.

Na camada controle é que a mensagem de cabeçalho é gerada. Caso o usuário não preencha os campos obrigatórios o sistema vai emitir uma mensagem para preencher estes campos. Segue a Figura 19 com o exemplo codificado.

```

} else {
    // coloca o cabeçalho da mensagem
    $this->Session->setFlash('Por favor, corrija os erros abaixo.');
```

// le os dados novamente para corrigir os erros

```

    $this->set('usuarios', $this->Dicionarioseap->Usuario->generateList(null,'ds_usuario', null, '{n}.
    if(empty($this->data['Usuario']['Usuario'])) { $this->data['Usuario']['Usuario'] = null; }
    $this->set('selectedUsuarios', $this->data['Usuario']['Usuario']);
    $this->set('atividades', $this->Dicionarioseap->Atividade->generateList("Atividade.user_created =
    $this->set('projetos', $this->Dicionarioseap->Projeto->generateList("Projeto.user_created = '$usua
    $this->set('tipospecificacoes', $this->Dicionarioseap->Tipospecificacao->generateList(null,'ds
}

```

Figura 19 - Codificação da classe consistindo campos obrigatórios

A Figura 20 apresenta a função `index` de uma classe com filtro para que o usuário visualize somente os registros por ele criados.

```

0003
0004     var $name = 'Dicionarioseaps';
0005     var $helpers = array('Html', 'Form', 'Ajax', 'Javascript');
```

0006

```

0007     function index() {
0008         $usuario = $_SESSION['Usuario'];
0009         $this->Dicionarioseap->recursive = 0;
0010         $this->set('dicionarioseaps', $this->Dicionarioseap->findAll("Dicionarioseap.user_created = '$usuario");
0011     }
0012

```

Figura 20 - Codificação da função `index` de uma classe

A Figura 21 apresenta a função `view` de uma classe.

```

0011     }
0012
0013     function view($id = null) {
0014         if(!$id) {
0015             $this->Session->setFlash('ID inválido para Dicionarioseap.');
```

0016 \$this->redirect('/dicionarioseaps/index');

```

0017         }
0018         $this->set('dicionarioseap', $this->Dicionarioseap->read(null, $id));
0019     }
0020

```

Figura 21 - Codificação da função `view` de uma classe

A Figura 22 apresenta a função `add` de uma classe e nas classes associadas tem um filtro para que o usuário visualize somente os registros por ele criados.

```

0021     function add() {
0022         $usuario = $_SESSION['Usuario'];
0023         if(empty($this->data) {
0024             $this->set('usuarios', $this->Dicionarioseap->Usuario->generateList(null,'ds_usuario', null, '{n
0025             $this->set('selectedUsuarios', null);
0026             $this->set('atividades', $this->Dicionarioseap->Atividade->generateList("Atividade.user_created =
0027             $this->set('projetos', $this->Dicionarioseap->Projeto->generateList("Projeto.user_created = '$usua
0028             $this->set('tipospecificacoes', $this->Dicionarioseap->Tipospecificacao->generateList(null,'
0029             $this->render();
0030         } else {
0031             $this->cleanUpFields();
0032             if($this->Dicionarioseap->save($this->data) {
0033                 $this->Session->setFlash('Dicionarioseap inserido com sucesso.');
```

0034 \$this->redirect('/dicionarioseaps/index');

```

0035             } else {
0036                 $this->Session->setFlash('Por favor, corrija os erros abaixo.');
```

0037 \$this->set('usuarios', \$this->Dicionarioseap->Usuario->generateList(null,'ds\_usuario', null,

0038 if(empty(\$this->data['Usuario']['Usuario'])) { \$this->data['Usuario']['Usuario'] = null; }

0039 \$this->set('selectedUsuarios', \$this->data['Usuario']['Usuario']);

0040 \$this->set('atividades', \$this->Dicionarioseap->Atividade->generateList("Atividade.user\_crea

0041 \$this->set('projetos', \$this->Dicionarioseap->Projeto->generateList("Projeto.user\_created =

0042 \$this->set('tipospecificacoes', \$this->Dicionarioseap->Tipospecificacao->generateList(nu

```

0043             }
0044         }
0045     }

```

Figura 22 - Codificação da função `add` de uma classe

A Figura 23 apresenta a função edit de uma classe e nas classes associadas tem um filtro para que o usuário visualize somente os registros por ele criados.

```

0047 function edit($id = null) {
0048     $usuario = $_SESSION['Usuario'];
0049     if(empty($this->data)) {
0050         if(!$id) {
0051             $this->Session->setFlash('Id inválido para Dicionarioseap');
0052             $this->redirect('/dicionarioseaps/index');
0053         }
0054         $this->data = $this->Dicionarioseap->read(null, $id);
0055         $this->set('usuarios', $this->Dicionarioseap->Usuario->generateList(null,'ds_usuario', null, '{n
0056         if(empty($this->data['Usuario'])) { $this->data['Usuario'] = null; }
0057         $this->set('selectedUsuarios', $this->_selectedArray($this->data['Usuario']));
0058         $this->set('atividades', $this->Dicionarioseap->Atividade->generateList("Atividade.user_created =
0059         $this->set('projetos', $this->Dicionarioseap->Projeto->generateList("Projeto.user_created = '$usu
0060         $this->set('tipospecificacoes', $this->Dicionarioseap->Tipospecificacao->generateList(null,'d
0061     } else {
0062         $this->cleanUpFields();
0063         if($this->Dicionarioseap->save($this->data) {
0064             $this->Session->setFlash('Dicionarioseap editado com sucesso.');
```

Figura 23 - Codificação da função edit de uma classe

A Figura 24 apresenta a função delete de uma classe.

```

0077
0078 function delete($id = null) {
0079     if(!$id) {
0080         $this->Session->setFlash('ID inválido para Dicionarioseap');
```

Figura 24 - Codificação da função delete de uma classe

Na camada controle podem ser gravadas informações que os usuários não têm acesso como demonstra a Figura 25 com a função enviarcontabilide(\$id).

```

0099 function enviarcontabilide($id) {
0100     $this->checkAccess();
0101     $this->set('estudosviabilidade', $this->Estudosviabilidade->read(null, $id));
0102     $someone = $this->Estudosviabilidade->Usuario->findByDs_usuario($_SESSION['Usuario']);
0103     $projeto = $this->Estudosviabilidade->findById($id);
0104     //grava como enviado para contabilidade
0105     $dados = array("Estudosviabilidade"=>
0106     array(
0107         "id"=>$id,
0108         "log_envio_contabilidade"=>1
0109     )
0110     );
0111     $this->Estudosviabilidade->set($dados);
0112     $this->Estudosviabilidade->save();
0113     //gravar historico do envio para contabilidade
0114     $this->Estudosviabilidade->salvaHistorico(0,$projeto['Estudosviabilidade']['projeto_id'],
0115     $someone['Usuario']['id']);
0116     //gerar mensagem para analista contabil
0117     $this->Estudosviabilidade->enviaMensagem(1,$projeto['Estudosviabilidade']['projeto_id'],
0118     $someone['Usuario']['id']);
0119     $this->Session->setFlash('Estudo de Viabilidade Enviado para Contabilidade com sucesso!');
```

Figura 25 - Codificação da função enviarcontabilide da classe Estudosviabilidades

### 3.3.2.1.3 Camada visão

A camada visão que visualiza os dados lidos pela camada de controle e para cada função do controle tem uma tela mostrando os dados correspondentes, com a exceção da função delete que a partir da tela index ele chama a função delete para excluir o registro. Segue abaixo os exemplos de codificação para cada tela.

A Figura 26 apresenta a codificação da tela index.

```

<div class="projetos">
<h2>Projetos</h2>

//cabecalho da tabela
<table cellpadding="0" cellspacing="0">
<tr>
<th>Id</th>
<th>Estado</th>
<th>Titulo</th>
<th>Gerente</th>
<th>Patrocinador</th>
<th>Prazo</th>
<th>Ações</th>
</tr>
//le todos os registros da tabela projeto
<?php foreach ($projetos as $projeto): ?>
<tr>
<td><?php echo $projeto['Projeto']['id']; ?></td>
<td>&nbsp;<?php echo $html->link($projeto['Estado']['ds_estado'],
'/estados/view/' . $projeto['Estado']['id']); ?></td>
<td><?php echo $projeto['Projeto']['ds_titulo']; ?></td>
<td><?php echo $projeto['Projeto']['user_created']; ?></td>
<td>&nbsp;<?php echo $html->link($projeto['Usuario']['nm_usuario'],
'/usuarios/view/' . $projeto['Usuario']['id']); ?></td>
<td><?php echo $html->formata_pt($projeto['Projeto']['dt_prazo']); ?></td>
<td class="actions">
<?php echo $html->link('Detalhar','/projetos/view/' . $projeto['Projeto']['id']); ?>
<?php echo $html->link('Editar','/projetos/edit/' . $projeto['Projeto']['id']); ?>
<?php echo $html->link('Excluir','/projetos/delete/' . $projeto['Projeto']['id'],
null, 'Deseja realmente excluir : id ' . $projeto['Projeto']['id']); ?>
</td>
</tr>
<?php endforeach: ?>
</table>

```

Figura 26 - Codificação da tela index de uma classe

A Figura 27 apresenta a codificação da tela view.

```

<dl>
<dt>Id</dt>
<dd>&nbsp;<?php echo $projeto['Projeto']['id']; ?></dd>
<dt>Estado</dt>
<dd>&nbsp;<?php echo $html->link($projeto['Estado']['ds_estado'],
'/estados/view/' . $projeto['Estado']['id']); ?></dd>
<dt>Titulo</dt>
<dd>&nbsp;<?php echo $projeto['Projeto']['ds_titulo']; ?></dd>
<dt>Gerente</dt>
<dd>&nbsp;<?php echo $projeto['Projeto']['user_created']; ?></dd>
<dt>Patrocinador</dt>
<dd>&nbsp;<?php echo $html->link($projeto['Usuario']['nm_usuario'],
'/usuarios/view/' . $projeto['Usuario']['id']); ?></dd>
<dt>Prazo</dt>
<dd>&nbsp;<?php echo $html->formata_pt($projeto['Projeto']['dt_prazo']); ?></dd>
</dl>
<ul class="actions">
<li><?php echo $html->link('Editar Projeto', '/projetos/edit/' .
$projeto['Projeto']['id']); ?> </li>
<li><?php echo $html->link('Excluir Projeto', '/projetos/delete/' .
$projeto['Projeto']['id'], null, 'Deseja realmente excluir : id ' .
$projeto['Projeto']['id'] . '?' . $projeto['Projeto']['id']); ?> </li>
<li><?php echo $html->link('Listar Projetos', '/projetos/index') ?> </li>
<li><?php echo $html->link('Inserir Projeto', '/projetos/add') ?> </li>
<li><?php echo $html->link('Listar Estado', '/estados/index/') ?> </li>
<li><?php echo $html->link('Inserir Estados', '/estados/add/') ?> </li>
</ul>
</div>
// mostra todas as classes relacionadas com essa classe
<div class="related">
<h3>Anexos relacionados</h3>
<?php if (!empty($projeto['Anexo'])): ?>
<table cellpadding="0" cellspacing="0">
<tr>

```

Figura 27 - Codificação da tela view de uma classe

A Figura 28 apresenta a função `add` de uma classe.

```

<h2>Novo Projeto</h2>
<form action="<?php echo $html->url('/projetos/add'); ?>" method="post">
<div class="required">
  <?php echo $form->labelTag('Projeto/estado_id', 'Estado');?>
  <?php echo $html->selectTag('Projeto/estado_id', $estados,
    $html->tagValue('Projeto/estado_id'), array(), array(), false);?>
  // campos obrigatorios tem a mensagem aqui
  <?php echo $html->tagErrorMsg('Projeto/estado_id', 'Por favor selecione Estado.')}?>
</div>
<div class="required">
  <?php echo $form->labelTag('Projeto/ds_titulo', 'Ds Titulo');?>
  <?php echo $html->input('Projeto/ds_titulo', array('size' => '100'));?>
  // campos obrigatorios tem a mensagem aqui
  <?php echo $html->tagErrorMsg('Projeto/ds_titulo', 'Por favor informe Ds Titulo.')}?>
</div>
<div class="required">
  <?php echo $form->labelTag('Projeto/usuario_id', 'Patrocinador');?>
  <?php echo $html->selectTag('Projeto/usuario_id', $usuarios,
    $html->tagValue('Projeto/usuario_id'), array(), array(), false);?>
  // campos obrigatorios tem a mensagem aqui
  <?php echo $html->tagErrorMsg('Projeto/usuario_id', 'Por favor selecione Usuario.')}?>
</div>
<div class="optional">
  <?php echo $form->labelTag('Projeto/dt_prazo', 'Dt Prazo');?>
  <?php echo $html->dateTimeOptionTag('Projeto/dt_prazo',
    'DMY', 'NONE', $html->tagValue('Projeto/dt_prazo'), array());?>
  // campos obrigatorios tem a mensagem aqui
  <?php echo $html->tagErrorMsg('Projeto/dt_prazo', 'Por favor selecione Dt Prazo.')}?>
</div>
<div class="submit">
  <?php echo $html->submit('Adicionar');?>
</div>
</form>
<ul class="actions">
<li><?php echo $html->link('Listar Projetos', '/projetos/index')?></li>
<li><?php echo $html->link('Detalhes Estados', '/estados/index/');?></li>
<li><?php echo $html->link('Adicionar Estados', '/estados/add/');?></li>

```

Figura 28 - Codificação da função `add` de uma classe

A Figura 29 apresenta a função `edit` de uma classe.

```

<h2>Editar Projeto</h2>
<form action="<?php echo $html->url('/projetos/edit/'. $html->tagValue('Projeto/id')); ?>" method="post">
<div class="required">
  <?php echo $form->labelTag('Projeto/ds_titulo', 'Titulo');?>
  <?php echo $html->input('Projeto/ds_titulo', array('size' => '100'));?>
  <?php echo $html->tagErrorMsg('Projeto/ds_titulo', 'Por favor informe o Titulo.')}?>
</div>
<div class="required">
  <?php echo $form->labelTag('Projeto/usuario_id', 'Patrocinador');?>
  <?php echo $html->selectTag('Projeto/usuario_id', $usuarios,
    $html->tagValue('Projeto/usuario_id'), array(), array(), false);?>
  <?php echo $html->tagErrorMsg('Projeto/usuario_id', 'Por favor selecione Usuario.')}?>
</div>
<div class="optional">
  <?php echo $form->labelTag('Projeto/dt_prazo', 'Prazo');?>
  <?php echo $html->dateTimeOptionTag('Projeto/dt_prazo', 'DMY',
    'NONE', $html->tagValue('Projeto/dt_prazo'), array());?>
  <?php echo $html->tagErrorMsg('Projeto/dt_prazo', 'Por favor selecione Dt Prazo.')}?>
</div>
<?php echo $html->hidden('Projeto/id')?>
<div class="submit">
  <?php echo $html->submit('Salvar');?>
</div>
</form>
<ul class="actions">
<li><?php echo $html->link('Excluir', '/projetos/delete/' .
  $html->tagValue('Projeto/id'), null, 'Deseja realmente excluir : id ' . $html->tagValue('Projeto/id'));?></li>
<li><?php echo $html->link('Listar Projetos', '/projetos/index')?></li>
<li><?php echo $html->link('Detalhar Estados', '/estados/index/');?></li>
<li><?php echo $html->link('Adicionar Estados', '/estados/add/');?></li>
</ul>

```

Figura 29 - Codificação da função `edit` de uma classe

### 3.3.3 Operacionalidade da implementação

Para apresentar a operacionalidade da aplicação são apresentadas telas do sistema, visando apresentar um caso real de utilização do mesmo.

Ao iniciar o sistema, a tela de *login* do SICGP é apresentada ao usuário como apresenta a Figura 30.

Figura 30 - Tela de login do sistema

Ao acessar o sistema, o gerente de projetos vai entrar no mapa para gerente de projetos (Figura 31).

<u>Mapa para Gerente de Projetos</u>
<b>Integração</b>
<a href="#">1 - Projeto</a>
<a href="#">3 - Plano de projeto</a>
<a href="#">2 - Estudo de viabilidade</a>
<a href="#">16 - Controle de mudança</a>
<a href="#">15 - Lição aprendida</a>
<a href="#">14 - Relatório de desempenho</a>
<a href="#">17 - Termo de aceite do projeto</a>
<b>Escopo</b>
<a href="#">6 - Atividade</a>
<a href="#">5 - Recurso</a>
<a href="#">4 - Declaração de escopo</a>
<a href="#">6.1 - Dicionário de EAP</a>
<a href="#">6.2 - Elemento de EAP</a>
<b>Custo</b>
<a href="#">7 - Plano de gerenciamento de custo</a>
<a href="#">9 - Orçamento</a>
<b>Aquisição</b>
<a href="#">8 - Plano de gerenciamento de aquisição</a>
<a href="#">8.1 - Processo de aquisição</a>
<b>Comunicação</b>
<a href="#">11 - Plano de gerenciamento de comunicação</a>
<a href="#">12 - Mapa de comunicação</a>
<a href="#">13 - Ata de reunião</a>
<b>Tempo</b>
<a href="#">10 - Cronograma</a>

Figura 31 - Tela de mapa do gerente de projetos

Ao selecionar a opção “1 – Projeto”, a tela de projetos do gerente é apresentada ao usuário. Para cadastrar um novo projeto, selecionar a opção “Inserir Projeto”.

Na Figura 32 pode ser vista a tela de cadastro de projeto. As principais informações nesse momento são: estado, título, patrocinador para um novo projeto, sem elas não será possível cadastrar um projeto, para finalizar o cadastro selecionar a opção “Adicionar”.

**Novo Projeto**

Estado: iniciado

Título: Sistema de projetos

Patrocinador: Daniel Gielow

Prazo: 2 de Fevereiro de 2008

Adicionar

- [Listar Projetos](#)
- [Listar Estados](#)
- [Adicionar Estado](#)

Figura 32 - Tela de cadastro de projetos do sistema

Após cadastrar um projeto vai voltar para consulta de projetos do gerente, já com o projeto cadastrado. Para continuar com o cadastramento do projeto selecionar a opção “Detalhar”.

As informações que podem ser cadastradas e fazem parte do projeto são:

- anexos;
- atas de reuniões;
- atividades;
- controles de mudança;
- cronogramas;
- estudos de viabilidade;
- históricos;
- lições aprendidas;
- mensagens;
- orçamentos;
- planos de aquisição;
- planos de comunicação;

- m) planos de custo;
- n) planos de projetos;
- o) recursos;
- p) relatórios de desempenho;
- q) termos de aceite de projeto.

Na tela de detalhe do projeto o usuário deve selecionar a opção “Inserir Estudo de viabilidade” para continuar o cadastramento do projeto ou voltar ao menu da Figura 31 e selecionar a opção “Estudo de viabilidade”.

Na Figura 33 pode ser vista a tela de cadastro de estudo de viabilidade. As principais informações nesse momento são: tipo de serviço, projeto, início da validade, fim da validade e texto de encerramento para um novo projeto. Sem elas, não será possível cadastrar um estudo de viabilidade. Para finalizar o cadastro, selecionar a opção “Adicionar”.

**Novo Estudo de viabilidade**

**Tipo de serviço** analise

**Projeto** Projeto Água Verde

**Início da Validade** 4, Março, 2008

**Fim da validade** 5, Março, 2008

**Texto de encerramento**  
projeto destinado a automatizar a área de projetos da empresa.

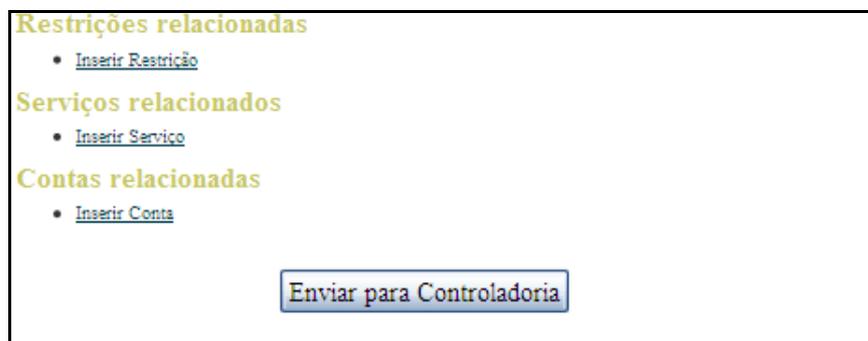
**Usuários relacionados**  
Alexandre Gielow  
Daniel Gielow  
Guilherme Gielow  
Roseane Gielow  
Simone Bugmann  
Wilhelm Gielow

Figura 33 - Tela de cadastro do estudo de viabilidade

Após adicionar o estudo de viabilidade, o sistema vai para a tela de estudos de viabilidade. Selecionar a opção “Detalhar” para inserir mais informações que são:

- a) atividades com os recursos utilizados;
- b) justificativas;
- c) marcos;
- d) objetivos;
- e) produtos;
- f) requisitos;
- g) responsabilidades;
- h) restrições;
- i) serviços;
- j) contas;
- k) usuários.

Após terminar os cadastros de estudo de viabilidade, no final da tela detalhe do estudo de viabilidade, o usuário deve selecionar a opção “Enviar para Controladoria”, conforme na Figura 34.



**Restrições relacionadas**

- [Inserir Restrição](#)

**Serviços relacionados**

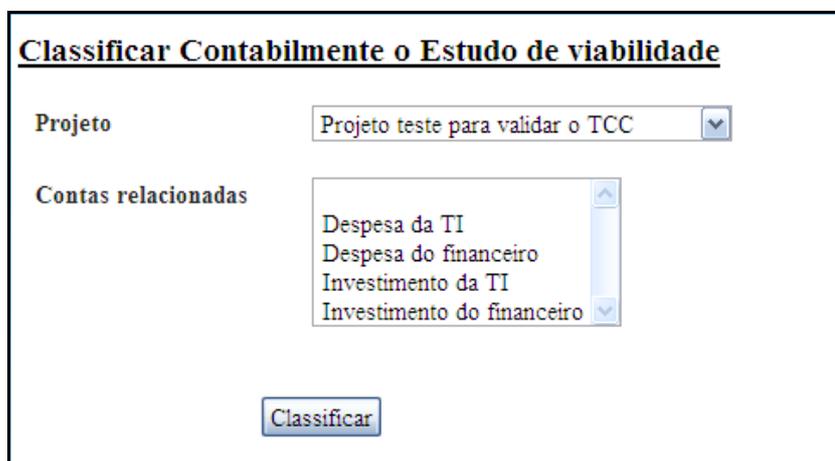
- [Inserir Serviço](#)

**Contas relacionadas**

- [Inserir Conta](#)

Figura 34 – Tela para enviar estudo de viabilidade para contabilidade

O usuário da controladoria vai receber um correio eletrônico, alocar uma ou mais contas para o estudo de viabilidade e selecionar a opção “Classificar”, conforme Figura 35.



**Classificar Contabilmente o Estudo de viabilidade**

Projeto

Contas relacionadas

Figura 35 – Tela para classificar contabilmente o estudo de viabilidade

O gerente de projeto, então, vai enviar ao patrocinador para que ele aprove o projeto (Figura 36).

**Contas relacionadas**

id	vl_conta	ds_contacusto	Ações
1	10215040	Despesa do financeiro	<a href="#">Detalhar</a> <a href="#">Editar</a> <a href="#">Excluir</a>

• [Inserir Conta](#)

Figura 36 – Tela para enviar estudo de viabilidade para patrocinador

O patrocinador vai receber um correio eletrônico como na Figura 37, aprovar o estudo de viabilidade e selecionar a opção “Aprovar Estudo de Viabilidade” (Figura 38).

de SICGPI <guilhermegielow@yahoo.com> [ocultar detalhes](#)  
 responder a SICGPI <guilhermegielow@yahoo.com>  
 para guilhermegielow@gmail.com  
 data 9 de julho de 2008 23:17  
 assunto Mensagem do SICGPI do projeto Trabalho de Conclusão de Curso  
 BCC  
 enviado por yahoo.com  
 assinado por yahoo.com  
 - Ocultar texto das mensagens anteriores -  
 Existe um Projeto para você aprovar favor verificar.

[Responder](#) [Encaminhar](#)

Figura 37 - Tela do correio eletrônico recebido pelo patrocinador

**Contas relacionadas**

id	vl_conta	ds_contacusto
1	10215040	Despesa do financeiro

Figura 38 – Tela para aprovar estudo de viabilidade

Após a aprovação do estudo de viabilidade, o gerente de projetos continua o cadastramento do projeto.

No menu, o usuário seleciona a opção “3 - Plano de Projeto”. Na tela de cadastro de plano de projeto, as principais informações nesse momento são: projeto, plano de projeto e a versão para um novo projeto, sem elas não será possível cadastrar um plano de projeto. Para finalizar o cadastro selecionar a opção “Adicionar”.

No menu, o usuário seleciona a opção “4 - Declaração de escopo”. Na tela de cadastro de declaração de escopo as principais informações nesse momento são: projeto, declarações de escopo para um novo projeto, sem elas não será possível cadastrar uma declaração de escopo, para finalizar o cadastro selecionar a opção “Adicionar”.

Após adicionar a declaração de escopo o sistema vai para a tela de declarações de escopo. Na tela de declarações de escopo selecionar a opção “Detalhar” para inserir mais informações que são:

- a) critérios de aceitação;
- b) não incluídos no escopo;
- c) estratégias de condução;
- d) objetivos do escopo;
- e) premissas;
- f) produtos do escopo;
- g) restrições do escopo;
- h) serviços do escopo.

No menu o usuário seleciona a opção “5 - Recurso”. Na tela de cadastro de recurso as principais informações nesse momento são: projeto, grupo de recurso, unidade e recurso para um novo projeto, sem elas não será possível cadastrar um recurso, para finalizar o cadastro selecionar a opção “Adicionar”.

No menu o usuário seleciona a opção “6 - Atividade”. Na tela de cadastro de atividade as principais informações nesse momento são: projeto, atividade pai, atividade e dias de prazo para um novo projeto, sem elas não será possível cadastrar uma atividade, para finalizar o cadastro selecionar a opção “Adicionar”.

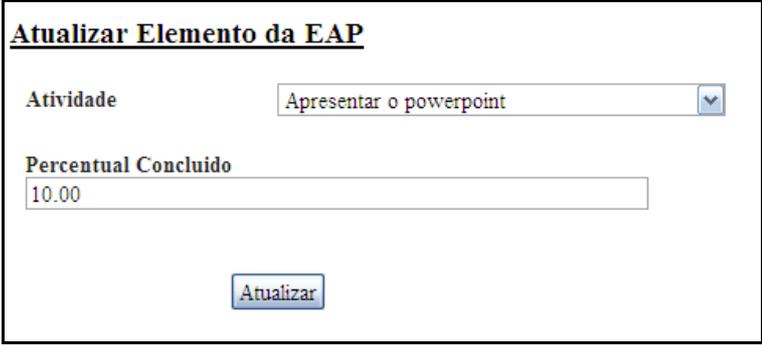
No menu o usuário seleciona a opção “6.1 – Dicionário da EAP”. Na tela de cadastro de dicionário EAP as principais informações nesse momento são: atividade, projeto, tipo de especificação e a descrição da EAP para um novo projeto, sem elas não será possível cadastrar um dicionário da EAP, para finalizar o cadastro selecionar a opção “Adicionar”.

No menu o usuário seleciona a opção “6.2 – Elemento da EAP”. Na tela de cadastro de elemento da EAP as principais informações nesse momento são: atividade, valor planejado, percentual concluído, data planejada, usuário solicitante e usuário responsável para um novo projeto, sem elas não será possível cadastrar um elemento da EAP, para finalizar o cadastro selecionar a opção “Adicionar”.

Após o cadastro do elemento da EAP o analista de projeto vai receber um e-mail avisando que ele tem uma atividade pendente e que deve atualizá-la conforme a Figura 39.

No menu o usuário seleciona a opção “7 – Plano de gerenciamento de custo”. Na tela de cadastro de plano de custo as principais informações nesse momento são: projeto e plano de custo para um novo projeto, sem elas não será possível cadastrar um plano de custo, para finalizar o cadastro selecionar a opção “Adicionar”.

No menu o usuário seleciona a opção “8 – Plano de gerenciamento de aquisição”. Na tela de cadastro de plano de aquisição as principais informações nesse momento são: projeto e plano de aquisição para um novo projeto, sem elas não será possível cadastrar um plano de aquisição, para finalizar o cadastro selecionar a opção “Adicionar”.



A captura de tela mostra uma janela de diálogo intitulada "Atualizar Elemento da EAP". O formulário contém os seguintes campos:

- Atividade: Um menu suspenso com o texto "Apresentar o powerpoint" e uma seta para baixo.
- Percentual Concluído: Um campo de entrada de texto contendo o valor "10.00".
- Botão "Atualizar": Um botão de ação localizado na parte inferior do formulário.

Figura 39 – Tela para atualizar elemento da EAP

Após adicionar plano de aquisição o sistema vai para a tela de planos de aquisição. Na tela de planos de aquisição selecionar a opção “Detalhar” para inserir mais informações que é o processo de aquisição.

No menu o usuário seleciona a opção “8.1 – Processo de aquisição”. Na tela de cadastro de processo de aquisição as principais informações nesse momento são: plano de aquisição, fornecedor, processo de aquisição para um novo projeto, sem elas não será possível cadastrar um processo de aquisição, para finalizar o cadastro selecionar a opção “Adicionar”.

Após o cadastro do processo de aquisição, o gerente vai enviar o “Processo de aquisição” para o patrocinador aprovar. Selecionando a opção “Enviar para Patrocinador” (Figura 40).

**Processo de aquisição**

<b>Id</b>
3
<b>Plano de aquisição</b>
<a href="#">teste de workflow</a>
<b>Fornecedor</b>
<a href="#">Furo</a>
<b>Valor do orçamento</b>
200.00
<b>Dias de prazo</b>
30
<b>Descrição</b>
Compra de um notebook para o gerente de projetos

- [Editar Processo de aquisição](#)
- [Excluir Processo de aquisição](#)
- [Listar Processos de aquisições](#)
- [Inserir Processo de aquisição](#)
- [Listar Planos de aquisições](#)
- [Inserir Plano de aquisição](#)
- [Listar Fornecedores](#)
- [Inserir Fornecedor](#)

Figura 40 - Tela de envio de processo de aquisição para o patrocinador

Neste momento, o patrocinador receberá um correio eletrônico para aprovar o processo de aquisição. Acessa a tela de aprovação e seleciona a opção “Aprovar Processo de Aquisição” (Figura 41).

**Aprovar Processo de aquisição**

<b>Id</b>
3
<b>Plano de aquisição</b>
<a href="#">teste de workflow</a>
<b>Fornecedor</b>
<a href="#">Furo</a>
<b>Valor do orçamento</b>
200.00
<b>Dias de prazo</b>
30
<b>Descrição</b>
Compra de um notebook para o gerente de projetos

Figura 41 - Tela de aprovação do processo de aquisição

No menu o usuário seleciona a opção “9 – Orçamento”. Na tela de cadastro de orçamento as principais informações nesse momento são projeto e valor do projeto para um novo projeto, sem elas não será possível cadastrar um orçamento, para finalizar o cadastro selecionar a opção “Adicionar”.

No menu o usuário seleciona a opção “10 – Cronograma”. Na tela de cadastro de cronograma as principais informações nesse momento são projeto para um novo projeto, sem

ela não será possível cadastrar um cronograma, para finalizar o cadastro selecionar a opção “Adicionar”.

No menu o usuário seleciona a opção “11 – Plano de gerenciamento de comunicação”. Na tela de cadastro de plano de comunicação as principais informações nesse momento são projeto, plano de comunicação e políticas para um novo projeto, sem elas não será possível cadastrar um plano de comunicação, para finalizar o cadastro selecionar a opção “Adicionar”.

No menu o usuário seleciona a opção “12 – Mapa das comunicações”. Na tela de cadastro de mapa de comunicação as principais informações nesse momento são projeto, assunto, tipo de documento, tipo de meio e dias de frequência para um novo projeto, sem elas não será possível cadastrar um mapa de comunicação, para finalizar o cadastro selecionar a opção “Adicionar”.

No menu o usuário seleciona a opção “13 – Ata de Reunião”. Na tela de cadastro de ata de reunião as principais informações nesse momento são tipo de reunião, projeto e ata de reunião para um novo projeto, sem elas não será possível cadastrar uma ata de reunião, para finalizar o cadastro selecionar a opção “Adicionar”.

Após adicionar a ata de reunião o sistema vai para a tela de atas de reuniões. Na tela de atas de reuniões selecionar a opção “Detalhar” para inserir mais informações que são:

- a) ações;
- b) assuntos tratados na reunião;
- c) decisões tomadas na reunião.

No menu o usuário seleciona a opção “14 – Relatório de Desempenho”. Na tela de cadastro de relatório de desempenho as principais informações nesse momento são projeto e relatório de desempenho para um novo projeto, sem elas não será possível cadastrar um relatório de desempenho, para finalizar o cadastro selecionar a opção “Adicionar”.

Após adicionar o relatório de desempenho o sistema vai para a tela de relatórios de desempenho. Na tela de relatórios de desempenho selecionar a opção “Detalhar” para inserir mais informações que são as ações a serem tomadas a partir do resultado do relatório.

No menu o usuário seleciona a opção “15 – Lição Aprendida”. Na tela de cadastro de lição aprendida as principais informações nesse momento são projeto e lição aprendida para um novo projeto, sem elas não será possível cadastrar uma lição aprendida, para finalizar o cadastro selecionar a opção “Adicionar”.

Após adicionar o relatório de desempenho o sistema vai para a tela de relatórios de desempenho. Na tela de relatórios de desempenho selecionar a opção “Detalhar” para inserir mais informações que são as ações a serem tomadas a partir do resultado do relatório.

No menu o usuário seleciona a opção “16 – Controle de Mudança”. Na tela de cadastro de controle de mudança as principais informações nesse momento são projeto, tipo de mudança, prioridade, controle de mudança, justificativo e impacto para um novo projeto, sem elas não será possível cadastrar um controle de mudança, para finalizar o cadastro selecionar a opção “Adicionar” (Figura 42).

Figura 42 - Tela de cadastro de controle de mudança

Após o cadastro do controle de mudança o gerente vai enviar o controle de mudança para o patrocinador aprovar selecionando a opção “Enviar para Patrocinador” conforme a Figura 43.

Figura 43 - Tela para enviar controle de mudança para patrocinador

Neste momento o patrocinador receberá um e-mail para aprovar o controle de mudança, vai acessar a tela de aprovação e vai selecionar a opção “Aprovar Controle de Mudança” conforme na Figura 44.

Data de prazo
00/00/0000
Impacto
asdasd
Data de comunicação
00/00/0000
Data de envio de recursos
00/00/0000

Figura 44 - Tela para aprovar o controle de mudança

No menu o usuário seleciona a opção “17 – Termo de Aceite do Projeto”. Na tela de cadastro de termo de aceite de projeto as principais informações nesse momento são projeto, local e resumo do projeto para um novo projeto, sem elas não será possível cadastrar um termo de aceite de projeto, para finalizar o cadastro selecionar a opção “Adicionar”.

Após o cadastro do termo de aceite de projeto o gerente vai enviar o termo de aceite de projeto para o patrocinador aprovar o projeto selecionando a opção “Enviar para Patrocinador” conforme a Figura 45.

**Termo de aceite de projeto**

Id
3
Projeto
Aprovação do TCC
Data de aceite
00/00/0000
Local
Blumenau
Resumo do projeto
resumo

- [Editar Termo de aceite de projeto](#)
- [Excluir Termo de aceite de projeto](#)
- [Listar Termos de aceite de projetos](#)
- [Inserir Termo de aceite de projeto](#)
- [Listar Projetos](#)
- [Inserir Projeto](#)

**Usuários relacionados**

- [Inserir Usuário](#)

Figura 45 - Tela para enviar termo de aceite de projeto para o patrocinador

Neste momento o patrocinador receberá um correio eletrônico para aprovar o projeto, vai acessar a tela de aprovação e vai selecionar a opção “Aprovar Projeto” conforme na Figura 46.

**Aprovar Projeto**

Id
1
Projeto
<a href="#">Projeto teste para validar o TCC</a>
Data de aceite
00/00/0000
Local
Blumenau
Resumo de projeto
asssst

Figura 46 - Tela para aprovar o projeto

A partir deste momento o projeto será encerrado pelo gerente de projeto e o ciclo do projeto dentro do sistema está completo.

Para medir o custo e o tempo gasto para cada departamento envolvido no projeto, foi desenvolvido um relatório de custos por departamento como apresentado na Figura 47.

[Mapa](#) | Usuário [gzielow](#) logado | [Logout](#)

**Relatório de custo por departamento do projeto testando o fluxo todo**

Departamento	Valor previsto	Tempo previsto
Contabilidade	20.00	30
Diretoria	90.00	80
Financeiro	89.00	10
TI	240.00	138

Sistema de Informações para Controle de Gerenciamento de Projetos 2008  
Desenvolvido por Guilherme Fernando Gielow

Figura 47 - relatório de custos por departamento do projeto

### 3.4 RESULTADOS E DISCUSSÃO

O sistema desenvolvido atendeu as características necessárias reunidas nos trabalhos correlatos. O sistema é web, baseado no PMBOK terceira edição, possui *workflow* para aprovação, utiliza as seis áreas do conhecimento e atende a qualquer área de negócio.

A partir da geração do *script bake* do cake PHP foi rápido e eficiente gerar o CRUD da base de dados do SICGP em três camadas, restando a parte da lógica de negócio, as telas específicas e alguns detalhes de *layout* para aprimorar. A padronização do cake PHP facilita o desenvolvimento de novas telas e novas funcionalidades no sistema, sem muito esforço por parte do desenvolvedor e do analista de sistemas.

O *workflow* tem o importante papel de manter o fluxo contínuo do projeto, informar aos responsáveis o andamento do projeto e delegar atividades para cada membro do projeto.

O ambiente web e a geração do CRUD das telas padronizaram a forma de apresentação dos dados, desta forma, facilitou a utilização do sistema, assim, o usuário terá facilidade para acessar outra tela inicialmente desconhecida.

Ser baseado no PMBOK terceira edição, deu ao sistema o peso das melhores práticas utilizadas na área de gerenciamento de projetos, diminuindo a distância entre a teoria e a prática. No sistema foram utilizadas seis áreas do conhecimento do PMBOK que são:

- a) gerência de custos;
- b) gerência de aquisições;
- c) gerência de comunicações;
- d) gerência de integração;
- e) gerência de tempo;
- f) gerência de escopo.

As características do sistema desenvolvido e dos trabalhos correlatos estão relacionadas no Quadro 18.

Trabalhos	Características				
	ambiente	metodologia	workflow	áreas do conhecimento	área de negócio
SICGP	web	PMBOK 3ª edição	sim	6 áreas	qualquer área
Sistema de gerenciamento de projetos (lplanner)	web	PMBOK 3ª edição	não se aplica	9 áreas	qualquer área
Projeto dotproject	web	PMBOK 3ª edição	sim	9 áreas	qualquer área
Ferramenta de gerenciamento de tempo de projetos	desktop	PMBOK 2ª edição	não se aplica	gerência de tempo	qualquer área
Aplicativo de gerência de projetos	web	própria	sim	não se aplica	informática

Quadro 18 - Comparação entre os trabalhos correlatos e o sistema desenvolvido

Os testes foram realizados no início do desenvolvimento, a cada alteração e após o fim do desenvolvimento do sistema.

Em virtude das limitações do cake PHP, a base de dados foi a parte do sistema que mais sofreu modificações, porém, a fácil utilização do *bake* permitiu que as modificações ganhassem velocidade para alterar ou gerar as novas telas. Nos testes, foi utilizado o *phpmyadmin* para identificar se os dados estavam íntegros e gravados nas tabelas.

O uso do SICGP é útil para controlar e direcionar o trabalho do gerente de projetos e de toda a sua equipe, proporcionando melhores resultados e o envolvimento do analista contábil, do patrocinador, do analista de projeto e principalmente do gerente de projetos.

## 4 CONCLUSÕES

Para um gerenciamento de projeto eficaz não basta somente um gerente e uma equipe competente, mas também um sistema alinhado às necessidades da empresa, como o SICGP, desenvolvido a partir do *framework* cake PHP, que utiliza padrões de projeto que facilitam a customização e o desenvolvimento de novas funcionalidades.

As informações de projeto foram organizadas em áreas de conhecimento, baseados em documentos do PMBOK e estes foram alocados no sistema utilizando o *bake* para gerar os cadastros básicos.

O sistema é web, o acesso é restrito conforme as permissões para cada grupo de usuários e os gerentes visualizam somente as informações de seus projetos.

Para quantificar o custo e o tempo, foi desenvolvido um relatório de custos que verifica os elementos EAPs que foram executados por cada membro do projeto de cada departamento, faz a soma por departamento e traz o valor em reais e o tempo gasto em horas.

O controle do fluxo de gerenciamento de projetos foi atendido pela funcionalidade do *workflow*, através da divisão do projeto em passos e do envio de correio eletrônico para cada passo executado.

O desenvolvimento com a tecnologia do *framework* cake PHP, baseado nas melhores práticas do PMBOK, terceira edição, e as funcionalidades do *workflow* demonstram a relevância do sistema para desenvolvedores web e para gerentes de projeto.

As principais vantagens do sistema são os conceitos de gerenciamento de projetos baseados no PMBOK, que divide o gerenciamento de projetos em áreas de conhecimento e para cada área, atividades bem definidas. A interface do sistema é web e fácil de utilizar, o fluxo é controlado pelo *workflow* e a segurança do sistema é bem definida.

O sistema desenvolvido possui algumas limitações que são: possui somente 6 áreas de conhecimento do PMBOK, foi desenvolvido para gerente de projetos e não gera informação para nenhum outro sistema. Os três itens foram incluídos na seção de extensões para o trabalho.

#### 4.1 EXTENSÕES

Sugere-se como extensão para este trabalho, incluir no SICGP um painel gerencial com relatórios e telas, para o gerente de escritório de projeto ter um controle de todos os projetos do sistema e apresentar relatórios sucintos dos resultados à diretoria da empresa.

Outra modificação que complementa o sistema seria incluir as áreas de conhecimento do PMBOK que não foram contempladas no SICGP que são: gerência de qualidade, gerência de riscos e gerência de recursos humanos, para que possa atender a todas as áreas de negócio.

O SICGP poderia gerar informação para contabilizar os gastos dos projetos em sistemas ERP, fazendo com que os custos de investimento em projetos pudessem ser controlados e analisados no ambiente da empresa como um todo.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ALMEIDA, Rodrigo R. de. **Model-view-controller**. [S.l.], 2007. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/arqu/mvc/mvc.htm>>. Acesso em: 28 jan. 2008.
- ANGULARE. **Desenvolvimento de sites e sistemas web personalizados**. [S.l.], 2007. Disponível em: <<http://www.angulare.com.br/servicos.htm>>. Acesso em: 16 maio 2007.
- ARAUJO, Renata; BORGES, Marcos. **Sistemas de workflow – uso em organizações**. [S.l.], 2002. Disponível em: <<http://chord.nce.ufrj.br/cursos/sctci2004/documentos/WorkflowOrganizacoes.pdf>>. Acesso em: 21 jan. 2008.
- BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 2. ed. Rio de Janeiro: Elsevier, 2007.
- BOENTE, Alfredo. **Gerenciamento & controle de projetos**. Rio de Janeiro: Axcel Books, 2003.
- CAKEPHP BRASIL. **CakePHP framework**. [S.l.], 2007. Disponível em: <<http://cakephp.com.br/modules/news/index.php?storytopic=2>>. Acesso em: 19 mar. 2007.
- CRUZ, Tadeu. **Workflow, a tecnologia que vai revolucionar processos**. 2. ed. São Paulo: Atlas, 2000.
- DALL’OGLIO, Pablo. **PHP: programando com orientação a objetos**. São Paulo: Novatec, 2007.
- DOTPROJECT-BR. **O que é o dotproject?** [S.l.], 2007. Disponível em: <[http://www.dotproject.com.br/index.php?option=com\\_content&task=view&id=12&Itemid=9](http://www.dotproject.com.br/index.php?option=com_content&task=view&id=12&Itemid=9)>. Acesso em: 18 mar. 2007.
- GAMMA, Erich et al. **Padrões de projetos**. São Paulo: Bookman, 2000.
- GOETZ, Carlos H. **Um ambiente para implementação de modelo de gerência de projetos utilizando técnicas de Workflow**. 2003. 91 f. Trabalho de Conclusão de Curso (Superior de Tecnologia de Informática) – Universidade Luterana do Brasil, Canoas.
- GONÇALVES, Edson. **Dominando Ajax: as melhores práticas ligadas a aplicações WEB escritas tanto em Java como em PHP 5 utilizando Ajax**. Rio de Janeiro: Ciência Moderna, 2006.

GONÇALVES, Leandro de C. **Modelagem de dados usando o modelo entidade-relacionamento**. [S.l.], 2008. Disponível em: <<http://www2.dc.ufscar.br/~carvalho/GUBD/aulas/3.html>>. Acesso em: 02 abr. 2008.

IPLANNER. **Sistema de gerenciamento de projetos**. [S.l.], 2007. Disponível em: <<http://www.iplanner.com.br/v2/iplanner/csticas.html>>. Acesso em: 18 mar. 2007.

MARTINS, José C. C. **Gestão de projetos de desenvolvimento de software PMI - UML**. Rio de Janeiro: Brasport, 2002.

MASETO, Jhony M. **Análise avaliativa entre frameworks de PHP**. 2006. 103 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Comunitária Regional de Chapecó, Chapecó.

MELO, Alexandre A. de; MICHELAZZO, Paulino. Padrões de projeto. **PHP magazine**, São Paulo, n.1, p.7-10, 2007.

MENDES, João R. B. **Gerenciamento de projetos: na visão de um gerente de projetos**. São Paulo: Ciência Moderna, 2006.

MINETTO, Elton L. **Frameworks para desenvolvimento em PHP**. São Paulo: Novatec, 2007.

MORO, Mirella M. **Workflow na web**. [S.l.], 2007. Disponível em: <<http://www.inf.ufrgs.br/~mirella/Workflow/work.html>>. Acesso em: 19 mar. 2007.

NASCIMENTO, Mauricio G. F. do; MELO, Alexandre A. de. **PHP profissional: aprenda a desenvolver sistemas profissionais orientados a objetos com padrões de projeto**. São Paulo : Novatec, 2007.

NETO, Almir. **Desenvolvimento em três camadas utilizando MVC e PHP 5**. [S.l.], 2007. Disponível em: <<http://fgsl.aslgo.org.br/fgsl3/images/atividades/mvc-php5.pdf>>. Acesso em: 28 dez. 2007.

POSSI, Marcus. **Gerenciamento de projetos: guia de trabalho**. Rio de Janeiro: Brasport, 2005.

PHPMYADMIN. **PHPmyadmin MySQL administration tool**. [S.l.], 2008. Disponível em: <[http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)>. Acesso em: 13 maio 2008.

PROJECT MANAGEMENT INSTITUTE. **Um guia do conjunto de conhecimentos em gerenciamento de projetos: guia PMBOK**. 3. ed. Newton Square, 2004.

ROMANI, Roberto. **Usabilidade na web**. [S.l.], 2008. Disponível em: <[http://www.revistaw.com.br/index.php?cat\\_id=25&pag\\_id=311&secao=&acao=download](http://www.revistaw.com.br/index.php?cat_id=25&pag_id=311&secao=&acao=download)>. Acesso em: 30 jan. 2008.

SANTANA, Jonh W. S de. **Sistemas workflow: uma aplicação ao IC.** 2006. 27 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Instituto de Computação, Universidade Federal de Alagoas, Maceió.

SOETHE, Fabio. **Ferramenta para gerenciamento de tempo de projetos.** 2004. 58 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

VARGAS, Ricardo V. **Manual prático do plano de projeto: utilizando o PMBOK guide.** Rio de Janeiro: Brasport, 2007.

## APÊNDICE A – Dicionário de dados do SICGP

O sistema SICGP é composto por vinte e uma tabelas principais nas quais todas elas estão relacionadas diretamente com um projeto.

O Quadro 19 apresenta a tabela projetos que trata do projeto.

Tabela projetos					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de projeto	integer		não	PK
estado_id	identificador único de estado	integer		não	FK
usuario_id	identificador único de usuário (patrocinador)	integer		não	FK
ds_titulo	título do projeto	varchar	100	não	
dt_prazo	data de prazo de projeto	date		sim	
dt_encerramento	data de encerramento do projeto	date		sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 19 - Detalhamento da tabela projetos

O Quadro 20 apresenta a tabela licoesaprendidas que trata da lição aprendida no projeto.

Tabela licoesaprendidas					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de lição aprendida	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
ds_licao_aprendida	descrição de lição aprendida	varchar	100	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 20 - Detalhamento da tabela licoesaprendidas

O Quadro 21 apresenta a tabela historicos que trata do histórico do projeto.

Tabela historicos					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de histórico	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
usuario_id	usuário que registrou o histórico	integer		não	FK
ds_historico	descrição do histórico	varchar	2000	sim	

Quadro 21 - Detalhamento da tabela historicos

O Quadro 22 apresenta a tabela atasreunioes que trata da ata de reunião do projeto.

Tabela atasreunioes					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de ata de reunião	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
tiposreuniao_id	identificador único de tipo de reunião	integer		não	FK
ds_ata_reuniao	descrição da ata da reunião	varchar	100	sim	
nr_versão	versão da ata da reunião	float	9,2	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 22 - Detalhamento da tabela atasreunioes

O Quadro 23 apresenta a tabela estudosviabilidades que trata do estudo de viabilidade do projeto.

Tabela estudosviabilidades					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de estudo de viabilidade	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
tiposervico_id	identificador único de tipo de serviço	integer		não	FK
dt_inicio_validade	data de inicio de validade do estudo de viabilidade	date		sim	
dt_fim_validade	data de fim de validade do estudo de viabilidade	date		sim	
ds_texto_encerramento	descrição do texto de encerramento	varchar	2000	sim	
dt_aprovacao	data de aprovação do estudo de viabilidade	date		sim	
user_aprovacao	usuário de aprovação do estudo de viabilidade	date		sim	
log_envio_contabilidade	0= não enviado, 1= enviado para controladoria	bool		sim	
log_classificado	0= não classificado, 1= classificado contabilmente	bool		sim	
log_envio_patrocinador	0= não enviado, 1= enviado para patrocinador	bool		sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 23 - Detalhamento da tabela estudosviabilidades

O Quadro 24 apresenta a tabela cronogramas que trata do cronograma do projeto.

Tabela cronogramas					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de cronograma	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 24 - Detalhamento da tabela cronogramas

O Quadro 25 apresenta a tabela controlesmudancas que trata do controle de mudança do projeto.

Tabela controlesmudancas					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de controle de mudança	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
tiposmudanca_id	identificador único de tipo de mudança	integer		não	FK
ds_controle_mudanca	descrição de controle de mudança	varchar	2000	sim	
ds_justificativa	descrição das justificativas	varchar	2000	sim	
dt_prazo	data de prazo para realizar a mudança	date		sim	
ds_impacto	descrição do impacto no projeto	varchar	2000	sim	
dt_comunicacao	data de comunicação da mudança	date		sim	
dt_envio_recursos	data de envio de recursos	date		sim	
user_aprovacao	usuário de aprovação do controle de mudança	varchar	20	sim	
dt_aprovacao	data de aprovação do controle de mudança	date		sim	
log_envio_patrocinador	0= não enviado, 1=enviado ao patrocinador	bool		sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 25 - Detalhamento da tabela controlemudancas

O Quadro 26 apresenta a tabela termosaceitesprojetos que trata do termo de aceite do projeto.

Tabela termosaceitesprojetos					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de termo de aceite de projeto	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
user_aceite	usuário que aceitou o projeto	varchar	20	sim	
dt_aceite	data de aceite do projeto	datetime		sim	
ds_local	descrição do local do aceite	varchar	100	sim	
ds_resumo_projeto	descrição do resumo do projeto	varchar	2000	sim	
log_envio_patrocinador	0= não enviado, 1= enviado ao patrocinador	bool		sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 26 - Detalhamento da tabela termosaceitesprojetos

O Quadro 27 apresenta a tabela mensagens que trata da mensagem do projeto.

Tabela mensagens					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de mensagem	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
ds_mensagem	descrição da mensagem	text		sim	
dt_mensagem	data da mensagem	datetime		sim	
enviado	0= não enviado, 1= enviado	bool		sim	
nm_envolvidos	nome dos envolvidos na mensagem	text		sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 27 - Detalhamento da tabela mensagens

O Quadro 28 apresenta a tabela passoworkflows que trata do passo do workflow do projeto.

Tabela passoworkflows					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de passo de workflow	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
nr_passo	número do passo do workflow	integer		não	
ds_passo	descrição do passo do workflow	varchar	30	não	
user_responsavel	usuário responsável	varchar	100	não	
mensagem_inicial	mensagem inicial	varchar	2000	não	
mensagem_expirada	mensagem expirada	varchar	2000	não	
nr_passo_aprovado	número do passo após aprovação	integer		não	
dt_ultima_expiracao	data da última expiração	date		sim	
hr_expiracao	quantas horas expiram a próxima vez	integer		não	
dt_termino	data de término do passo do workflow	datetime		sim	

Quadro 28 - Detalhamento da tabela passoworkflows

O Quadro 29 apresenta a tabela relatoriosdesempenhos que trata do relatório de desempenho do projeto.

Tabela relatoriosdesempenhos					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de relatório de desempenho	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
ds_relatorio_desempenho	descrição do relatório de desempenho	varchar	100	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 29 - Detalhamento da tabela relatoriosdesempenhos

O Quadro 30 apresenta a tabela mapascomunicacoes que trata do mapa de comunicação do projeto.

Tabela mapascomunicacoes					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de mapa de comunicação	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
ds_assunto	descrição do assunto	varchar	100	sim	
tp_documento	tipo de documento	varchar	100	sim	
tp_meio	tipo de meio	varchar	100	sim	
dias_frequencia	dias de frequência de envio de mensagem	integer		sim	
ds_acao_esperada	descrição da ação esperada	varchar	100	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 30 - Detalhamento da tabela mapascomunicacoes

O Quadro 31 apresenta a tabela planosprojetos que trata do plano do projeto.

Tabela planosprojetos					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de plano de projeto	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
ds_plano_projeto	descrição de plano de projeto	varchar	2000	sim	
nr_versao	versão do plano de projeto	float	9,2	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 31 - Detalhamento da tabela planosprojetos

O Quadro 32 apresenta a tabela planosquisicoes que trata do plano de aquisição do projeto.

Tabela planosquisicoes					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de plano de aquisição	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
ds_plano_aquisicao	descrição de plano de aquisição	varchar	2000	sim	
ds_documento_referencia	descrição dos documentos de referência de compras	varchar	2000	sim	
ds_avaliacao_fornecedores	descrição dos pontos de avaliação de fornecedores	varchar	2000	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 32 - Detalhamento da tabela planosquisicoes

O Quadro 33 apresenta a tabela planosacomunicacoes que trata do plano de comunicação do projeto.

Tabela planoscomunicacoes					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de plano de comunicação	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
ds_plano_comunicação	descrição de plano de comunicação	varchar	2000	sim	
ds_politicas	descrição das políticas de comunicação do projeto	varchar	2000	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 33 - Detalhamento da tabela planoscomunicacoes

O Quadro 34 apresenta a tabela planosacustos que trata do plano de custo do projeto.

Tabela planosacustos					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de plano de custo	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
ds_plano_custo	descrição de plano de custo	varchar	2000	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 34 - Detalhamento da tabela planosacustos

O Quadro 35 apresenta a tabela orcamentos que trata do orçamento do projeto.

Tabela orcamentos					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de orçamento	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
vl_projeto	valor do projeto	float	9,2	sim	
vl_reserva	valor de reserva do projeto	float	9,2	sim	
vl_administracao	valor de administração do projeto	float	9,2	sim	
vl_total	valor total do projeto	float	9,2	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 35 - Detalhamento da tabela orcamentos

O Quadro 36 apresenta a tabela declaracoesescopos que trata da declaração do escopo do projeto.

Tabela declaracoesescopos					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de declaração de escopo	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
ds_declaracao_escopo	descrição de declaração de escopo	varchar	200	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 36 - Detalhamento da tabela declaracoesescopos

O Quadro 37 apresenta a tabela anexos que trata do anexo do projeto.

Tabela anexos					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de anexo	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
ds_anexo	descrição de anexo	varchar	200	sim	
arq_anexo	nome do arquivo anexado	varchar	200	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 37 - Detalhamento da tabela anexos

O Quadro 38 apresenta a tabela dicionarioseaps que trata do dicionário EAP do projeto.

Tabela dicionarioseaps					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de dicionário EAP	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
atividade_id	identificador único de atividade	integer		não	FK
tipospecificacao_id	identificador único de tipo de especificação	integer		não	FK
ds_descricao_eap	descrição de EAP	varchar	2000	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 38 - Detalhamento da tabela dicionarioseaps

O Quadro 39 apresenta a tabela atividades que trata da atividade do projeto.

Tabela atividades					
Coluna	Descrição	Tipo	Tamanho	Nulo	Chave
id	identificador único de atividade	integer		não	PK
projeto_id	identificador único de projeto	integer		não	FK
tiposatividade_id	identificador único de tipo de atividade	integer		não	FK
ds_atividade	descrição de atividade	varchar	200	sim	
cd_atividade_pai	identificador único de atividade pai	integer		não	
vl_custo	valor da atividade	float	9,2	sim	
dias_prazo	dias de prazo para executar a atividade	integer		sim	
vl_realizado	valor da atividade realizado	float	9,2	sim	
created	dia e hora da criação	datetime		sim	
modified	dia e hora da alteração	datetime		sim	
user_created	usuário da criação	varchar	20	sim	
user_modified	usuário da alteração	varchar	20	sim	

Quadro 39 - Detalhamento da tabela atividades