

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

**SISTEMA WEB DE APOIO AO PROCESSO DE GERÊNCIA
DE PROJETOS BASEADO NA NORMA ABNT ISO/IEC TR
29110**

THIAGO ANDRE PICCININI

**BLUMENAU
2013**

2013/2-23

THIAGO ANDRE PICCININI

**SISTEMA WEB DE APOIO AO PROCESSO DE GERÊNCIA
DE PROJETOS BASEADO NA NORMA ABNT ISO/IEC TR
29110**

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 Sistemas
de Informação — Bacharelado.

Prof. Everaldo Artur Grahl, Mestre - Orientador

**BLUMENAU
2013**

2013/2-23

**SISTEMA WEB DE APOIO AO PROCESSO DE GERÊNCIA
DE PROJETOS BASEADO NA NORMA ABNT ISO/IEC TR
29110**

Por

THIAGO ANDRE PICCININI

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

Presidente: _____
Prof. Everaldo Artur Grahl, Mestre – Orientador, FURB

Membro: _____
Prof. Jacques Robert Heckmann, Mestre – FURB

Membro: _____
Prof. Wilson Pedro Carli, Mestre – FURB

Blumenau, 29 de novembro de 2013.

Dedico este trabalho a todos os amigos,
especialmente aqueles que me ajudaram
diretamente na realização deste.

AGRADECIMENTOS

À Deus, pelo seu imenso amor e graça.

À minha família, Osmar, Denise e Laura, pelo amor, valores ensinados e compartilhados que me tornaram a pessoa que sou hoje.

Ao meu pai, Osmar Piccinini, pelos empurrões e cobranças.

À minha noiva, Aline Coelho, pela compreensão das ausências e força nos momentos difíceis.

Aos colegas, Elton Persike, Ismael Costa, Kauê Raizer e Paulo Danker, pelo apoio, sugestões, contribuições e trocas de experiências, possibilitando a realização deste trabalho.

Ao meu orientador, professor Everaldo Artur Grahl, por ter acreditado na conclusão deste trabalho.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau por suas contribuições durante os semestres letivos.

Seu trabalho vai ocupar uma grande parte da sua vida, e a única maneira de estar verdadeiramente satisfeito é fazendo aquilo que você acredita ser um ótimo trabalho. E a única maneira de fazer um ótimo trabalho é fazendo o que você ama fazer.

Steve Jobs

RESUMO

Este trabalho apresenta um sistema *web* para apoiar o processo de gerência de projetos previsto pela norma ABNT ISO/IEC TR 29110. Este sistema permite que qualquer micro-organização controle seus projetos, auxiliando no controle de entregas, tarefas, recursos, mudanças, riscos, reuniões, entre outros. Apresenta também gráficos para acompanhamento do projeto, baseando-se na data de conclusão de cada entrega e de cada tarefa. Para o desenvolvimento do sistema utilizou-se somente tecnologias *open source*. O *framework* de desenvolvimento *web* Django, é baseado na linguagem de programação Python, e para persistência dos dados usou-se o banco de dados PostgreSQL. Como resultado, obteve-se um sistema de fácil utilização, atendendo as principais tarefas do processos de gerência de projetos previstos pela norma ABNT ISO/IEC TR 29110, sendo que aproximadamente 75% das atividades são atendidas por completo.

Palavras-chave: Gerência de projetos. ABNT ISO/IEC TR 29110. Micro-organizações.

ABSTRACT

This paper presents a web based system to support the project management process provided by ABNT ISO/IEC TR 29110 standard. This system allows any micro organization (very small entities) to control their projects, helping in the delivery control, tasks, resources, changes, risks, meetings, among others. It also shows graphics in order to monitor the project, based on the conclusion date of each delivery and each task. For the development of the system only used *open source* technologies was used. The Django web development framework, is based on the Python programming language. For the data persistence the PostgreSQL database was used. As a result, a system easy to use was created, attending the main tasks of project management provided by ABNT ISO/IEC TR 29110 standard, then approximately 75% of the activities are met in full.

Key words: Project management. ABNT ISO/IEC TR 29110 standard. Micro organizations. Very small entities.

LISTA DE FIGURAS

Figura 1 - Ciclo de vida de um projeto.....	14
Figura 2 – Principais responsabilidades de um gerente de projetos	16
Figura 3- Diagrama do processo de Gerência de Projeto	19
Figura 4 - Fluxo do processo <i>Model Template View</i> utilizado pelo Django	22
Figura 5 - Relação de projetos cadastrados	23
Figura 6 - Cadastro de atividades dentro de um projeto.....	24
Figura 7 - Cadastro de um novo projeto	25
Figura 8 - Diagrama de casos de uso	30
Figura 9 - Modelo de entidades e relacionamentos	31
Figura 10 – Pacotes do projeto	32
Figura 11 - Diagrama de classes do pacote <i>Choices</i>	32
Figura 12 - Diagrama de classes do pacote <i>Models</i>	33
Figura 13 - Diagrama de classes do pacote <i>Admin</i>	34
Figura 14 - Código fonte do arquivo <i>settings.py</i>	36
Figura 15 - Código fonte do arquivo <i>url.py</i>	36
Figura 16 - Tela de autenticação.....	38
Figura 17 - Tela de relação de projetos cadastrados.....	38
Figura 18 - Código fonte para selecionar os projetos na base de dados.....	39
Figura 19 - Código fonte para listar os projetos no <i>template</i>	39
Figura 20 – Tela de cadastro de projeto (Informações básicas)	40
Figura 21 - Tela de cadastro de projeto (Informações específicas).....	40
Figura 22 - Tela de visualização do projeto (Parte superior)	41
Figura 23 - Código fonte para gerar o gráfico entrega	42
Figura 24 - Código fonte para buscar as informações do projeto.....	43
Figura 25 – Tela de relação de entregas	44
Figura 26 - Tela de cadastro de entregas	44
Figura 27 - Tela de relação de usuários vinculados ao projeto	45
Figura 28 - Tela de cadastro de usuário a equipe	45
Figura 29 – <i>E-mail</i> enviado ao usuário cadastrado pelo responsável do projeto	46
Figura 30 - Código fonte para envio de <i>e-mail</i>	46
Figura 31 - Tela de relação de recursos.....	47

Figura 32 - Tela de relação de tarefas.....	47
Figura 33 - Tela de cadastro de tarefas.....	48
Figura 34 - Tela de relação de solicitações de mudanças.....	48
Figura 35 - Tela de relação de riscos.....	49
Figura 36 - Tela de relação de correções.....	49
Figura 37 - Tela de relação de reuniões.....	49
Figura 38 - Tela de relação de anexos.....	50
Figura 39 - Tela de relação de <i>backups</i>	50
Figura 40 - Código fonte para gerar o arquivo de <i>backup</i>	51
Figura 41 - Tela de histórico de modificações do projeto.....	52
Figura 42 - Barra de navegação na tela de cadastro de reunião.....	52

LISTA DE QUADROS

Quadro 1 - Papéis e abreviações.....	19
Quadro 2 - Lista de tarefas da atividade Planejamento do Projeto.....	20
Quadro 3 - Lista de tarefas da atividade Execução do Plano do Projeto.....	20
Quadro 4 - Lista de tarefas da atividade Avaliação e Controle do Projeto	21
Quadro 5 - Lista de tarefas da atividade Encerramento do Projeto.....	21
Quadro 6 - Requisitos funcionais	27
Quadro 7 - Requisitos não funcionais	29
Quadro 8 - Comparativo entre funções do sistema e as tarefas previstas pela norma.....	53
Quadro 9 - Comparativo do sistema desenvolvido com os trabalhos correlatos.....	57
Quadro 10 - Caso de uso Cadastrar Projeto.....	62
Quadro 11 - Caso de uso Registrar Entrega	63
Quadro 12 - Caso de uso Definir Equipe de Trabalho	65
Quadro 13 - Caso de uso Registrar Tarefa	67
Quadro 14 - Caso de uso Cadastrar Recurso	69
Quadro 15 - Caso de uso Cadastrar Risco	71
Quadro 16 - Caso de uso Registrar Reunião	72
Quadro 17 - Caso de uso Registrar Mudança.....	74
Quadro 18 - Caso de uso Cadastrar Correção	75
Quadro 19 - Caso de uso Incluir Documento	77
Quadro 20 - Dicionário de dados da entidade auth_user.....	79
Quadro 21 - Dicionário de dados da entidade project	80
Quadro 22 - Dicionário de dados da entidade project_workteam	80
Quadro 23 - Dicionário de dados da entidade project_delivery	81
Quadro 24 - Dicionário de dados da entidade project_activity	81
Quadro 25 - Dicionário de dados da entidade project_activity_delivery	81
Quadro 26 - Dicionário de dados da entidade project_resource.....	82
Quadro 27 - Dicionário de dados da entidade project_changerequest	82
Quadro 28 - Dicionário de dados da entidade project_changerequest_delivery	83
Quadro 29 - Dicionário de dados da entidade project_correct	83
Quadro 30 - Dicionário de dados da entidade project_risk	83
Quadro 31 - Dicionário de dados da entidade project_meeting	84

Quadro 32 - Dicionário de dados da entidade project_meeting_team.....	84
Quadro 33 - Dicionário de dados da entidade project_file.....	85
Quadro 34 - Dicionário de dados da entidade project_reportcheck	85
Quadro 35 - Dicionário de dados da entidade project_backup.....	86
Quadro 36 - Dicionário de dados da entidade django_admin_log	86

LISTA DE SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

API – *Application Programming Interface*

BSD – *Berkeley Software Distribution*

CSS – *Cascading Style Sheet*

EA – *Enterprise Architect*

FK – *Foreign Key*

FURB – Universidade Regional de Blumenau

GUI – *Graphical User Interface*

GWT – *Google Web Toolkit*

HTML – *HyperText Markup Language*

IEC – *International Electrotechnical Commission*

ISO – *International Organization for Standardization*

JDBC – *Java Database Connectivity*

JSON – *JavaScript Object Notation*

MER – Modelo de Entidade e Relacionamento

MPS.BR – Modelo de Referência de Melhoria de Processo de Software Brasileiro

MTV – *Model Template View*

ODBC – *Open Database Connectivity*

OECD – *Organization for Economic Cooperation and Development*

OO – Orientação a Objetos

PEP – *Python Enhancement Proposals*

PHP – *Hypertext Preprocessor*

PK – *Primary Key*

PMBOK – *Project Management Body of Knowledge*

PME – Pequenas e Médias Empresas

PMI – *Project Management Institute*

RF – Requisito funcional

RNF – Requisito não funcional

SGDBR - Sistema Gerenciador de Banco de Dados

SQL – *Structured Query Language*

TCC – Trabalho de Conclusão de Curso

UC – *Use Case*

URL – *Uniform Resource Locator*

VSE – *Very Small Entities*

ZIP – *Zone Improvement Plan Code*

XML – *eXtensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS DO TRABALHO	13
1.2 ESTRUTURA DO TRABALHO	13
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 GERÊNCIA DE PROJETOS	14
2.2 ABNT ISO/IEC TR 29110	17
2.2.1 Processo de Gerência de Projetos	18
2.3 PYTHON E FRAMEWORK WEB DJANGO.....	22
2.4 TRABALHOS CORRELATOS	23
3 DESENVOLVIMENTO	26
3.1 LEVANTAMENTO DE INFORMAÇÕES	26
3.2 ESPECIFICAÇÃO	27
3.2.1 Requisitos do Sistema	27
3.2.2 Diagrama de Casos de Uso	29
3.2.3 Modelo Entidade Relacionamento	30
3.2.4 Diagrama de Classes	31
3.2.4.1 Diagrama de Classes do Pacote <i>Choices</i>	32
3.2.4.2 Diagrama de Classes do Pacote <i>Model</i>	33
3.2.4.3 Diagrama de Classes do Pacote <i>Admin</i>	34
3.3 IMPLEMENTAÇÃO	35
3.3.1 Técnicas e Ferramentas Utilizadas	35
3.3.1.1 <i>Framework web</i> Django	35
3.3.1.2 PostgreSQL.....	37
3.3.2 Operacionalidade da Implementação.....	37
3.4 RESULTADOS E DISCUSSÃO	53
4 CONCLUSÕES	58
4.1 EXTENSÕES	59
REFERÊNCIAS	60
APÊNDICE A – Descrição dos Casos de Uso	62
APÊNDICE B – Descrição do Dicionário de Dados	79

1 INTRODUÇÃO

A maior parte das Pequenas e Médias Empresas (PMEs) ou micro-organizações enfrenta dificuldade em relacionar normas de qualidade com suas necessidades de negócio, visto que a maioria destas normas não contemplam a sua realidade. Estas PMEs ficam limitadas e não têm meios de serem reconhecidas como entidades que produzem *software* de alta qualidade. Devido a este fato, conseqüentemente são muitas vezes excluídas de alguma atividade econômica. Muitas PMEs não possuem um processo de Gerência de Projetos bem definido. Isto está ligado, muitas vezes a dificuldade para encontrar um *software* de fácil utilização que atenda às suas necessidades principais, pois a maioria dos *softwares* existentes são complexos, visando as grandes organizações. Com isto, surgiu a ideia do desenvolvimento de um sistema *web* simples e prático que atenda as principais tarefas da gerência de projetos visando atender a norma ABNT ISO/IEC TR 29110.

Esta norma é aplicável a micro-organizações (*Very Small Entities – VSEs*), que para efeitos da Associação Brasileira de Normas Técnicas (ABNT) NBR ISO/IEC TR 29110 são empresas, departamentos, organizações ou projetos de até vinte e cinco pessoas. O relatório da *Organization for Economic Cooperation and Development* (OECD) diz que pequenas e médias empresas constituem a forma dominante de organizações em todos os países do mundo, respondendo por mais de 95% e até 99% da população de negócios, dependendo do país (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2012).

Na norma ABNT ISO/IEC TR 29110 dois processos são definidos: Gerência de Projetos (PM) e Implementação de Software (SI). O processo de Gerência de Projetos tem como propósito estabelecer e realizar de forma sistemática as tarefas do processo de Implementação de Software, para que os objetivos do projeto sejam concluídos com a qualidade, tempo e custo esperados. Já o processo de Implementação de Software tem como propósito a execução sistemática das atividades, análises, *design*, construção, integração e testes para produtos novos ou modificados de acordo com os requisitos especificados. Esta norma pode ser utilizável pela VSE para estabelecer processos para implementação de qualquer abordagem ou metodologia de desenvolvimento, como: ágil, evolutivo, incremental, desenvolvimento guiado por teste, de acordo com as necessidades da VSE ou do projeto (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2012).

O foco principal deste trabalho é atender as principais tarefas que dão apoio ao processo de Gerência de Projetos previstas pela norma, onde o responsável pelo projeto terá

total controle sobre o mesmo. Para isto desenvolveu-se um sistema *web* baseado em *software* livre, onde utilizou-se o *framework* de desenvolvimento *web* Django que é baseado na linguagem de programação Python. Para a persistência dos dados utilizou-se o banco de dados PostgreSQL.

1.1 OBJETIVOS DO TRABALHO

O objetivo geral deste trabalho é apresentar o desenvolvimento de um sistema *web* que suporte o processo de Gerência de Projetos definidos pela norma ABNT ISO/IEC TR 29110.

Os objetivos específicos do trabalho são:

- a) permitir o registro das principais atividades previstas no processo de Gerência de Projetos;
- b) atender às principais tarefas previstas para cada atividade do processo de Gerência de Projetos.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está organizado em quatro capítulos, sendo que no primeiro tem-se a introdução, o objetivo geral e os objetivos específicos e como o trabalho será estruturado.

No segundo capítulo apresenta-se a fundamentação teórica com conceitos de gerência de projetos, ABNT ISO/IEC TR 29110 e os trabalhos correlatos.

O terceiro capítulo apresenta o desenvolvimento do sistema iniciando-se com o levantamento de informações, tendo na sequência a elaboração dos principais diagramas para auxiliar na compreensão do sistema, bem como as técnicas e ferramentas utilizadas, a operacionalidade do sistema e os resultados e discussões.

No quarto capítulo têm-se as conclusões deste trabalho bem como são apresentadas as sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

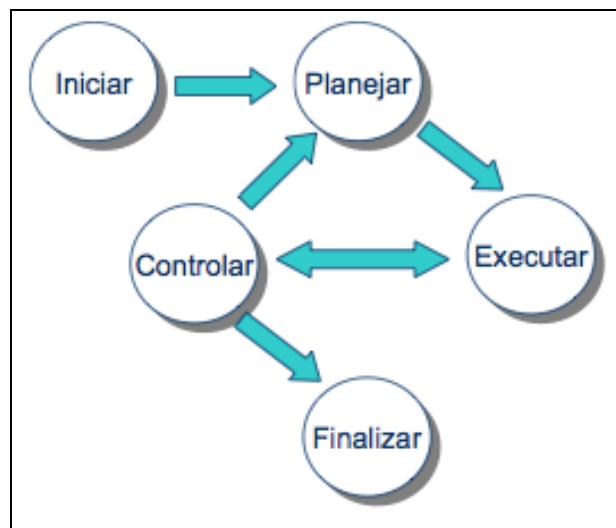
Neste capítulo serão abordados os principais assuntos que auxiliarão no entendimento deste trabalho: Gerência de Projetos, ABNT ISO/IEC TR 29110 e Python e *Framework web Django*, assim como os trabalhos correlatos.

2.1 GERÊNCIA DE PROJETOS

Para entender melhor o que é Gerência de Projetos é necessário entender primeiramente o que é um projeto. Um projeto é algo que é único de alguma forma, tem um fim bem definido e um objetivo claro (que quando atingido caracteriza o fim do projeto). O *Project Management Institute* (PMI) diz que um projeto é um conjunto de atividades temporárias, realizadas em grupo, destinado a produzir um produto, serviço ou resultado único (PROJECT MANAGEMENT INSTITUTE, 2013a).

Analisando este conceito, pode-se concluir que um projeto é tudo que precisa ser planejado para que seja executado, seja ele organizar um casamento, uma festa de aniversário, uma alteração na infraestrutura ou a criação de um sistema. Porém, um projeto somente pode ser caracterizado como projeto quando o mesmo possui um final, ou seja, possui um ciclo bem definido e um ponto final. A Figura 1 ilustra o ciclo de vida de um projeto.

Figura 1 - Ciclo de vida de um projeto



Fonte: Pietro (2002).

Segundo FreeBits (2010) o ciclo de vida de um projeto consiste na divisão do projeto em fases que oferece uma estrutura básica para um bom gerenciamento, garantindo a consistência do mesmo. Para os padrões do PMI o ciclo de vida de um projeto está dividido em cinco fases. São elas:

- a) iniciar: são os processos realizados para definir um novo projeto ou uma nova fase de um projeto existente através da obtenção de autorização para iniciar o projeto ou fase ;
- b) planejar: definir o escopo do projeto, refinar os objetivos, desenvolver o curso de ação necessária para alcançar os objetivos do projeto;
- c) controlar: acompanhar, revisar, regular o progresso e o desempenho do projeto, identificar todas as áreas nas quais serão necessárias mudanças no plano e iniciar a mudanças correspondentes;
- d) executar: executar o trabalho definido no plano de gerenciamento do projeto;
- e) finalizar: finalizar as atividades de todos os grupos de processos visando encerrar formalmente o projeto ou fase.

Para que um projeto seja bem executado é necessário que se tenha um plano de projeto bem definido e documentado, assim como um responsável por gerenciar as etapas definidas no plano do projeto, denominado Gerente de Projetos. O plano de projeto nada mais é que o detalhamento de como as metas do projeto serão cumpridas dentro das limitações impostas.

Planejar um projeto não é prever ou predizer o futuro, é preparar-se para um futuro conhecido. Predizer ou prever o futuro é difícil, mas temos que prever algum futuro (mesmo que incerto). Sem planejamento não é possível estabelecer um cronograma e fazer medidas, sem cronogramas e medidas não há controle, sem controle não há possibilidade de gerência, sem gerência não há produtividade, sem produtividade os custos serão muito altos e os prazos dificilmente serão cumpridos. O fato de existir um planejamento não garante um bom projeto, mas a ausência do planejamento com certeza trará inúmeros problemas para o projeto (ORTH; PRIKLADNICK, 2009). O plano do projeto é o produto final da fase de planejamento e é um documento que deve ser formalmente aprovado e utilizado para gerenciar a execução do projeto.

O Gerente de Projetos é o responsável direto pelo gerenciamento do projeto, que tem como responsabilidade planejar e controlar a execução do projeto. A Figura 2 demonstra as principais responsabilidades de um gerente de projetos.

Figura 2 – Principais responsabilidades de um gerente de projetos



Fonte: Freebits (2010).

O Gerente de Projetos deve ser escolhido levando em conta diversos critérios, pois um erro de critério na seleção pode ocasionar um problema sem volta.

Gerente de Projetos são agentes de mudanças: eles próprios estabelecem os objetivos de um projeto e usam suas habilidades e competências para inspirar um sentimento de propósito compartilhado dentro da equipe do projeto. Eles tiram proveito da adrenalina organizada de novos desafios e a responsabilidade de conduzir a resultados (PROJECT MANAGEMENT INSTITUTE, 2013b, grifo do autor).

O grande desafio da escolha deste profissional é escolher uma pessoa que tenha experiência na gestão de pessoas e pouco conhecimento técnico ou uma pessoa com conhecimento técnico e pouca experiência na gestão de pessoas. O *Project Management Body of Knowledge* (PMBOK) diz que os aspectos interpessoais tem mais relevância, pois três fatores interdisciplinares são definidos: conhecimento e prática na área de gerência de projetos, conhecimento de práticas ligadas a área do projeto e conhecimento e experiência em práticas de gestão geral (ORTH; PRIKLADNICKI, 2009).

Segundo FreeBits (2010) as principais características de um Gerente de Projetos eficaz são:

- a) experiência e conhecimento coerentes com a necessidade do projeto;
- b) experiência estratégica e liderança;
- c) especialista na área do projeto para tomar boas decisões;

- d) competência interpessoal e habilidade com pessoas;
- e) habilidade gerencial comprovada em termos histórico de realizações;
- f) identificar as necessidades do projeto;
- g) estabelecer os objetivos claramente;
- h) atender as expectativas de todas as partes, os *stakeholders*;
- i) tripla restrição: Saber balancear o tempo, o escopo e o custo.

O gerenciamento é uma situação sistemática, mas ao mesmo tempo flexível e ágil para tratar todos os aspectos relevantes à realização de um empreendimento. O gerenciamento é necessário durante todo o ciclo de um projeto, desde sua composição até a sua avaliação final, pois é o responsável pelo planejamento, organização, coordenação, monitoramento e controle durante o ciclo de vida do projeto (PFEIFFER; MAGALHÃES, 2005). Na gerência não se deve apenas ter preocupações em atingir os objetivos, deve-se também facilitar o processo para satisfazer as necessidades e expectativas das pessoas envolvidas ou afetadas pelo projeto.

2.2 ABNT ISO/IEC TR 29110

A norma ABNT ISO/IEC TR 29110 foi lançada em março de 2012 pela Associação Brasileira de Normas Técnicas (ABNT) a fim de definir um guia de gestão e engenharia para micro-organizações (*Very Small Entities – VSEs*), que são entidades (empresas, organizações, departamento ou projeto) de até vinte e cinco pessoas, porém pode ser utilizada por organizações maiores que VSEs, através do processo de Gerência de Projetos (PM) e Implementação de Software (SI). Segundo a Associação Brasileira de Normas Técnicas (2012) a VSE que utilizar esta norma poderá obter os seguintes benefícios:

- a) um conjunto acordado de requisitos do projeto e produtos esperados é entregue ao cliente;
- b) um processo de gerenciamento disciplinado que fornece visibilidade do projeto e ações corretivas para os problemas e desvios do projeto é executado;
- c) um processo de implementação de *software* sistemático que satisfaz as necessidades do cliente e que garantem produtos de qualidade é seguido.

O objetivo principal desta norma é ser utilizada pela VSE para estabelecer processos para implementar qualquer abordagem ou metodologia de desenvolvimento, como: ágil, evolutivo, incremental, desenvolvimento guiado por testes, com base nas suas necessidades.

Para isto, dois processos são definidos, o de Gerência de Projetos (PM) e o de Implementação de Software (SI), sendo que o foco deste trabalho será no processo de Gerência de Projetos (PM) que será detalhado na seção 2.2.1.

Esta norma também é aderente ao Modelo de Referência de Melhoria de Processo de Software Brasileiro (MPS.BR), onde o modelo de vida da VSE é aderente, em partes, ao nível G do MPS.BR (ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARES BRASILEIRO, 2012).

2.2.1 Processo de Gerência de Projetos

Segundo a Associação Brasileira de Normas Técnicas (2012) o processo de gerência de projetos tem o propósito de estabelecer e realizar de forma sistemática as tarefas do processo de Implementação de Software (SI), para que os objetivos sejam cumpridos na qualidade, tempo e custo esperados. Este processo possui sete objetivos principais para que seja executado com o êxito. São eles:

- a) desenvolver o plano de projeto de acordo com a declaração de trabalho revisada e aceita pelo cliente;
- b) registrar e monitorar o progresso do projeto de acordo com o plano de projeto;
- c) tratar as solicitações de mudança, avaliando-as de acordo com o seu impacto no custo, prazo;
- d) realização de reuniões com a equipe de trabalho e o cliente;
- e) identificar os riscos durante o desenvolvimento do projeto;
- f) estabelecer uma estratégia de controle de versão, identificando e definindo os itens de configuração de *software* e controlando as modificações e liberações dos itens que são entregues ao cliente;
- g) garantir que a qualidade do *software* cumpra os requisitos do plano do projeto.

Este processo é dividido em quatro atividades bem definidas que serão descritas a seguir, mas para o melhor entendimento é necessário ter o conhecimento dos papéis e abreviações deste processo, conforme demonstrado no Quadro 1.

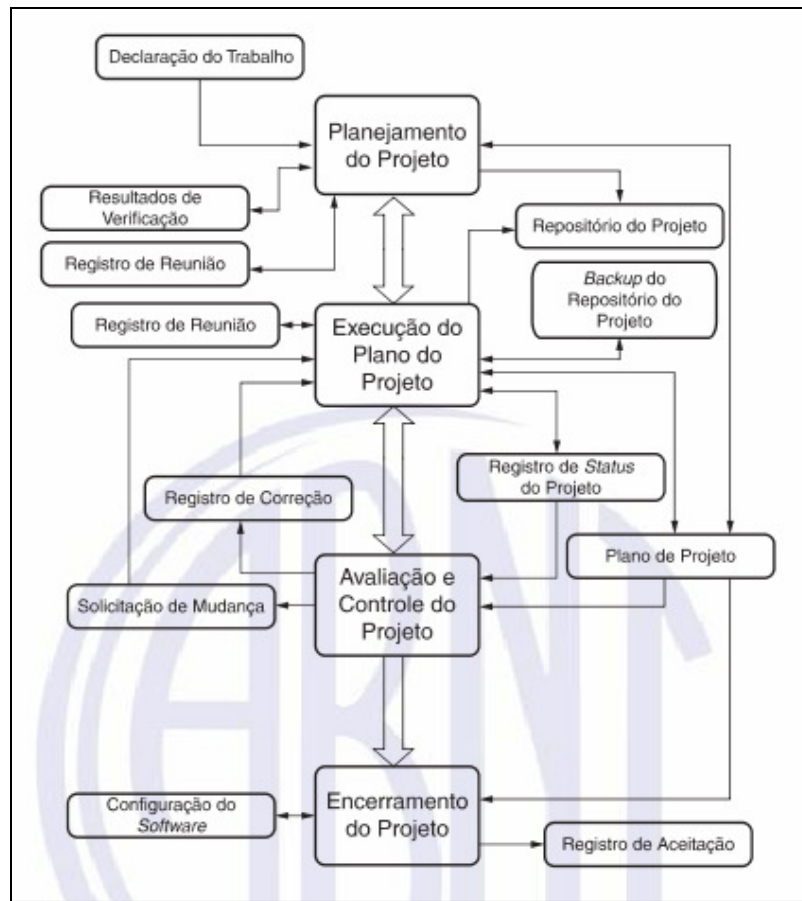
Quadro 1 - Papéis e abreviações

Papel	Abreviação
Cliente	CUS
Gerente de Projeto	PM
Líder Técnico	TL
Equipe de Trabalho	WT

Fonte: Associação Brasileira de Normas Técnicas (2012).

A Figura 3 demonstra o diagrama do processo de Gerência de Projetos (PM).

Figura 3- Diagrama do processo de Gerência de Projeto



Fonte: Associação Brasileira de Normas Técnicas (2012).

A atividade Planejamento de Projeto documenta os detalhes necessários para gerenciar o projeto onde a declaração do trabalho é revisada, o ciclo do projeto é definido, a estratégia da garantia de qualidade é definida, as funções da Equipe de Trabalho (WT) são definidas, os recursos necessários são alocados e os riscos são identificados. O Quadro 2 demonstra a lista de tarefas previstas nesta atividade, assim como os envolvidos em cada uma.

Quadro 2 - Lista de tarefas da atividade Planejamento do Projeto

Descrição	Papel
Revisar a declaração de trabalho.	PM / TL
Definir com o cliente as instruções de entrega.	PM / CUS
Identificar as tarefas específicas a serem realizadas para produzir os entregáveis.	PM / TL
Estabelecer a duração estimada para executar cada tarefa.	PM / TL
Identificar e documentar os recursos.	PM / TL
Estabelecer a composição da equipe de trabalho.	PM / TL
Atribuir datas de início e de conclusão estimadas para cada uma das tarefas.	PM / TL
Calcular e documentar a estimativa de custos e esforço do projeto.	PM
Documentar a estratégia de controle de versão.	PM / TL
Gerar o plano do projeto.	PM
Incluir a descrição do produto, escopo, objetivo e entregáveis.	PM / TL
Verificar e obter aprovação do plano do projeto.	PM / TL
Revisar e aceitar o plano do projeto.	PM CUS
Estabelecer o repositório de projeto.	PM / TL

Fonte: Associação Brasileira de Normas Técnicas (2012).

A atividade Execução do Plano de Projeto é responsável por implementar o plano documentado no projeto, onde os registros de *status* do progresso são atualizados, as solicitações de mudanças são avaliadas e as aprovadas são adicionadas no plano do projeto, a revisão e a aceitação com a Equipe de Trabalho (WT) e o Cliente (CUS) é realizada. O Quadro 3 demonstra a lista de tarefas previstas nesta atividade, assim como os envolvidos em cada uma.

Quadro 3 - Lista de tarefas da atividade Execução do Plano do Projeto

Descrição	Papel
Monitorar a execução do plano do projeto e registrar o status de progresso.	PM / TL / WT
Analisar e avaliar a solicitação de mudança quanto ao impacto de custo, de prazo e técnico.	PM / TL
Conduzir reuniões de revisão com a equipe de trabalho, identificar	PM / TL / WT

problemas, rever status dos riscos e registrar as decisões.	
Fazer backups de acordo com a estratégia de controle de versão.	PM
Fazer a recuperação do repositório do projeto.	PM

Fonte: Associação Brasileira de Normas Técnicas (2012).

A atividade Avaliação e Controle do Projeto é responsável por avaliar o desempenho do plano do projeto de acordo com os compromissos documentados, onde a avaliação da realização do plano e progresso é feita. Os desvios e problemas são identificados e avaliados, os riscos são revisados e novos são identificados e as solicitações de mudanças são documentadas e a ação corretiva é definida. O Quadro 4 demonstra a lista de tarefas previstas nesta atividade, assim como os envolvidos em cada uma.

Quadro 4 - Lista de tarefas da atividade Avaliação e Controle do Projeto

Descrição	Papel
Avaliar o progresso do projeto com respeito ao plano do projeto.	PM / TL / WT
Estabelecer ações para corrigir problemas de desvios e riscos identificados.	PM / TL / WT
Identificar mudanças nos requisitos e/ou plano de projeto para enfrentar desvios significativos.	PM / TL / WT

Fonte: Associação Brasileira de Normas Técnicas (2012).

Por último, a atividade de Encerramento do Projeto é responsável por fornecer o produto e documentar o projeto de acordo com os requisitos do contrato, onde o produto especificado é liberado, o suporte ao produto é definido e o registro de aceitação é assinado. O Quadro 5 demonstra a lista de tarefas previstas nesta atividade, assim como os envolvidos em cada uma.

Quadro 5 - Lista de tarefas da atividade Encerramento do Projeto

Descrição	Papel
Formalizar a conclusão do projeto de acordo com as instruções de entrega estabelecidas.	PM / CUS
Atualizar o repositório do projeto.	PM

Fonte: Associação Brasileira de Normas Técnicas (2012).

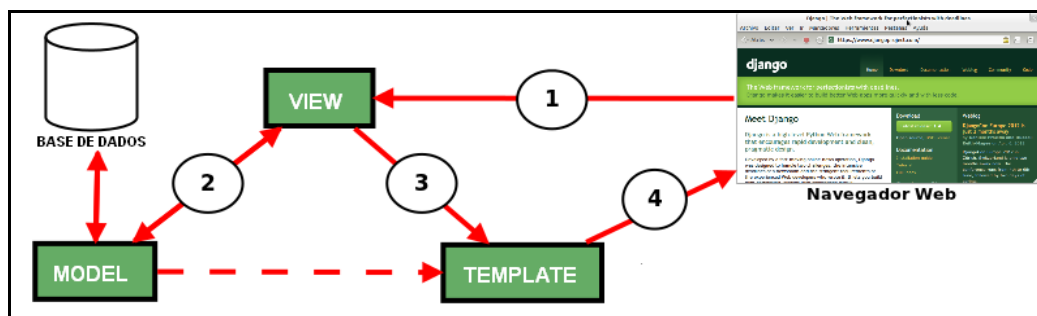
2.3 PYTHON E FRAMEWORK WEB DJANGO

Python teve seu desenvolvimento iniciado em 1989 pelo holandês Guido Van Rossum, que queria criar uma linguagem de altíssimo nível, que agregasse características importantes de diversas linguagens de programação, mas com uma sintaxe simples e clara (SANTANA NETO; GALES, 2010). É uma linguagem de programação que está sob licença *open source* que é administrada pela Python Software Foundation. Python atualmente é uma das mais populares linguagens dinâmicas de programação. Hoje, é usada para tudo, desde *scripts* simples de uso único a grandes servidores *web* que proveem serviço ininterrupto. É usada também em *Graphical User Interface* (GUI) e programação para banco de dados, programação *web* tanto no lado do cliente quanto no servidor (ROSSUM, 2009). Segundo a Python (2013) as principais características da linguagem são:

- a) *syntaxe* clara e legível;
- b) orientação a objetos intuitiva;
- c) modularidade completa, suportando pacotes hierárquicos;
- d) tratamento de erros baseado em exceção;
- e) tipos de dados dinâmicos.

Django é um *framework* de desenvolvimento *web*, escrito em Python, criado inicialmente para criação da versão *web* de jornais do grupo “The Word Company”. Em 2005 foi liberado sob a licença *Berkeley Software Distribution* (BSD). Segundo o DjangoProject (2013), Django é um *framework web* de alto nível, que estimula o desenvolvimento rápido e *design clean*, pragmático. O *framework* utiliza o paradigma *Model Template View* (MTV) para desenvolvimento de suas aplicações. Este padrão não é obrigatório, mas é extremamente recomendado pelo DjangoProject (SANTANA NETO; GALES, 2010). A Figura 4 apresenta o fluxo do processo do MTV utilizado pelo Django.

Figura 4 - Fluxo do processo *Model Template View* utilizado pelo Django



Fonte: MaestrosDelWeb (2012).

O paradigma *Model Template View* (MTV) basicamente modulariza o código fonte e separa a regra de negócio da interface com o usuário. Segundo MasetrosDelWeb (2012) a divisão é feita da seguinte forma:

- a) o *model* define os dados armazenados na forma de classe Python. Cada tipo de dados a serem armazenados é uma variável com diversos parâmetros;
- b) a *view* determina quais dados serão exibidos. Não tem a ver com o estilo de apresentação dos dados. Também pode ser utilizada para outros fins, como: envio de *e-mail*, serviços de autenticação e validação de dados externos por meio de formulários;
- c) o *template* é basicamente uma página *HyperText Markup Language* (HTML) que aceita etiquetas (*tags*) específicas do *framework* Django.

2.4 TRABALHOS CORRELATOS

Foram estudados três trabalhos de conclusão de curso desenvolvidos na Universidade Regional de Blumenau (FURB).

O trabalho de conclusão de curso (TCC) de Robson Ricardo Giacomozzi implementa uma ferramenta que fornece apoio a empresas que utilizam metodologias ágeis de desenvolvimento, utilizando o método Kanban. Na Figura 5 é possível visualizar a tela de cadastro das atividades do projeto desenvolvida por Giacomozzi. A aplicação é *web* e foi desenvolvida utilizando o *framework* CakePHP (GIACOMOZZI, 2011).

Figura 5 - Relação de projetos cadastrados



The screenshot shows a web application interface for 'GAPS gestão ágil de projetos de software'. The user is logged in as 'Robson Ricardo'. The main content area displays a list of projects under the heading 'Projetos'. The list has columns for 'Título' and 'AÇÕES'. There are three project entries, each with a 'times de trabalho' link and 'editar' and 'deletar' buttons.

Título	AÇÕES
Gestão de avaliação de RH	[times de trabalho] [editar] [deletar]
Software para ensino à distância	[times de trabalho] [editar] [deletar]
Aplicação de relatórios	[times de trabalho] [editar] [deletar]

Fonte: Giacomozzi (2011).

O trabalho de conclusão de curso (TCC) de Taynara Bittelbrunn implementa uma ferramenta de gerenciamento de projetos baseado no PMBOK, onde foram abordados os seguintes tópicos: Gerenciamento de Prazos, Gerenciamento de Custos e Gerenciamento de Escopo. Na Figura 6 é possível visualizar a tela de cadastro das atividades do projeto desenvolvida por Bittelbrunn. A aplicação é *web* e foi desenvolvida utilizando o *framework* GTW-Ext (BITTELBRUNN, 2009).

Figura 6 - Cadastro de atividades dentro de um projeto

Selecionar Participantes		
<input type="checkbox"/>	Nome	Departamento
<input type="checkbox"/>	Administrador	Administrativo
<input type="checkbox"/>	Gil Correa Jr	Gerência de Projetos
<input checked="" type="checkbox"/>	Juliana Sílveira	Comercial
<input type="checkbox"/>	Julio Cavalheiro	Consultoria
<input checked="" type="checkbox"/>	Taynara Bittelbrunn	Consultoria
<input checked="" type="checkbox"/>	Wendel David Przygoda	Desenvolvimento

Fonte: Bittelbrunn (2009).

O trabalho de conclusão de curso (TCC) de Guilherme Fernando Gielow implementa uma ferramenta para controle de gerenciamento de projetos baseado no PMBOK, desenvolvido para *web* e teve seu fluxo controlado por *workflow*. Na Figura 7 é possível visualizar a tela de cadastro de projeto desenvolvida por Gielow. A aplicação é *web* e foi desenvolvida utilizando o *framework* CakePHP (GIELOW, 2008).

Figura 7 - Cadastro de um novo projeto



Novo Projeto

Estado: iniciado ▼

Título: Sistema de projetos

Patrocinador: Daniel Gielow ▼

Prazo: 2 ▼ Fevereiro ▼ 2008 ▼

Adicionar

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

Fonte: Gielow (2008).

A principal diferença entre os trabalhos correlatos e o trabalho desenvolvido é que os correlatos foram baseados em conceitos tradicionais de gerência de projetos como o PMBOK e metodologias ágeis. Já o trabalho desenvolvido está focado na norma ABNT ISO/IEC TR 29110 recém lançada, visando as micro-organizações.

3 DESENVOLVIMENTO

Neste capítulo estão descritas as particularidades do sistema desenvolvido, tais como a descrição do levantamento de informações, a apresentação dos requisitos funcionais e não funcionais, a especificação do diagrama de casos de uso, o diagrama de entidade e relacionamento, o diagrama de classes, a implementação, as técnicas e ferramentas utilizadas, a operacionalidade da implementação e os resultados e discussões.

3.1 LEVANTAMENTO DE INFORMAÇÕES

Desenvolveu-se um sistema *web* que permite ao responsável pelos projetos ter uma ampla visão de todas as informações referentes aos seus projetos previstos pela norma ABNT ISO/IEC TR 29110. O sistema apresenta ao usuário uma relação de projetos cadastrados por ele e uma relação de projetos a que ele está vinculado, permitindo que o usuário visualize detalhadamente cada projeto. Neste detalhamento o usuário terá uma visão completa de todos os tópicos do projeto e poderá fazer qualquer alteração, caso o mesmo seja o responsável pelo projeto. Caso for um projeto que o usuário está vinculado, o papel do mesmo no projeto dirá o que o usuário poderá fazer.

O controle de acesso ao sistema é por endereço de correio eletrônico (*e-mail*) e senha, onde os usuários podem cadastrar-se e utilizar todas as funcionalidades do sistema. O responsável pelo projeto poderá vincular ao projeto usuários que ainda não estão cadastrados. Neste caso ele informará os dados do novo usuário e um *e-mail* será encaminhado ao mesmo, indicando que foi cadastrado por um determinado usuário e vinculado ao determinado projeto.

Para a construção do sistema, utilizaram-se as seguintes ferramentas de desenvolvimento:

- a) Django, *framework* para desenvolvimento *web* baseado na linguagem de programação Python;
- b) Bootstrap, para customização do *layout* de telas;
- c) Google *Charts*, para geração de gráficos;
- d) PostgreSQL, para armazenamento de todas as informações;
- e) *Enterprise Architect* (EA), para visualização e criação do diagrama de casos de uso

e diagrama de classes;

- f) *MySQL Workbench*, para visualização das entidades e criação do Modelo de Entidade e Relacionamento (MER).

3.2 ESPECIFICAÇÃO

Nesta seção é apresentado os requisitos funcionais e não funcionais do sistema desenvolvido, bem como o diagrama de casos de uso, modelo de entidade e relacionamento e os diagramas de classes.

3.2.1 Requisitos do Sistema

Nesta subseção são apresentados os requisitos funcionais e não funcionais do sistema desenvolvido.

O Quadro 6 apresenta os requisitos funcionais e sua rastreabilidade, ou seja, a ligação com os casos de uso associados.

Quadro 6 - Requisitos funcionais

Requisitos Funcionais	Caso de Uso
RF01: O sistema deve permitir ao gerente de projeto o cadastro de projeto.	UC01
RF02: O sistema deve permitir ao gerente de projeto o registro de entrega.	UC02
RF03: O sistema deve permitir ao gerente de projeto o cadastro da data estimada para entrega da entrega.	UC02
RF04: O sistema deve permitir ao gerente de projeto e ao líder técnico o registro de tarefas.	UC04
RF05: O sistema deve permitir ao gerente de projeto e ao líder técnico o cadastro da duração estimada para executar a tarefa.	UC04
RF06: O sistema deve permitir ao gerente de projeto e ao líder técnico o cadastro da data de início e fim estimada para a conclusão da tarefa.	UC04

RF07: O sistema deve apresentar ao usuário o plano do projeto.	UC11
RF08: O sistema deve permitir ao gerente de projeto e ao líder técnico o registro de solicitações de mudanças.	UC08
RF09: O sistema deve permitir ao gerente de projeto a definição da equipe de trabalho.	UC03
RF10: O sistema deve permitir ao gerente de projeto e ao líder técnico o cadastro de recursos.	UC05
RF11: O sistema deve permitir ao gerente de projeto e a equipe de trabalho o cadastro de riscos.	UC06
RF12: O sistema deve permitir ao gerente de projeto e a equipe de trabalho o cadastro de correções.	UC09
RF13: O sistema deve permitir ao gerente de projeto e ao líder técnico o registro de reuniões.	UC07
RF14: O sistema deve permitir ao usuário a inclusão de documentos.	UC10
RF15: O sistema deve apresentar ao usuário o <i>status</i> de progresso do projeto.	UC12
RF16: O sistema deve permitir ao gerente de projeto a formalização da conclusão do projeto.	UC13
RF17: O sistema deve permitir ao gerente de projeto o cadastro de usuários a equipe de trabalho a partir do registro de tarefas.	UC03
RF18: O sistema deve permitir ao gerente de projeto o cadastro de entregas a partir do registro de tarefas.	UC02
RF19: O sistema deve permitir ao gerente de projeto o cadastro de usuários a partir do cadastro de equipe de trabalho.	UC03
RF20: O sistema deve permitir ao usuário a visualização do histórico de mudanças.	UC11
RF21: O sistema deve permitir ao gerente de projeto e ao líder técnico o registro do resultado de verificação do projeto.	UC14
RF22: O sistema deve permitir ao gerente de projeto a geração de arquivos de <i>backup</i> .	UC15
RF23: O sistema deve permitir ao gerente de projeto o registro de entregas a partir do registro de solicitações de mudanças.	UC02
RF24: O sistema deve permitir ao gerente de projeto o cadastro de	UC03

usuários a equipe de trabalho a partir do cadastro de correções.	
RF25: O sistema deve permitir ao gerente de projeto o cadastro de usuários a equipe de trabalho a partir do registro de reuniões.	UC03

O Quadro 7 apresenta os requisitos não funcionais previstos para o sistema.

Quadro 7 - Requisitos não funcionais

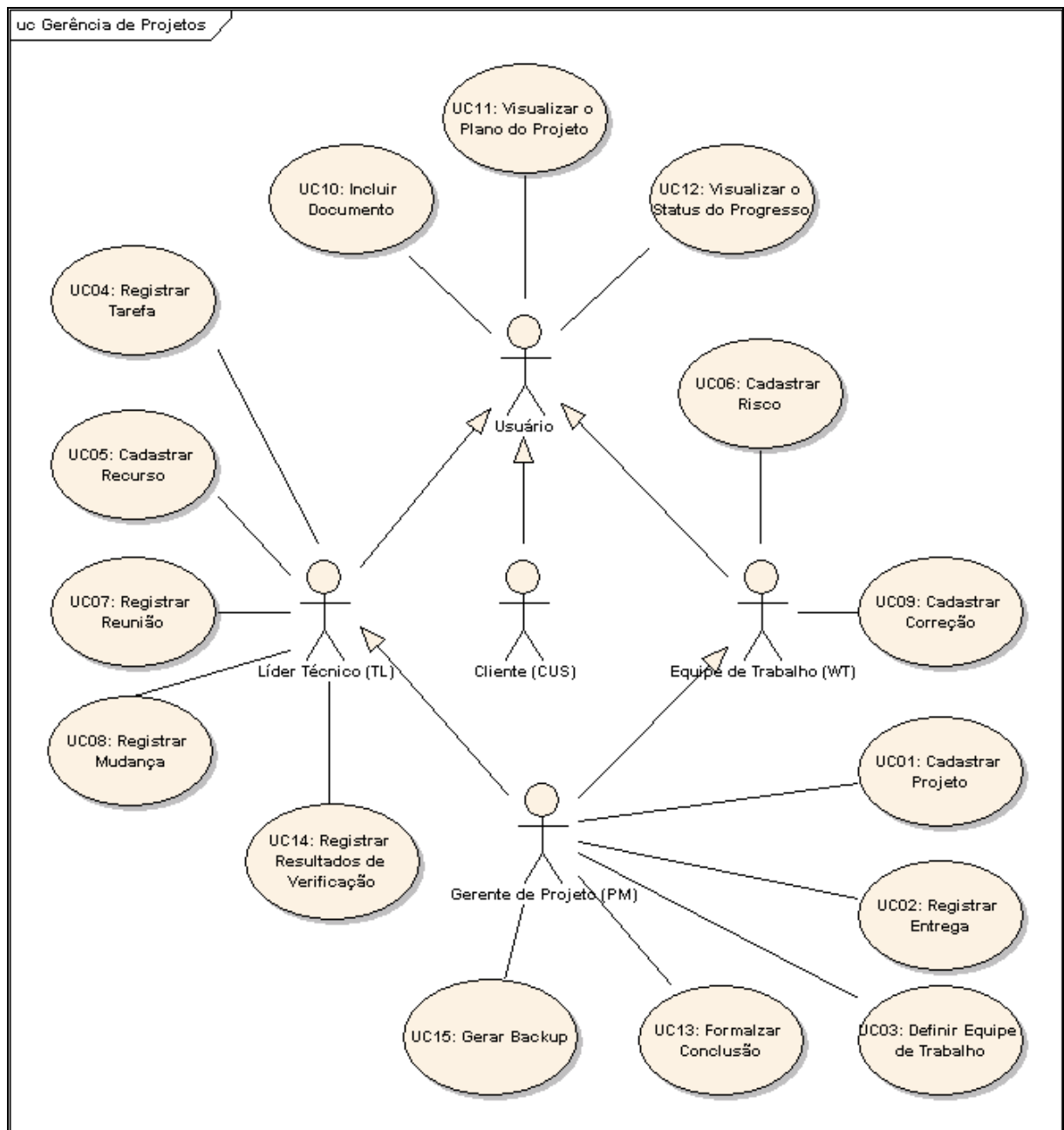
Requisitos Não Funcionais
RNF01: O sistema deve ter acesso controlado por usuário (endereço de correio eletrônico) e senha criptografada.
RNF02: O sistema deve ser desenvolvido para plataforma <i>web</i> .
RNF03: O sistema deve ser desenvolvido utilizando a linguagem Python 2.7.
RNF04: O sistema deve ser desenvolvido utilizando o <i>framework</i> Django 1.5.
RNF05: O sistema deve utilizar o banco de dados PostgreSQL 9.2.
RNF06: O sistema deve ser compatível com os navegadores Google Chrome 31, Internet Explorer 11, Mozilla Firefox 25 e Safari 7.
RNF07: O sistema deve apresentar uma barra de navegação em todas as telas (<i>Breadcrumb</i>).

3.2.2 Diagrama de Casos de Uso

Nesta subseção é apresentado o diagrama de casos de uso do sistema desenvolvido, para melhor exemplificação das atividades exercidas por cada um dos quatro atores caracterizados como: Cliente, Equipe de Trabalho, Gerente de Projetos e Líder Técnico. A descrição dos principais casos de uso (UC01, UC02, UC03, UC04, UC05, UC06, UC07, UC08, UC09 e UC10) está disponível no Apêndice A.

A Figura 8 apresenta o diagrama de casos de uso apresentando as ações que cada ator pode realizar no sistema.

Figura 8 - Diagrama de casos de uso



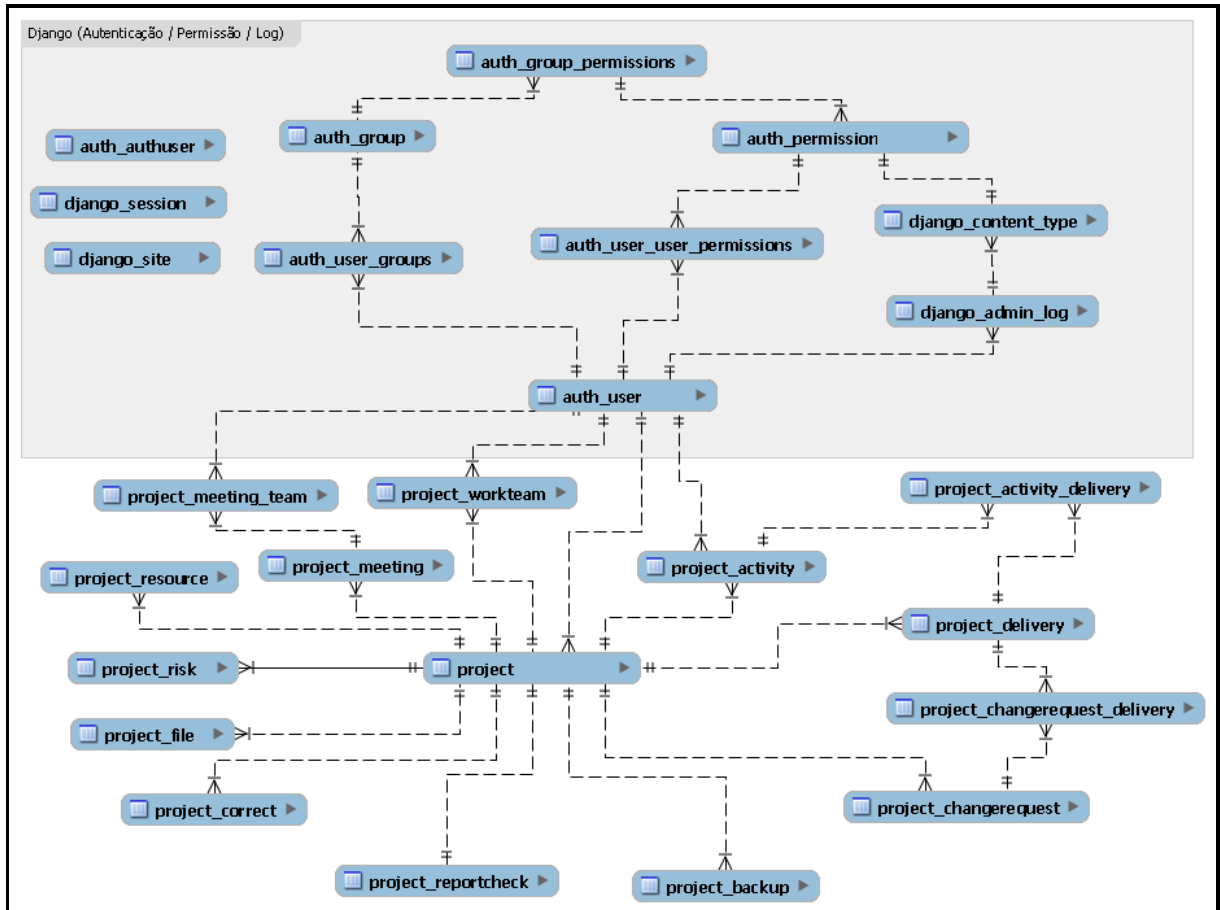
3.2.3 Modelo Entidade Relacionamento

Nesta subseção é apresentado o Modelo de Entidade e Relacionamento (MER) do sistema desenvolvido, onde são mostradas as entidades de autenticação e controle de usuários gerados pelo próprio *framework* Django e também as entidades responsáveis pela persistência de dados gerados pelos usuários do sistema em acordo com as funcionalidades previstas. O dicionário de dados das principais entidades está disponível no Apêndice B. Para

padronização, todas as entidades possuem um prefixo e os nomes de entidades e atributos estão em inglês, que são padrões sugeridos pelo Python *Enhancement Proposals* (PEP).

A Figura 9 apresenta o Modelo de Entidade e Relacionamento, assim como a associação entre as entidades.

Figura 9 - Modelo de entidades e relacionamentos

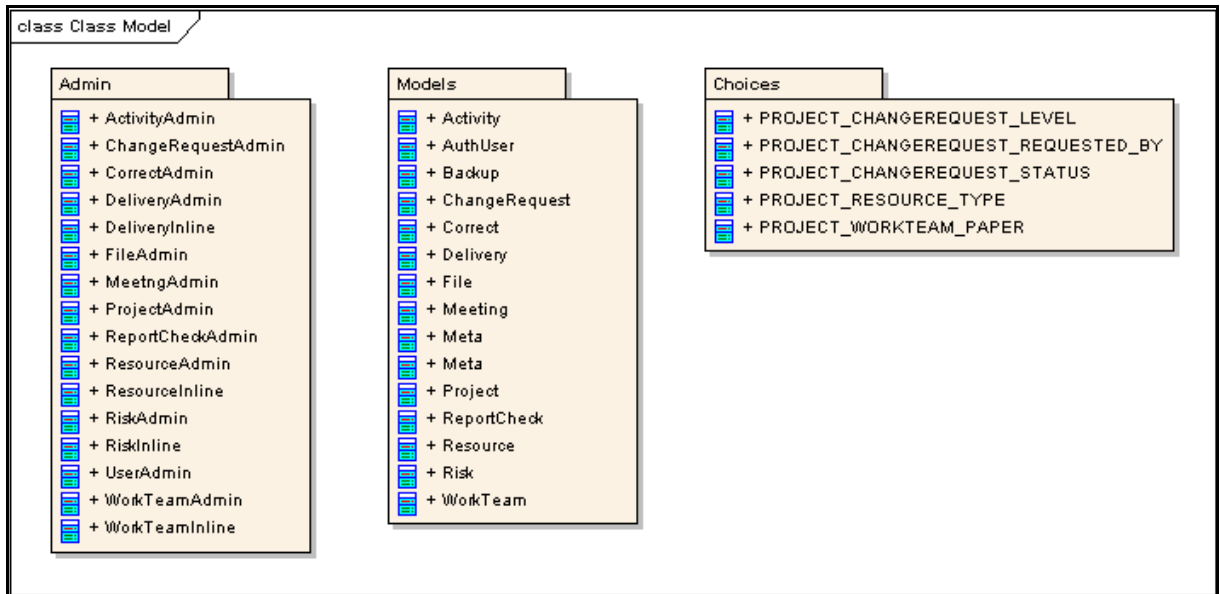


3.2.4 Diagrama de Classes

Nesta subseção é apresentado o diagrama de classes do sistema, seus principais atributos e a associação entre si. As classes seguem a mesma linha das entidades, onde todos os nomes e atributos estão em inglês, assim como o nome dos atributos são equivalentes à respectiva entidade.

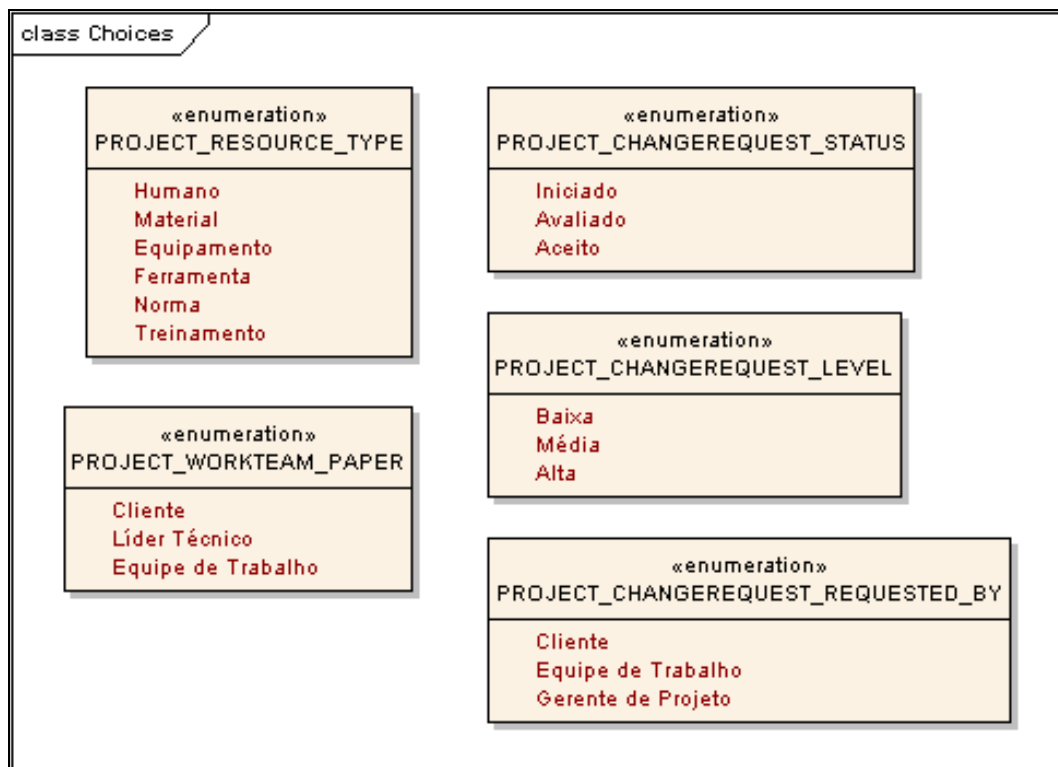
Este projeto possui as classes de modelo (*models*), enumeradores (*choices*) e classes de administração (*admin*). Os pacotes do projeto podem ser visualizados na Figura 10.

Figura 10 – Pacotes do projeto



3.2.4.1 Diagrama de Classes do Pacote *Choices*

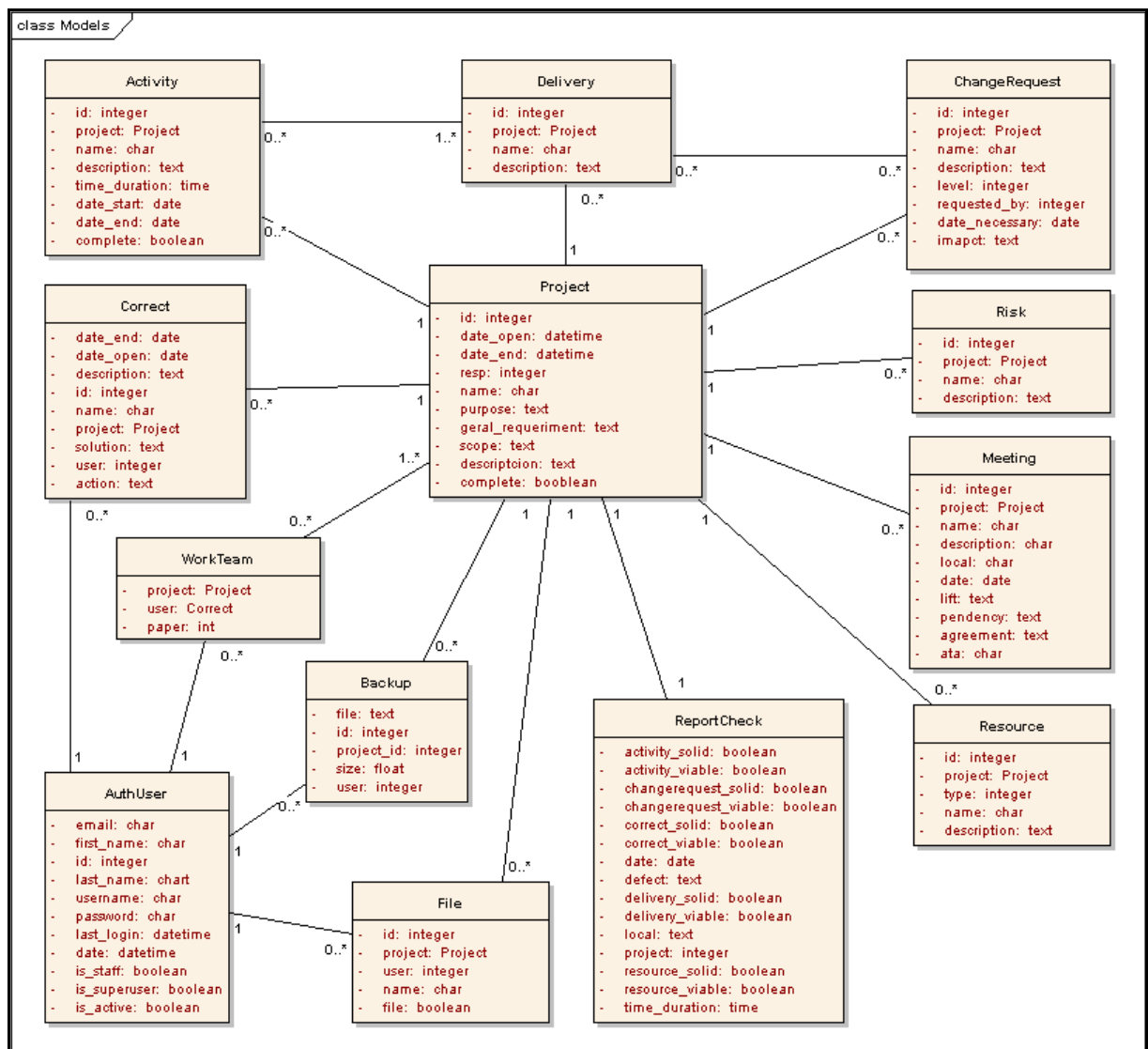
Este pacote possui todas as classes de enumeradores, que são classes com valores fixos que o usuário poderá escolher em determinados formulários do sistema desenvolvido. Na Figura 11 pode-se observar o diagrama de classes dos enumeradores.

Figura 11 - Diagrama de classes do pacote *Choices*

3.2.4.2 Diagrama de Classes do Pacote *Model*

Este pacote possui todas as classes de modelo, que são classes que referenciam todos os objetos previstos nos cadastros do sistema. As classes (exceto a classe `AuthUser`) deste diagrama herdam a classe `models.Model`, que é uma classe base de modelo do *framework* Django que possui todos os métodos necessários para a iteração dos objetos com as demais classes. Já a classe `AuthUser` herda a classe `models.User`, que é a classe padrão o *framework* Django para controle de usuários. Além disso, todas as classes do diagrama possuem uma subclasse chamada `Meta`. Esta subclasse possui atributos específicos para integração dinâmica com os *templates*. Na Figura 12 pode-se observar o diagrama de classes dos modelos.

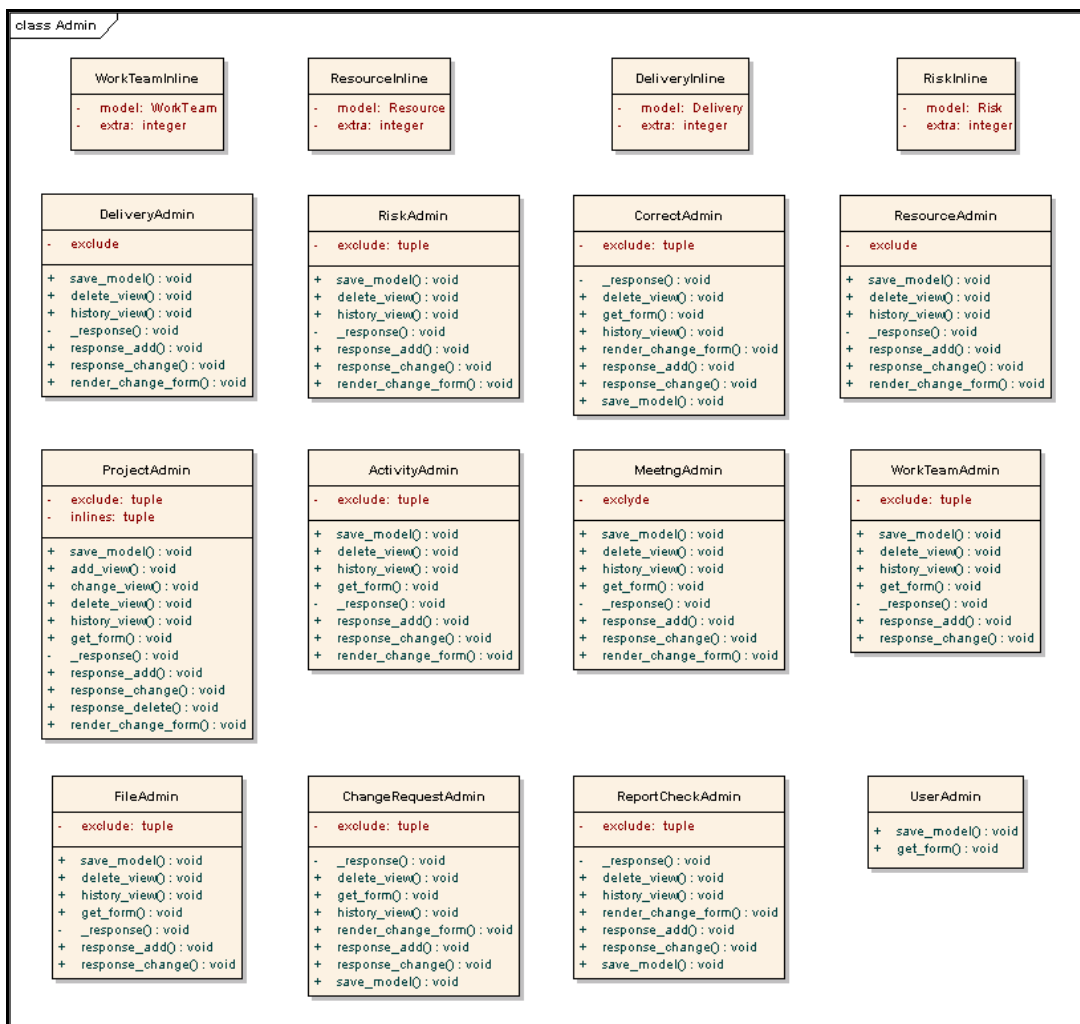
Figura 12 - Diagrama de classes do pacote *Models*



3.2.4.3 Diagrama de Classes do Pacote *Admin*

Este pacote possui todas as classes de administração, que para o *framework* Django são classes de integração dinâmica com os *templates*, que serão interpretadas para apresentação dos formulários. As classes deste diagrama herdam a classe `admin.ModelAdmin`, que é uma classe base de administração do *framework* Django, que possuem todos os métodos necessários para a interação com os *templates*, porém alguns métodos foram sobrescritos, para melhor controle das ações de acordo com a necessidade do sistema desenvolvido. Já as classes onde os nomes são terminados em “*Inline*” herdam a classe `admin.StackedInline`, que são classes base do *framework* Django para permitir a adição de vários registros dentro de um único cadastro, que neste caso é utilizado apenas para a rotina de inserção de projetos, como poderá ser visto na seção 3.3.2 deste documento. Na Figura 13 pode-se observar o diagrama de classes administrativas.

Figura 13 - Diagrama de classes do pacote *Admin*



3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

3.3.1 Técnicas e Ferramentas Utilizadas

Para o desenvolvimento do sistema, foi utilizada a linguagem de programação Python juntamente com o *framework* de desenvolvimento *web* Django. Para a persistência dos dados foi utilizado o banco de dados PostgreSQL.

3.3.1.1 *Framework web* Django

Para desenvolvimento do sistema utilizou-se o *framework* de desenvolvimento *web* Django, que é baseado na linguagem de programação Python e tem por objetivo principal o desenvolvimento rápido com *design clean*.

Os arquivos de código fonte estão estruturados de acordo com a estrutura recomendada pelo DjangoProject, onde cada módulo do projeto trabalha com uma pasta específica contendo toda a sua implementação. No caso deste sistema, utilizou-se o arquivo `models.py` utilizado para definição dos objetos e persistência dos dados, o arquivo `views.py` utilizado para a regra de negócio com saída para um *template HyperText Markup Language* (HTML), onde encontra-se as estruturas de *Cascading Style Sheet* (CSS) e JavaScript e o arquivo `admin.py` possui as regras para disposição de campos nos formulários. O *framework* prevê ainda o arquivo `forms.py`, onde é possível criar regras específicas para formulários, mas neste sistema foi utilizado como base o ambiente customizado de administração do Django, onde não há a necessidade da utilização deste arquivo, sendo que o mesmo foi utilizado apenas para a tela de cadastro de usuário (DJANGOPROJECT, 2013).

Os módulos do projeto estão associados no arquivo `settings.py`, que é responsável por todas as configurações de uma aplicação desenvolvida em Django. Já as *Uniform Resource Locator* (URL) estão mapeadas no arquivo `url.py`, onde o framework localizará as funções a serem executadas sempre que uma requisição for efetuada (DJANGOPROJECT, 2013). A

Figura 14 apresenta parte do código fonte do arquivo settings.py utilizado para o desenvolvimento do sistema.

Figura 14 - Código fonte do arquivo settings.py

```
# Variáveis de modo DEBUG.
DEBUG = TEMPLATE_DEBUG = True

# Admins do serviço.
ADMINS = MANAGERS = (('Thiago Piccinini', 'gpisolectr29110@gmail.com'),)

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycpg2', #'postgresql', 'mysql', 'sqlite3' or 'oracle'.
        'NAME': 'tcc', # Or path to database file if using sqlite3.
        'USER': 'postgres',
        'PASSWORD': 'postgres',
        'HOST': 'localhost', # Empty for localhost through domain sockets or '127.0.0.1' for localhost through TCP.
        'PORT': '5432', # Set to empty string for default.
    }
}

# Backend para controle de autenticação custom
AUTHENTICATION_BACKENDS = ("pm.auth.authbackend.EmailModelBackend",)

# Constantes de controle para o serviço web.
BASE_URL = SITE_URL = 'http://localhost:8080/'
MEDIA_URL = BASE_URL + 'media/'
MEDIA_ROOT = os.path.join(PROJECT_ROOT, 'media')
STATIC_URL = '/static/'

# Dados do servidor de e-mail
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = '587'
EMAIL_HOST_USER = 'gpisolectr29110@gmail.com'
EMAIL_HOST_PASSWORD = '1234'
EMAIL_USE_TLS = True
```

Já a Figura 15 apresenta parte do código fonte do arquivo url.py utilizado para o desenvolvimento do sistema.

Figura 15 - Código fonte do arquivo url.py

```
from django.conf.urls import patterns, include, url
from django.contrib import admin
from django.conf import settings
admin.autodiscover()

urlpatterns = patterns('',

    # Endereço de media: CSS, JavaScript, Imagens, etc.
    (r'^media/(.*)$', 'django.views.static.serve', {'document_root' : settings.MEDIA_ROOT}),

    # Endereço para registro de novos usuários.
    url(r'^auth/register/', 'pm.auth.views.register_user'),

    # Endereços customizados para atender as necessidades específicas do projeto.
    url(r'^project/(?P<appname>\w+)/(?P<id>\d+)/cancel/$', 'pm.views.cancel'),
    url(r'^project/(?P<appname>\w+)/(?P<id>\w+)/cancel/$', 'pm.views.cancel'),
    url(r'^project/project/list', 'pm.project.views.get_projects'),
    url(r'^project/project/(?P<id>\d+)/view/$', 'pm.project.views.project_view'),
    url(r'^project/project/(?P<id>\d+)/backup/$', 'pm.project.views.project_backup'),
    url(r'^project/project/(?P<id>\d+)/complete/$', 'pm.project.views.project_complete'),
    url(r'^project/activity/(?P<id>\d+)/complete/$', 'pm.project.views.activity_complete'),

    # Endereço para testes.
    url(r'^teste/', 'pm.views.teste'),

    # Endereço customizado para logout, redirecionando para a tela inicial (de login).
    url(r'^logout/', 'django.contrib.auth.views.logout', {'next_page': settings.BASE_URL}),

    # Endereços padrões do ambiente de administração do Django.
    url(r'^', include(admin.site.urls)),
)
```


Para os *layouts* das telas utilizou-se uma biblioteca chamada *Bootstrap Admin* que é baseada na usabilidade e funcionalidades do *Twitter*. Esta biblioteca foi desenvolvida baseada nos *templates* padrão do ambiente de administração do *framework* Django, onde para utilização basta adicioná-la na relação de aplicativos instalados do projeto, no arquivo *settings.py*. Como algumas telas do sistema desenvolvido necessitaram de customizações específicas todos os *templates* padrões foram copiados para a pasta principal do projeto, permitindo assim a realização das customizações necessárias.

3.3.1.2 PostgreSQL

O PostgreSQL é um Sistema Gerenciador de Banco de Dados Relacional (SGBDR) que possui mais de 15 anos de desenvolvimento ativo e uma arquitetura comprovada que lhe rendeu uma forte reputação de confiabilidade, integridade e correção. É compatível com os principais sistemas operacionais e suporta o armazenamento de grandes objetos binários, incluindo imagens, sons e vídeos (POSTGRESQL, 2013).

Segundo Pereira Neto (2007) o PostgreSQL é um SGBDR que permite a utilização de *Structured Query Language* (SQL), *triggers*, e toda a gama de programação e construção pertinente aos mais famosos SGBDR do mercado, como Oracle, InterBase, SQL Server e DB/2. Além disso, o PostgreSQL possui diversos *Open Database Connectivity* (ODBC) e *Java Database Connectivity* (JDBC) para interface com ambientes e linguagens de programação, dentre elas, por exemplo: Borland Delphi, Borland C++ Builder, Perl, XML, Java e Python.

O PostgreSQL é um banco de dados à altura do Oracle. Porém em abril de 2002 um artigo de comparação entre bancos de dados foi publicado pelo Linux in Brazil, que mostra claramente que o Oracle tem mais recursos que os demais bancos, mas vem seguido de perto pelo PostgreSQL. Em comparativo de performance *online*, o Oracle e o PostgreSQL são líderes na área de recursos, são similares em escalabilidade, mas o PostgreSQL sente a carga um pouco antes do Oracle (NEVES, 2002).

3.3.2 Operacionalidade da Implementação

Nesta subseção apresentam-se as principais telas do sistema com uma apresentação

sobre as suas funcionalidades, ligação com os objetivos e requisitos e trechos de código fonte relevantes para entendimento de algumas rotinas.

Ao abrir o sistema tem-se a tela de autenticação de usuário, que deve ser efetuado via endereço de correio eletrônico (*e-mail*) e senha. Caso o usuário não seja cadastrado ele poderá se cadastrar, através do *link* disponível na página inicial. Na Figura 16 tem-se a tela principal do sistema.

Figura 16 - Tela de autenticação

Após a autenticação, ou cadastro, o usuário será redirecionado à tela principal do sistema, onde ele terá a relação completa de projetos. Esta relação será dividida em duas listas: uma apresentará os projetos que o usuário criou e é responsável, e a outra apresentará a relação de projetos à qual o usuário foi vinculado. Na Figura 17 tem-se a tela de relação de projetos cadastrados.

Figura 17 - Tela de relação de projetos cadastrados

Título	Criado em	Finalizado em
Cobrança Bancária	16/11/2013	-
Módulo de Compras	16/11/2013	-
Integração ERP	16/11/2013	16/11/2013
Módulo Fiscal	16/11/2013	16/11/2013

Título	Responsável	Criado em	Finalizado em
Terminal de AutoAtendimento	Kaue Raizer	16/11/2013	-

Para carregar a lista de projetos, utilizou-se a tecnologia *JavaScript Object Notation* (JSON). Na Figura 18 tem-se parte do código fonte responsável por encontrar na base de

dados os projetos cadastrados pelo usuário ou os que ele está vinculado, retornando um objeto do tipo JSON.

Figura 18 - Código fonte para selecionar os projetos na base de dados

```

@login_required
def get_projects(request):
    if not request.user:
        return HttpResponseRedirect("/")

    # Se cair aqui, zera a variável, por garantia.
    request.session['PROJECTVIEW'] = None

    json = simplejson.dumps([
        # Projetos próprios.
        [(id, name, date_open.strftime("%d/%m/%Y"), date_end and date_end.strftime("%d/%m/%Y") or "-") for id, name, date_open, date_end in \
         list(Project.objects.filter(resp=request.user.id).values_list("id", "name", "date_open", "date_end").order_by("name", "date_open"))],

        # Projetos de terceiros (Vinculados).
        [(id, name, date_open.strftime("%d/%m/%Y"), User.objects.get(id=resp).get_full_name(), date_end and date_end.strftime("%d/%m/%Y") or "-") \
         for id, name, resp, date_open, date_end in \
         list(Project.objects.filter(id__in=WorkTeam.objects.filter(user=request.user.id).values_list("project")).values_list("id", "name", \
         "resp", "date_open", "date_end").order_by('name'))],
    ])

    return HttpResponse(json, mimetype="application/json")

```

Na Figura 19, tem-se parte do código fonte responsável pela listagem de projetos no *template*.

Figura 19 - Código fonte para listar os projetos no *template*

```

{% block extrahead %}
<script type="text/javascript">

    $(document).ready(function () {
        $.ajax({
            type: "GET",
            url: "{{ BASE_URL }}project/project/list/",
            data: '',
            dataType: "json",
            success: function(result) {
                $.each(
                    result[0],
                    function(i, project) {
                        var line = '';
                        line += '<tr class="row'+i+'">';
                        line += '    <td><a href="{{ BASE_URL }}project/project/'+project[0]+'/'>'+project[1]+'</a></td>';
                        line += '    <td style="text-align: center;"><p>'+project[2]+'</p></td>';
                        line += '    <td style="text-align: center;"><p>'+project[3]+'</p></td>';
                        line += '</tr>';

                        $("#my-projects #changelist #result_list tbody").append(line);
                    }
                )
                $.each(
                    result[1],
                    function(i, project) {
                        var line = '';
                        line += '<tr class="row'+i+'">';
                        line += '    <td><a href="{{ BASE_URL }}project/project/'+project[0]+'/'>'+project[1]+'</a></td>';
                        line += '    <td><p>'+project[3]+'</p></td>';
                        line += '    <td><p>'+project[2]+'</p></td>';
                        line += '</tr>';

                        $("#other-projects #changelist #result_list tbody").append(line);
                    }
                )
            }
        });
    });
</script>
{% endblock %}

```

Na tela de relação de projetos, o usuário poderá cadastrar novos projetos do seu interesse. Esta funcionalidade atende ao requisito funcional RF01. Na Figura 20 tem-se o formulário para cadastro de projetos.

Figura 20 – Tela de cadastro de projeto (Informações básicas)

Gerenciamento de Projetos - ABNT ISO/IEC TR 29110

Bem-vindo(a), Thiago Piccinini. thiago@tcc.com.br

Início > Adicionar Projeto

Adicionar Projeto

Título: Cobrança Bancária

Propósito: Criar um novo módulo que faça toda a integração de cobrança com os bancos.

Requisitos gerais: O módulo deve atender:
- Emissão de boletos;
- Envio de arquivos de remessa;
- Instruções Bancárias;
- Importar arquivo de retorno.

Escopo: O módulo deve ser capaz de controlar toda a integração com o banco e também ligado ao módulo financeiro. Deverá ser compatível com os bancos:
- Banco do Brasil;
- Banrisul;
- Bradesco;
- Itaú;
- HSBC.

Cancelar Salvar e continuar editando Salvar e adicionar outro(a) Salvar

Ainda na tela de cadastro de Projeto, logo após as informações principais, o usuário, se quiser, poderá cadastrar algumas informações específicas, como: Equipe de Trabalho, Entregas, Recursos e Riscos. Neste caso, ele poderá cadastrar quantos registros forem necessários. Na Figura 21 tem-se a tela de cadastro de projeto contendo as informações específicas.

Figura 21 - Tela de cadastro de projeto (Informações específicas)

Gerenciamento de Projetos - ABNT ISO/IEC TR 29110

Bem-vindo(a), Thiago Piccinini. thiago@tcc.com.br

Equipe De Trabalho

Equipe De Trabalho: #1

Papél: Líder Técnico

Usuário: Elton Persike (elton@tcc.com.br)

Adicionar Equipe De Trabalho

Entregas

Adicionar Entrega

Recursos

Adicionar Recurso

Riscos

Risco: #1

Título: Tempo de Homologação

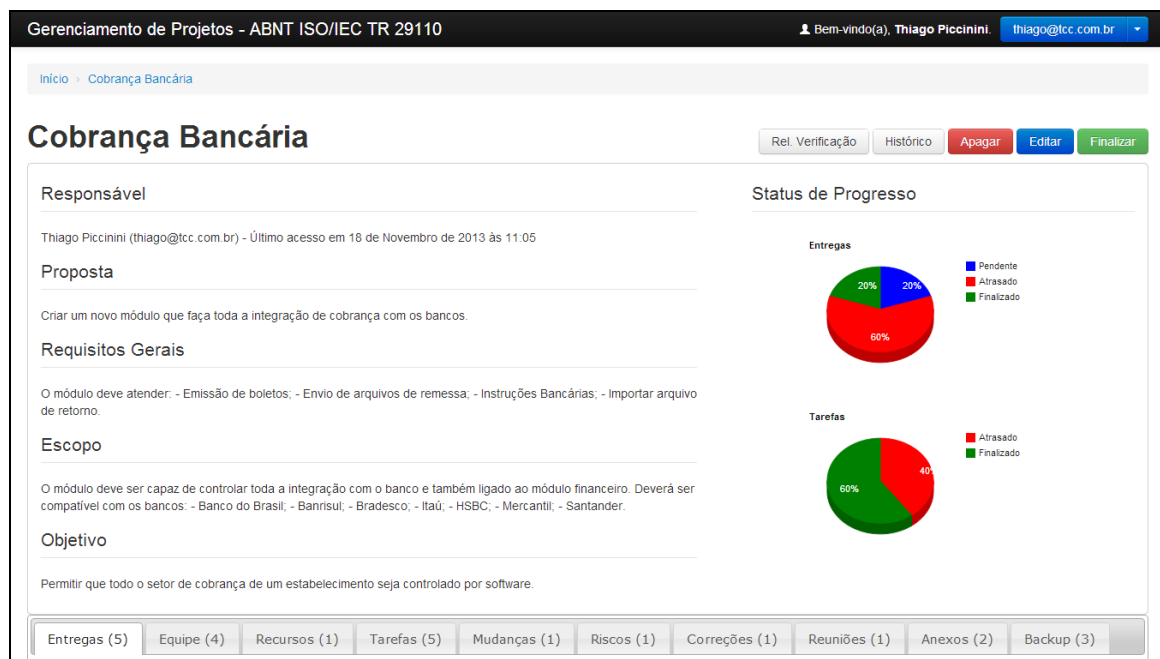
Descrição: Os bancos levarem muito tempo para homologarem os produtos.

Adicionar Risco

Cancelar Salvar e continuar editando Salvar e adicionar outro(a) Salvar

Após o cadastro de um projeto será exibida a tela de visualização. Esta mesma tela é apresentada caso o usuário opte por visualizar um projeto que já está cadastrado. Para facilitar o entendimento dos dados do projeto, esta tela está dividida em duas partes, na parte superior será possível visualizar as informações principais do projeto e também dois gráficos. Os gráficos demonstram o progresso do projeto, indicando as entregas e tarefas que estão pendentes, atrasadas ou concluídas, dando uma visão geral para o usuário do real andamento do projeto. Já na parte inferior da tela de visualização de projetos tem-se as guias responsáveis por cada ação que poderá ser realizada no projeto. Estas guias são: Entregas, Equipe, Recursos, Tarefas, Mudanças, Riscos, Correções, Reuniões, Anexos e *Backup*. Esta funcionalidade atende aos requisitos funcionais RF07 e RF15. Na Figura 22 tem-se a tela de visualização de projetos.

Figura 22 - Tela de visualização do projeto (Parte superior)



Na tela de visualização do projeto o usuário terá a opção de criar o relatório de verificação. Este é um relatório que indica se as informações do projeto são viáveis e consistentes, garantindo que o mesmo seja executado com sucesso. Esta funcionalidade atende ao requisito funcional RF21.

Ainda nesta tela será possível indicar a conclusão do projeto, onde nenhum campo extra será solicitado, apenas será alterado o *status* do mesmo e será gravada a data de conclusão do mesmo. Esta funcionalidade atende ao requisito funcional RF16.

Os gráficos foram gerados utilizando uma *Application Programming Interface* (API) do Google, denominada *Google Charts*. Na Figura 23 tem-se parte do código fonte

responsável por gerar o gráfico de entrega.

Figura 23 - Código fonte para gerar o gráfico entrega

```
{% block extrahead %}
<script type="text/javascript">
  // Carrega a biblioteca de gráficos.
  google.load("visualization", "1", {packages:["corechart"]});

  // Condição para exibir o gráfico somente quando o projeto possuir entregas.
  {% if delivery_list %}
    google.setOnLoadCallback(drawChartDelivery);

    // Variável contendo os tipos de dados que o gráfico apresentará.
    function drawChartDelivery() {
      var data = google.visualization.arrayToDataTable([
        ['Entregas', 'Entregas'],
        ['Pendente', {{ delivery_chart.0 }}],
        ['Atrasado', {{ delivery_chart.1 }}],
        ['Finalizado', {{ delivery_chart.2 }}],
      ]);

      // Atributos opcionais do gráfico (Título, Tamanho, 3D e Cores).
      var options = {
        'title': 'Entregas',
        'width': 400,
        'height': 200,
        'is3D': true,
        'colors': ['blue', 'red', 'green']
      };

      // Adiciona o gráfico na div reservada para o mesmo.
      var chart = new google.visualization.PieChart(document.getElementById('chart-delivery'));
      chart.draw(data, options);
    }
  {% endif %}
</script>
{% endblock %}
```

Na Figura 24 tem-se o código fonte responsável por buscar todas as informações referentes ao projeto para que todos os seus dados possam ser demonstrados.

Figura 24 - Código fonte para buscar as informações do projeto

```

@login_required
def project_view(request, id):

    if not request.user:
        return HttpResponseRedirect("/")

    # -----
    # Busca o projeto pelo ID da URK, caso não encontre retorna para a tela
    # inicial. Se encontrar, salva o mesmo na sessão para controle nos
    # formulários que estão associados ao projeto.

    project = Project.objects.get(id=id)

    if not project:
        return HttpResponseRedirect("/")

    request.session['PROJECTVIEW'] = project

    # -----
    # Dicionário contendo todas as informações necessárias para montar a tela
    # de visualização completa do projeto.

    ret = {
        'delivery_chart': [0, 0, 0], # Pendente, Atrasado, Finalizado
        'activity_chart': [0, 0, 0], # Pendente, Atrasado, Finalizado
        'project': project, # Projeto
        'delivery_list': Delivery.objects.filter(project_id=id).order_by('date_end', 'name'),
        'workteam_list': WorkTeam.objects.filter(project_id=id).order_by('user'),
        'resource_list': Resource.objects.filter(project_id=id).order_by('name', 'type'),
        'activity_list': Activity.objects.filter(project_id=id).order_by('date_start', 'name'),
        'changerequest_list': ChangeRequest.objects.filter(project_id=id).order_by('name'),
        'risk_list': Risk.objects.filter(project_id=id).order_by('name'),
        'correct_list': Correct.objects.filter(project_id=id).order_by('date_open', 'name'),
        'meeting_list': Meeting.objects.filter(project_id=id).order_by('date', 'name'),
        'file_list': File.objects.filter(project_id=id).order_by('date', 'name'),
        'backup_list': Backup.objects.filter(project_id=id).order_by('date').reverse(),
        'reportcheck': ReportCheck.objects.filter(project_id=id) and ReportCheck.objects.get(project_id=id) or None,
    }

    # -----
    # Verifica o papel do usuário no projeto.

    if project.resp.id == request.user.id:
        ret['paper'] = 'PM'

    else:
        for workteam in ret['workteam_list']:
            if workteam.user.id == request.user.id:
                ret['paper'] = workteam.paper
                break

    # -----
    # Controle para o gráfico de atividades.

    for activity in ret['activity_list']:

# -----
# Controle para o gráfico de entregas.
















    for delivery in ret['delivery_list']:

    return render_to_response('project_view.html', RequestContext(request, ret))

```

Na guia Entregas serão listadas todas as entregas associadas ao projeto. Os dados apresentados nesta guia são responsáveis por popular o gráfico de progresso de entregas. Ao lado do título de cada entrega será indicada a situação da mesma, equivalente ao gráfico. Esta funcionalidade atende aos requisitos funcionais RF02 e RF03. Na Figura 25 tem-se a relação das entregas do projeto.

Figura 25 – Tela de relação de entregas

Entregas (5) Equipe (4) Recursos (1) Tarefas (5) Mudanças (1) Riscos (1) Correções (1) Reuniões (1) Anexos (2) Backup (3)		
Título / Descrição	Previsão	Ação
Cadastros Principais (Finalizado) Criar os principais cadastros para atender as necessidades da cobrança bancária: 1) Cadastro de banco; 2) Cadastro de conta corrente; 3) Cadastro de instruções bancárias; 4) Cadastro de portador.	15/10/13	  
Rotina de Geração de Arquivo de Remessa (Atrasado) Criar uma rotina que permita ao usuário gerar os arquivos de remessa que deve ser enviado ao banco.	20/10/13	  
Rotina de Importação de Arquivo de Retorno (Atrasado) Criar uma rotina que permita ao usuário importar o arquivo de retorno fornecido pelo banco, dando baixa nas pendências financeiras.	24/10/13	  
Rotina de Geração de Boletos (Atrasado) Criar uma rotina que permita o usuário gerar os PDFs, assim como enviar por e-mail para cada um dos destinatários.	31/10/13	  
Faturamento (Pendente) Integrar a rotina de cobrança bancária com a rotinas e faturamento.	30/11/13	  

Ainda nesta guia, o usuário poderá cadastrar uma nova entrega cadastrada será necessário indicar uma data de previsão para a entrega. Na Figura 26 tem-se o formulário para o cadastro de novas entregas.

Figura 26 - Tela de cadastro de entregas

Gerenciamento de Projetos - ABNT ISO/IEC TR 29110 Bem-vindo(a), Thiago Piccinini. thiago@tcc.com.br

Início > Cobrança Bancária > Entrega > Adicionar

Adicionar Entrega

Título:

Descrição:

Data: Hoje | 📅
Data ESTIMADA para a entrega.

A guia Equipe demonstra todos os usuários vinculados ao projeto, contendo os dados principais, o seu papel no projeto e a última vez que o mesmo acessou o sistema. Na Figura 27 tem-se a relação de usuários vinculados ao projeto.

Figura 27 - Tela de relação de usuários vinculados ao projeto

Nome	E-mail	Papel	Ult. Acesso	Ação
Kaue Raizer	kaue@tcc.com.br	Equipe de Trabalho	16/11/13 - 01:11:32	  
Ismael Ricardo	ismael@tcc.com.br	Equipe de Trabalho	01/10/13 - 10:10:28	  
Elton Persike	elton@tcc.com.br	Líder Técnico	16/11/13 - 01:11:50	  
Everaldo Grahl	everaldo@tcc.com.br	Cliente	04/10/13 - 09:10:25	  

Nesta guia o responsável pelo projeto poderá vincular novos membros, onde será selecionado o papel e o usuário, sendo ele cadastrado ou não. Esta funcionalidade atende ao requisito funcional RF09. Na Figura 28 tem-se o formulário de cadastro de usuário.

Figura 28 - Tela de cadastro de usuário a equipe

Gerenciamento de Projetos - ABNT ISO/IEC TR 29110 Bem-vindo(a), Thiago Piccinini. thiago@tcc.com.br

Início > Cobrança Bancária > Equipe de Trabalho > Adicionar

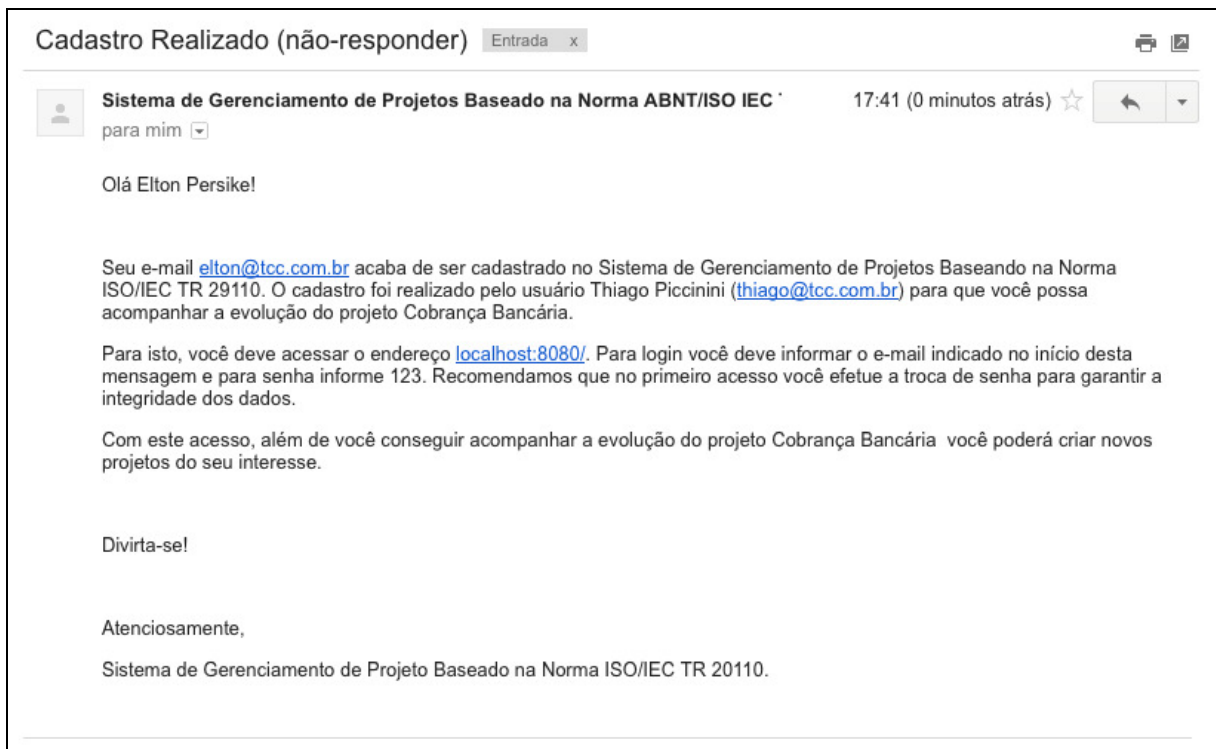
Adicionar Equipe de Trabalho

Papél: Equipe de Trabalho

Usuário: Ismael Ricardo (ismael@tcc.com.br)

Caso o usuário não esteja na lista de usuários cadastrados, o responsável pelo projeto poderá cadastrar o novo usuário ao sistema. Neste caso, ele deverá acessar a opção ao lado do campo Usuário e preencher todas as informações solicitadas no formulário. Após o cadastro, um *e-mail* será enviado para o *e-mail* cadastrado indicando o cadastro e o vínculo ao projeto. Esta funcionalidade atende ao requisito funcional RF09. Na Figura 29 tem-se o *e-mail* que será enviado ao usuário que acaba de ser cadastrado.

Figura 29 – E-mail enviado ao usuário cadastrado pelo responsável do projeto



Na Figura 30 tem-se o código fonte utilizado para enviar o *e-mail* aos usuários cadastros por outros usuários.

Figura 30 - Código fonte para envio de *e-mail*

```
def save_model(self, request, obj, form, change):
    message = "Olá %s %s!\n\nSeu e-mail %s acaba de ser cadastrado no sistema de Gerenciamento de Projetos Baseado "\
    "na Norma ISO/IE TR 29110. O cadastro foi realizado pelo usuário %s (%s) para que você possa acompanhar "\
    "a evolução do projeto %s.\n\nPara isto, você deve acessar o endereço %s. Para login você deve informar o e-mail "\
    "indicado no início desta mensagem e para senha informe %s. Recomendamos que no primeiro acesso você efetue a "\
    "troca de senha para garantir a integridade dos dados.\n\nCom este acesso, além de você acompanhar a evolução "\
    "do projeto %s, você poderá criar novos projetos do seu interesse.\n\n\nDivirta-se!\n\n\nAtenciosamente,\nSistema "\
    "de Gerenciamento de Projetos Baseado na Norma ISO/IEC TR 29110." % (obj.first_name, \
    obj.last_name, obj.email, request.user, request.user.email, request.session['PROJECTVIEW'], settings.SITE_URL, obj.password)

    obj.username = b64encode(md5(obj.email).digest(), '_.')[:30]
    obj.set_password(obj.password)
    obj.is_staff = True
    obj.is_superuser = True




    ret = super(UserAdmin, self).save_model(request, obj, form, change)

    send_mail('Cadastro (nao-responder)', message, 'Gerenciamento de Projetos - ABNT ISO/IEC TR 29110', [obj.email])

    return ret
```












Na guia de Recursos o responsável pelo projeto poderá definir quais os recursos necessários para a execução do projeto, que serão dos seguintes tipos: Humano, Material, Equipamento, Ferramenta, Norma e Treinamento. Esta funcionalidade atende ao requisito funcional RF10. Na Figura 31 tem-se a tela de relação de recursos.

Figura 31 - Tela de relação de recursos

Entregas (5) Equipe (4) Recursos (1) Tarefas (5) Mudanças (1) Riscos (1) Correções (1) Reuniões (1) Anexos (2) Backup (3)			
Título / Descrição		Tipo	Ação
Manuais Será necessário o manual de integração de todos os bancos no padrão Febrabam CNAB 240 e 400 posições.		Material	  

A guia Tarefas apresentará todas as tarefas necessárias para a conclusão dos entregáveis e a conclusão do projeto. Os dados apresentados nesta guia são responsáveis por popular o gráfico de progresso de tarefas, sendo que ao lado do título de cada tarefa será indicada a situação da mesma, equivalente ao gráfico. Esta funcionalidade atende aos requisitos funcionais RF04, RF05 e RF06. Na Figura 32 tem-se a relação das tarefas do projeto.

Figura 32 - Tela de relação de tarefas

Entregas (5) Equipe (4) Recursos (1) Tarefas (5) Mudanças (1) Riscos (1) Correções (1) Reuniões (1) Anexos (2) Backup (3)					
Título / Descrição		Período	Duração	Responsável	Ação
Cadastro de Banco (Finalizado) Criar uma rotina que permita o cadastro dos bancos, contendo código e descrição..		01/10/13 - 01/10/13	01:00	Ismael Ricardo (ismael@tcc.com.br)	
Cadastro de Conta Corrente (Finalizado) Criar uma rotina de cadastro de conta corrente, que permitirá o usuário informar os dados básicos da mesma e também a qual banco ela está ligada.		02/10/13 - 02/10/13	02:00	Kaue Raizer (kaue@tcc.com.br)	
Cadastro de Portador (Finalizado) Criar um cadastro onde o usuário possa fazer todos os vínculos do portador responsável por gerar a cobrança, incluindo: - Local para geração do arquivo de remessa; - Tipo de boleto a ser gerado; - Dados da cobrança (Carteira, Contrato, Convênio e etc).		04/10/13 - 05/10/13	04:00	Elton Persike (elton@tcc.com.br)	
Arquivo de Remessa (Atrasado) Criar uma rotina de que permita a geração do arquivo de remessa. Nesta rotina o usuário poderá pré-visualizar o que será gerado, assim como marcar o que realmente será gerado e também filtrar os dados apresentados.		20/10/13 - 24/10/13	10:00	Ismael Ricardo (ismael@tcc.com.br)	   
Geração de Boletos (Atrasado) Criar uma rotina onde o usuário terá a relação de boletos pendente de geração. Nesta tela ele poderá pré-visualizar e imprimir os boletos, assim como também poderá filtrar os dados apresentados.		21/10/13 - 23/10/13	09:00	Ismael Ricardo (ismael@tcc.com.br)	   

Nesta guia o usuário poderá indicar que a tarefa está concluída ou adicionar uma nova tarefa. Cada tarefa cadastrada será vinculada a uma ou mais entregas e ao um membro da equipe do projeto, assim como a data estimada de início e fim e a duração prevista. Assim como no cadastro de Equipe de Trabalho, caso o usuário não esteja cadastrado ele poderá ser criado, mas neste caso ele será adicionado a equipe com o papel “Equipe de Trabalho”, o mesmo ocorre com as entregas, caso não esteja cadastrada o usuário poderá cadastrá-la. Esta funcionalidade atende aos requisitos funcionais RF17 e RF18. Na Figura 33 tem-se a tela de

cadastro de tarefas.

Figura 33 - Tela de cadastro de tarefas

Gerenciamento de Projetos - ABNT ISO/IEC TR 29110 Bem-vindo(a), Thiago Piccinini | thiago@tcc.com.br

Início > Cobrança Bancária > Tarefa > Adicionar

Adicionar Tarefa

Título: Cadastro de Banco

Descrição: Criar uma rotina que permita o cadastro dos bancos, contendo código e descrição.

Responsável: Ismael Ricardo (ismael@tcc.com.br)

Duração: 01:00:00 Duração ESTIMADA da tarefa. [Agora](#) |

Início: 01/10/13 Data ESTIMADA para início da tarefa. [Hoje](#) |

Fim: 01/10/13 Data ESTIMADA para o término da tarefa. [Hoje](#) |

Atende as entregas:

- Rotina de Geração de Arquivo de Remessa
- Rotina de Importação de Arquivo de Retorno
- Cadastros Principais

A guia Mudanças apresenta as mudanças que surgiram durante o desenvolvimento do projeto. Na tela de cadastro de solicitação de mudança o usuário poderá também cadastrar novas entregas. Esta funcionalidade atende aos requisitos funcionais RF08 e RF23. Na Figura 34 tem-se a tela de relação de mudanças do projeto.

Figura 34 - Tela de relação de solicitações de mudanças

Entregas (5) Equipe (4) Recursos (1) Tarefas (5) **Mudanças (1)** Riscos (1) Correções (1) Reuniões (1) Anexos (2) Backup (3)

Título / Propósito	Necessário para	Solicitado por	Ação
Cadastro de Portador - Disposição dos dados Alterar o cadastro de portador afim de melhorar o layout e a disposição dos dados.	31/10/13	Cliente	

Na guia Riscos o responsável pelo projeto poderá cadastrar todos os fatores que poderão impactar diretamente na conclusão do projeto no prazo estimado. Esta funcionalidade atende ao requisito funcional RF11. Na Figura 35 tem-se a tela de relação de riscos do projeto.

Figura 35 - Tela de relação de riscos



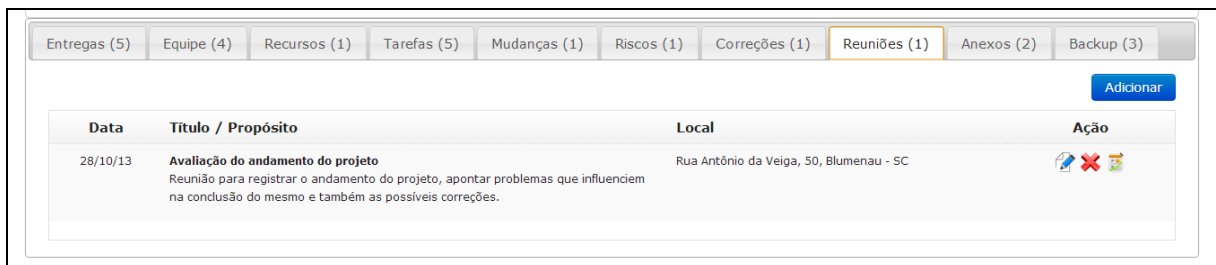
A guia Correções apresentará as correções necessárias que foram encontradas durante o desenvolvimento do projeto. Os usuários poderão documentar qualquer tipo de correção que julgarem necessário. Neste cadastro o usuário deve indicar um responsável pela correção, e assim como nos demais cadastros com este mesmo campo, novos usuários poderão ser adicionados e vinculados a equipe de trabalho do projeto. Esta funcionalidade atende aos requisitos funcionais RF12 e RF24. Na Figura 36 tem-se a tela de relação de correções.

Figura 36 - Tela de relação de correções



Na guia Reuniões poderão ser documentadas as reuniões que foram feitas para discutir o andamento do projeto, seja ela com o cliente ou com a equipe de trabalho. Neste cadastro existe um campo contendo todos os participantes da reunião, e caso o mesmo não esteja na relação também poderá ser adicionado neste momento. Esta funcionalidade atende ao requisitos funcionais RF13 e RF25. Na Figura 37 tem-se a tela de relação de reuniões.

Figura 37 - Tela de relação de reuniões



Na guia Anexos tem-se todos os documentos extras que serão necessários para o projeto. Estes documentos poderão ser adicionados por qualquer usuário vinculado ao projeto, sem restrição de tipo ou extensão. Qualquer usuário poderá também efetuar o *download*

destes arquivos. Esta funcionalidade atende ao requisito funcional RF14. Na Figura 38 tem-se a tela de relação de documentos anexados.

Figura 38 - Tela de relação de anexos

Data / Hora	Título / Propósito	Responsável	Ação
24/10/13 - 21:58:59	Manual Banco do Brasil - CNAB 240 anexos/BB-CNAB240.DOC	Thiago Piccinini (thiago@tcc.com.br)	
24/10/13 - 21:58:59	Manual HSBC- CNAB 440 anexos/HSBC-CNAB400.DOC	Thiago Piccinini (thiago@tcc.com.br)	

Na guia *Backup* tem-se todos os *backups* realizados para o projeto, permitindo também o *download* do mesmo. O *backup* é um arquivo do tipo *Zone Improvement Plan Code* (ZIP) constituído por dois arquivos principais contendo todas as informações do projeto, que serão do tipo *eXtensible Markup Language* (XML) e *JavaScript Object Notation* (JSON) e também uma pasta que conterà todos os documentos que foram anexados ao projeto pelos usuários. Esta funcionalidade atende ao requisito funcional RF22. Na Figura 39 tem-se a tela de relação dos *backups* realizados.

Figura 39 - Tela de relação de *backups*

Data / Hora	Arquivo	Usuário	Ação
18/11/13 - 19:43:04	anexos/backup/COBRANÇABANCÁRIA20131118194313.zip	Thiago Piccinini (thiago@tcc.com.br)	
18/11/13 - 19:24:54	anexos/backup/COBRANÇABANCÁRIA20131118192516.zip	Thiago Piccinini (thiago@tcc.com.br)	
18/11/13 - 19:22:51	anexos/backup/COBRANÇABANCÁRIA20131118192254.zip	Thiago Piccinini (thiago@tcc.com.br)	

Já na Figura 40 tem-se o código fonte responsável por realizar o arquivo de *backup* do projeto e também por inserir o registro do *backup* na base de dados.

Figura 40 - Código fonte para gerar o arquivo de *backup*

```

@login_required
def project_backup(request, id):
    if not request.user:
        return HttpResponseRedirect("/")

    project = Project.objects.get(id=id)

    if not project:
        return HttpResponseRedirect("/")

    project = Project.objects.get(id=id)
    file_list = File.objects.filter(project_id=id)

    # -----
    # Lista de todos os objetos que serão salvos no backup.

    backup_list = list(Project.objects.filter(id=id)) + list(Delivery.objects.filter(project_id=id)) + \
        list(WorkTeam.objects.filter(project_id=id)) + list(Resource.objects.filter(project_id=id)) + \
        list(Activity.objects.filter(project_id=id)) + list(ChangeRequest.objects.filter(project_id=id)) + \
        list(Risk.objects.filter(project_id=id)) + list(Correct.objects.filter(project_id=id)) + \
        list(Meeting.objects.filter(project_id=id)) + list(ReportCheck.objects.filter(project_id=id)) + \
        list(file_list)

    # -----
    # Cria o backup em um arquivo .zip (2 formatos: .json e .xml).

    fjson = os.path.join(settings.MEDIA_ROOT, 'tmp', '%s.json' % project.name.replace(' ', ''))
    f = open(fjson, 'wb').write(serializers.serialize("json", backup_list, indent=2))

    fxml = os.path.join(settings.MEDIA_ROOT, 'tmp', '%s.xml' % project.name.replace(' ', ''))
    f = open(fxml, 'wb').write(serializers.serialize("xml", backup_list, indent=2))

    # Cria um arquivo .zip com os dados do projeto.
    fzip = os.path.join(settings.MEDIA_ROOT, 'anexos', 'backup', '%s%s.zip' % (project.name.replace(' ', '').upper(), \
        datetime.datetime.now().strftime("%Y%m%d%H%M%S")))
    z = zipfile.ZipFile(fzip, "w", zipfile.ZIP_DEFLATED)

    # Adiciona o backup do projeto no .zip
    z.write(fjson)
    z.write(fxml)

    # Adiciona os documentos anexos ao projeto no .zip
    for obj in file_list:

    # Altera o nome do arquivo para eliminar os diretorios
    for i in z.infolist():

    # Salva o Backup
    Backup(project_id=project.id, user_id=request.user.id, file="anexos/backup/%s" % os.path.basename(fzip), \
        size=round((os.path.getsize(fzip)/1024.0/1024.0), 2)).save()

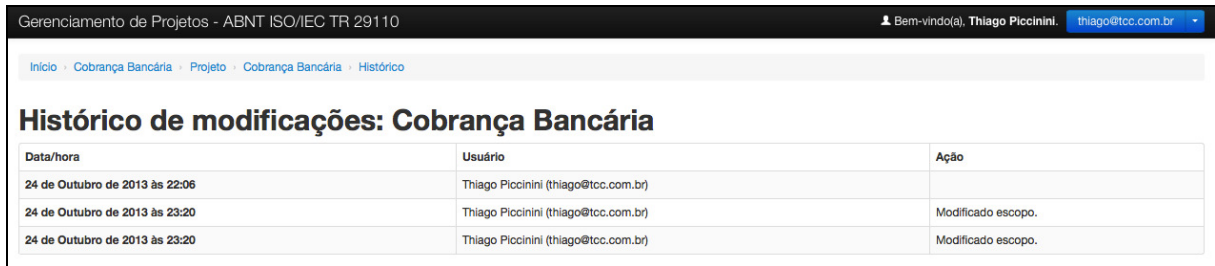
    # Remove os arquivos temporários
    for x in [fxml, fjson]:

    return HttpResponseRedirect("/project/project/%d/view/#tab-backup" % request.session['PROJECTVIEW'].id)

```

Os formulários possuem a opções de verificar o histórico de alterações efetuadas em cada um. Este histórico mostrará a data e hora, usuário e ação realizada. Esta funcionalidade estará disponível para todos os usuários vinculados ao projeto, sem restrição por papel. Esta funcionalidade atende ao requisito funcional RF20. Na Figura 41 tem-se a tela de histórico de modificações de um projeto.

Figura 41 - Tela de histórico de modificações do projeto



Data/hora	Usuário	Ação
24 de Outubro de 2013 às 22:06	Thiago Piccinini (thiago@tcc.com.br)	
24 de Outubro de 2013 às 23:20	Thiago Piccinini (thiago@tcc.com.br)	Modificado escopo.
24 de Outubro de 2013 às 23:20	Thiago Piccinini (thiago@tcc.com.br)	Modificado escopo.

Para auxiliar o usuário na navegação do sistema foi criado um *Breadcrumb*. O *Breadcrumb* nada mais é que uma barra de navegação disponível em todas as telas, indicando o fluxo que o usuário realizou para chegar até a página. Desta forma, ele poderá voltar facilmente para as páginas anteriores. Na Figura 42 tem-se o exemplo de *Breadcrumb* da tela de adicionar reunião.

Figura 42 - Barra de navegação na tela de cadastro de reunião



Todas as guias relacionadas acima possuem as seguintes ações: adicionar, editar, remover e histórico. Apenas a guia tarefa possui uma ação a mais, denominada finalizar. Caso uma tarefa esteja finalizada a mesma não terá mais as ações editar e remover, apenas histórico. O mesmo ocorre para as entregas, porém uma entrega mudará para finalizado apenas quando todas as tarefas vinculadas a ela estiverem finalizadas.

Além disso, as guias possuem também níveis de permissões diferenciados de acordo com o papel de cada usuário da equipe do projeto. Os usuários vinculados ao papel Cliente poderão visualizar todos os dados do projeto e anexar documentos ao projeto. Os usuários vinculados ao papel Líder Técnicos poderão registrar tarefas, cadastrar recursos, registrar reuniões e registrar mudanças. Já os usuários vinculados ao papel Equipe de Trabalho poderão cadastrar correções, cadastrar riscos e marcar uma tarefa como concluída quando o mesmo for responsável pela mesma. O responsável pelo projeto (Gerente de Projeto) é o único usuário que terá total controle sobre todas as funcionalidades do projeto.

3.4 RESULTADOS E DISCUSSÃO

O sistema desenvolvido neste trabalho atendeu as expectativas propostas, possibilitando que um projeto possa ser facilmente gerenciado, levando em consideração as principais tarefas indicadas na norma ABNT ISO/IEC TR 29110. No Quadro 8 tem-se um comparativo entre as funções do sistema de acordo com as tarefas da norma. A coluna resultado poderá apresentar os seguintes valores:

- a) A: Atende;
- b) AP: Atende Parcialmente;
- c) NA: Não Atende.

Quadro 8 - Comparativo entre funções do sistema e as tarefas previstas pela norma

Tarefas Gerenciamento de Projetos ABNT ISO/IEC TR 29110		Aderência a Norma ABNT ISO/IEC TR 29110	
PM.1 PLANEJAMENTO DO PROJETO	PM.1.1 Revisar a declaração de trabalho.	A	O sistema permite o registre dos tópicos apontados na declaração de trabalho.
	PM.1.2 Definir com o cliente as instruções de entrega.	A	O sistema permite o registro de entregas necessárias para que os objetivos do projeto sejam cumpridos.
	PM.1.3 Identificar as Tarefas específicas a serem realizadas para produzir os entregáveis.	A	O sistema permite o registro de todas as tarefas necessárias para concluir o projeto, indicando também a(s) entrega(s) que cada uma contemplará.
	PM.1.4 Estabelecer a duração estimada para executar cada tarefa.	A	O sistema permite o registre destas informações no cadastro de tarefas.
	PM.1.5 Identificar e documentar os recursos.	A	O sistema permite o cadastro de recursos necessários para executar o projeto.

PM.1.6 Estabelecer a composição da equipe de trabalho.	A	O sistema permite o vínculo de usuários a equipe de trabalho do projeto.
PM.1.7 Atribuir as datas de início e de conclusão estimadas para cada uma das tarefas.	A	O sistema permite o registro destas informações no cadastro de tarefas.
PM.1.8 Calcular e documentar a estimativa de custo e esforço do projeto.	NA	O sistema não possui nenhuma funcionalidade para esta tarefa, mas permite que os usuários anexem um documento com o cálculo.
PM.1.9 Identificar e documentar os riscos que podem afetar o projeto.	A	O sistema permite o cadastro de todos os riscos previstos para o projeto.
PM.1.10 Documentar a estratégia de controle de versão no plano do projeto.	NA	O sistema não possui nenhuma funcionalidade para esta tarefa, mas permite que os usuários anexem um documento com as estratégias.
PM.1.11 Gerar o plano do projeto integrando todos os elementos identificados e documentados anteriormente.	AP	O sistema gera um relatório em tela do plano de projeto, mas apresenta uma tela baseada nas informações previstas do plano do projeto. Esta tela apresentará todos os dados necessários para o entendimento e acompanhamento do projeto.
PM.1.12 Incluir a descrição do produto, escopo, objetivos, e entregáveis no plano do projeto.	A	O sistema permite que estas informações sejam descritas no formulário de cadastro de projeto.
PM.1.13 Verificar e obter a aprovação do plano do projeto.	A	O sistema possui uma funcionalidade para que os usuários registrem os resultados de verificação.
PM.1.14 Revisar e aceitar o plano do projeto.	A	O sistema permite que o registro de reuniões e apresenta o plano do projeto.

	PM.1.15 Estabelecer o repositório de projeto usando a estratégia de controle de versão.	NA	O sistema permite o anexo de diversos documentos ao projeto, o que pode ser utilizado como repositório do projeto.
PM.2 EXECUÇÃO DO PLANO DO PROJETO	PM.2.1 Monitorar a execução o plano do projeto e registrar o realizado no registro de <i>status</i> de progresso.	A	O sistema apresenta gráficos que demonstram o <i>status</i> de progresso do projeto. São dois gráficos baseados na situação das entregas e das atividades, que podem ser: Atrasado, Concluído ou Pendente.
	PM.2.2 Analisar e avaliar a solicitação de mudança.	A	O sistema permite o registro de mudanças que são visualizadas durante o desenvolvido do projeto, além de indicar quais entregas serão afetadas com a mudança registrada.
	PM.2.3 Conduzir reuniões de revisão com a equipe de trabalho.	A	O sistema permite o registro de reuniões, permitindo também que uma ata seja anexada.
	PM.2.4 Conduzir reuniões de revisão com o cliente	A	O sistema permite o registro de reuniões, permitindo também que uma ata seja anexada.
	PM.2.5 Fazer <i>backups</i> de acordo com a estratégia de controle de versão.	A	O sistema permite a criação de <i>backup</i> de todos os dados do projeto, e cada <i>backup</i> é registrado.
	PM.2.6 Fazer a recuperação do repositório do projeto.	NA	Esta tarefa não é atendida.

PM.3 AVALIAÇÃO E CONTROLE DO PROJETO	PM.3.1 Avaliar o progresso do projeto com respeito ao plano de projeto.	AP	O sistema não avalia o progresso do projeto, mas apresenta gráficos baseados em entregas e atividades, onde será demonstrado a situação de cada uma, que pode ser: Atrasado, Concluído ou Pendente.
	PM.3.2 Estabelecer ações para corrigir problemas de desvios e riscos identificados.	A	O sistema permite o registro de correções de erros e também os riscos identificados durante a execução do projeto.
	PM.3.3 Identificar mudanças nos requisitos e/ou plano do projeto.	A	O sistema permite o registro de mudanças que são visualizadas durante o desenvolvimento do projeto, além de indicar quais entregas serão afetadas com a mudança registrada.
PM.4 ENCERRAMENTO DO PROJETO	PM.4.1 Formalizar a conclusão do projeto.	A	O sistema permite a conclusão de um projeto criado como concluído e anexo o termo de conclusão do projeto, onde poderá constar toda a documentação de conclusão e também a assinatura de todos os envolvidos.
	PM.4.2 Atualizar o repositório do projeto.	AP	O sistema permite o anexo de documentos e também permite a alteração do mesmo.

Baseado no Quadro 8 pode-se afirmar que o sistema desenvolvido atende 84,61% (22 tarefas) das tarefas previstas pelo processo de Gerência de Projetos, sendo que 73,07% (19 tarefas) são atendidas por completo, 11,53% (3 tarefas) são atendidas parcialmente e 15,38% (4 tarefas) não são atendidas.

Foi realizado um comparativo resumido sobre os trabalhos correlatos estudados. Percebe-se que todos os trabalhos foram desenvolvidos para a plataforma *web*, porém o trabalho desenvolvido utiliza linguagem de programação e banco de dados diferenciados.

Também é possível perceber que os trabalhos correlatos foram baseados em padrões de gerenciamento de projetos complexos, já o trabalho desenvolvido foi baseado em uma norma recente visando as micro-organizações. No Quadro 9 tem-se um comparativo do sistema desenvolvido com os trabalhos correlatos.

Quadro 9 - Comparativo do sistema desenvolvido com os trabalhos correlatos

Tópico	GIELOW	BITELLBRUNN	GIACOMOZZI	PICCININI
Plataforma	<i>web</i>	<i>web</i>	<i>web</i>	<i>web</i>
Linguagem	PHP	Java	PHP	Python
<i>Framework</i>	CakePHP	GWT-Ext	CakePHP	Django
Banco de Dados	MySQL	MySQL e HeidiSQL	MySQL	PostgreSQL
Norma/Padrão	PMBOK	PMBOK	Scrum	ABNT ISO/IEC TR 29110

4 CONCLUSÕES

Devido à dificuldade em encontrar um sistema de gerência de projetos para as pequenas e médias empresas, desenvolveu-se um sistema focado nestas empresas, visando atender as principais tarefas do processo previsto pela norma ABNT ISO/IEC TR 29110.

Com o sistema desenvolvido neste trabalho, o gerente de projetos poderá controlar todas as etapas do projeto em uma interface simples, assim como os envolvidos no projeto também terão acesso a estes dados. Em resumo, todas as pessoas vinculadas ao projeto poderão visualizar facilmente as informações e acompanhar a evolução de cada projeto.

Os resultados obtidos com o desenvolvimento deste sistema foram satisfatórios, onde os requisitos propostos foram alcançados e os objetivos do trabalho foram atendidos. As principais tarefas e atividades foram implementadas, atendendo cerca de 75% das atividades previstas pela norma, permitindo que as micro-organizações controlem seus projetos e sejam reconhecidas como entidades que produzem *software*.

As ferramentas *open source* foram a escolha adequada para a implementação do trabalho, encaixando-se perfeitamente nas necessidades do mesmo. Como linguagem de programação principal utilizou-se o Python juntamente com o *framework web* Django que se mostrou muito eficiente na criação de formulários e regras. Já para a criação de *layout* de telas utilizou-se a biblioteca Bootstrap e para os gráficos a biblioteca Google *Charts*, dando ao sistema um aspecto limpo e objetivo. Procedeu-se de tal forma para extrair das ferramentas as suas qualidades, permitindo elaborar o sistema de forma satisfatória para atingir o objetivo proposto.

Por fim, conclui-se que o desenvolvimento deste trabalho contribuiu positivamente para o conhecimento profissional e pessoal do autor, onde aprendeu-se novas tecnologias até então não exploradas, assim como o conhecimento no processo de gerenciamento de projetos. O desenvolvimento deste proporcionou ao autor entusiasmo para seguir e se aperfeiçoar nos processos de gerenciamento de projetos, assim como novas qualificações possíveis.

4.1 EXTENSÕES

Como sugestão de extensões para o presente trabalho, tem-se:

- a) desenvolver novas funcionalidades para indicação de resultados, ou seja, relatórios, documentos e gráficos previstos pela norma ABNT ISO/IEC TR 29110;
- b) suportar o processo de Implementação de Software (SI) previsto na norma ABNT ISO/IEC TR 29110, para que um único sistema possa auxiliar a norma estudada;
- c) desenvolver um controle de testes de qualidade para cada tarefa e/ou entrega prevista no projeto, onde o usuário poderá documentar os testes e também o resultado de cada um.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ISO/IEC TR 29110-5-1-2: Engenharia de Software – Perfis de ciclo de vida para micro-organizações (VSEs)**. Rio de Janeiro, 2012.

ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARES BRASILEIRO. **MPS.BR – Melhorias de Processo do Software Brasileiro**. Brasília, 2012. Disponível em: <http://softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte_12_2012.pdf>. Acesso em: 08 abr. 2013.

BITTELBRUNN, Taynara. **Ferramenta de Gerência de Projetos Segundo Diretrizes do PMBOK**. 2009. 96 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <<http://campeche.inf.furb.br/tccs/2009-I/TCC2009-1-20-VF-TaynaraBittelbrunn.pdf>>. Acesso em: 10 out. 2013.

DJANGOPROJECT. **The Web framework for perfectionists with deadlines**. [S.l.], 2013. Disponível em: <<https://www.djangoproject.com/>>. Acesso em: 30 set. 2013.

FREEBITS. **Gerenciamento de Projetos**. São José dos Campos, 2010. Disponível em: <http://www.freebits.com.br/gerencia_projetos.jsf>. Acesso em: 23 nov. 2013.

GIACOMOZZI, Robson R. **Sistema para Gestão Ágil de Projetos de Software**. 2011, 64 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <<http://campeche.inf.furb.br/tccs/2011-I/TCC2011-1-20-VF-RobsonRGiacomozzi.pdf>>. Acesso em: 10 out. 2013.

GIELOW, Guilherme F. **Sistema de Informação para Controle do Gerenciamento de Projeto Baseado no PMBOK**. 2008. 82 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <<http://campeche.inf.furb.br/tccs/2008-I/2008-1-16-vf-guilhermefgielow.pdf>>. Acesso em: 10 out. 2013.

MAESTROSDDELWEB. **Curso Django: Entendiendo Como Trabaja o Django**. [S.l.], 2012. Disponível em: <<http://www.maestrosdelweb.com/editorial/curso-django-entendiendo-como-trabaja-django/>>. Acesso em: 3 dez. 2013.

NEVES, Denise Lemes Fernandes. **PostgreSQL: conceitos e aplicações**. São Paulo : Érica, 2002. 188p, il.

ORTH, Afonso Inácio; PRIKLADNICKI, Rafael. **Planejamento e gerência de projetos**. Porto Alegre : EIPUCRS, 2009. 179 p, il.

PEREIRA NETO, Álvaro. **PostgreSQL: técnicas avançadas : versões Open Source 7.x e 8.x : soluções para desenvolvedores e administradores de banco de dados**. 4. ed. São Paulo : Érica, 2007. 284 p, il.

PFEIFFER, Peter; MAGALHÃES, Giovana. **Gerenciamento de projetos de desenvolvimento: conceitos, instrumentos e aplicações**. Rio de Janeiro : BRASPORT, 2005. 182 p, il.

PIETRO, Alcides. **O que é gestão de projetos?**. São Paulo, 2002. Disponível em: <http://www.proage.com.br/proage/exe/empresa/publicacoes/artigo_oquegp.pdf>. Acesso em: 23 nov. 2013.

POSTGRESQL. **Sobre**. [S.l.], 2013. Disponível em: <<http://www.postgresql.org/about/>>. Acesso em: 30 set. 2013.

PROJECT MANAGEMENT INSTITUTE. **O que é Gerenciamento de Projetos?**. [S.l.], 2013a. Disponível em: <<http://brasil.pmi.org/brazil/AboutUS/WhatIsProjectManagement.aspx>>. Acesso em: 23 nov. 2013.

PROJECT MANAGEMENT INSTITUTE. **Quem são os Gerentes de Projeto?**. [S.l.], 2013b. Disponível em: <<http://brasil.pmi.org/brazil/AboutUS/WhoareProjectManagers.aspx>>. Acesso em: 23 nov. 2013.

PYTHON. **Python Software Foundation: About Python**. [S.l.], 2013. Disponível em: <<http://www.python.org/about/>>. Acesso em: 30 set. 2013.

ROSSUM, Guido Van; **The History of Python: Introduction and Overview**. [S.l.], 2009. Disponível em: <<http://python-history.blogspot.com.br/2009/01/introduction-and-overview.html>>. Acesso em: 24 set. 2013.

SANTANA NETO, Osvaldo; GALES, Thiago. **Python e Django: desenvolvimento ágil de aplicações Web**. São Paulo : Novatec, 2010. 279 p., il.

APÊNDICE A – Descrição dos Casos de Uso

Neste Apêndice apresenta-se o detalhamento dos principais casos de uso contemplados no diagrama apresentado na seção 3.2.2 deste trabalho.

No Quadro 10 tem-se o detalhamento do caso de uso Cadastrar Projeto.

Quadro 10 - Caso de uso Cadastrar Projeto

Caso de uso: UC01 – Cadastrar Projeto

Permitir que o Gerente de Projeto cadastre um novo projeto.

Ator: Gerente de Projeto.

Pré-condição: Estar logado no sistema.

Pós-condição: O projeto é cadastrado e a tela de visualização é apresentada.

Cenário principal:

1. Gerente de Projeto acessa a opção de adicionar projeto;
2. Sistema apresenta o formulário de cadastro de projeto;
3. Gerente de Projeto informa os campos solicitados no formulário;
4. Gerente de Projeto salva os dados do formulário;
5. Sistema valida os campos informados;
6. Sistema persiste os dados na base de dados;
7. Sistema apresenta a tela de visualização do projeto.

Cenário alternativo - Editar:

No passo 7 do cenário principal, o Gerente de Projeto possui a opção de editar os dados do projeto.

1. Gerente de Projeto acessa a opção de editar projeto;
2. Sistema apresenta o formulário de cadastro de projeto com os campos preenchidos;
3. Gerente de Projeto altera os campos que julgar necessário;
4. Gerente de Projeto salva as alterações realizadas;
5. Sistema valida os campos informados;
6. Sistema persiste os dados na base de dados;
7. Sistema apresenta a tela de visualização de projeto.

Cenário alternativo – Apagar:

No passo 7 do cenário principal, o Gerente de Projeto possui a opção de apagar o projeto.

1. Gerente de Projeto acessa a opção de apagar projeto;
2. Sistema valida as demais informações vinculadas ao projeto;
3. Sistema exclui todas as informações vinculadas ao projeto;
4. Sistema exclui o projeto;
5. Sistema apresenta a tela de relação de projetos.

Cenário alternativo – Histórico:

No passo 7 do cenário principal, o Gerente de Projeto possui a opção de visualizar o histórico de alterações do projeto.

1. Gerente de Projeto acessa a opção de histórico de modificações do projeto;
2. Sistema apresenta a relação de modificações do projeto.

Cenário de exceção:

No passo 5 do cenário principal e no passo 5 do cenário alternativo editar, caso os campos obrigatórios não forem preenchidos corretamente, o sistema apresentará uma mensagem informando que há erros nos dados informados que precisam ser corrigidos para prosseguir. Informará também quais os campos obrigatórios que não foram informados.

No Quadro 11 tem-se o detalhamento do caso de uso Registrar Entrega.

Quadro 11 - Caso de uso Registrar Entrega

Caso de uso: UC02 – Registrar Entrega

Permitir que o Gerente de Projeto registre uma nova entrega.

Ator: Gerente de Projeto.

Pré-condição: Estar logado no sistema e estar visualizando um projeto cadastrado.

Pós-condição: A entrega é registrada e a tela de visualização de projeto é apresentada, com foco na guia entregas.

Cenário principal:

1. Gerente de Projeto acessa a guia entregas;
2. Gerente de Projeto acessa a opção de adicionar entrega;

3. Sistema apresenta o formulário de registro de entrega;
4. Gerente de Projeto informa os campos solicitados no formulário;
5. Gerente de Projeto salva os dados do formulário;
6. Sistema valida os campos informados;
7. Sistema persiste os dados na base de dados;
8. Sistema apresenta a tela de visualização do projeto com foco na guia entrega.

Cenário alternativo – Editar:

No passo 1 do cenário principal, o Gerente de Projeto possui a opção de editar uma entrega.

1. Gerente de Projeto acessa a opção de editar entrega;
2. Sistema apresenta o formulário de registro de entrega com os campos preenchidos;
3. Gerente de Projeto altera os campos que julgar necessário;
4. Gerente de Projeto salva as alterações realizadas;
5. Sistema valida os campos informados;
6. Sistema persiste os dados na base de dados;
7. Sistema apresenta a tela de visualização do projeto com foco na guia entrega.

Cenário alternativo – Apagar:

No passo 1 do cenário principal, o Gerente de Projeto possui a opção de apagar uma entrega.

1. Gerente de Projeto acessa a opção de apagar entrega;
2. Sistema exclui a entrega;
3. Sistema apresenta a tela de visualização do projeto com foco na guia entrega.

Cenário alternativo – Histórico:

No passo 1 do cenário principal, o gerente de projeto possui a opção de verificar o histórico de modificações de uma entrega.

1. Gerente de Projeto acessa a opção de histórico de modificações da entrega;
2. Sistema apresenta a relação de modificações da entrega.

Cenário de exceção:

No passo 6 do cenário principal e no passo 5 do cenário alternativo editar, caso os campos obrigatórios não forem preenchidos corretamente, o sistema apresentará uma mensagem informando que há erros nos dados informados que precisam ser corrigidos para prosseguir.

Informará também quais os campos obrigatórios que não foram informados.

No passo 2 do cenário alternativo apagar, caso exista uma tarefa vinculada a entrega, o sistema não permitirá que a entrega seja excluída.

No Quadro 12 tem-se o detalhamento do caso de uso Definir Equipe de Trabalho.

Quadro 12 - Caso de uso Definir Equipe de Trabalho

Caso de uso: UC03 – Definir Equipe de Trabalho

Permitir que o Gerente de Projeto defina um novo membro a equipe de trabalho.

Ator: Gerente de Projeto.

Pré-condição: Estar logado no sistema e estar visualizando um projeto cadastrado.

Pós-condição: O usuário é vinculado como membro da equipe e a tela de visualização de projeto é apresentada, com foco na guia equipe.

Cenário principal:

1. Gerente de Projeto acessa a guia equipe;
2. Gerente de Projeto acessa a opção de adicionar membro a equipe;
3. Sistema apresenta o formulário de cadastro de membro da equipe;
4. Gerente de Projeto informa os campos solicitados no formulário;
5. Gerente de Projeto salva os dados do formulário;
6. Sistema valida os campos informados;
7. Sistema persiste os dados no banco de dados;
8. Sistema apresenta a tela de visualização do projeto com foco na guia equipe.

Cenário alternativo – Responsável:

No passo 3 do cenário principal, o Gerente de Projeto deve vincular um responsável para a função e caso o responsável não esteja cadastrado o gerente de projeto poderá cadastrá-lo.

1. Gerente de Projeto acessa a opção de adicionar usuário;
2. Sistema apresenta o formulário de cadastro de usuário;
3. Gerente de Projeto informa os campos solicitados no formulário;
4. Gerente de Projeto salva os dados do formulário;
5. Sistema valida os campos informados;

6. Sistema persiste os dados no banco de dados;
7. Sistema envia *e-mail* ao usuário cadastrado informando o cadastro;
8. Sistema volta para o formulário de cadastro de membro da equipe preenchendo o campo usuário com o cadastrado.

Cenário alternativo – Editar:

No passo 1 do cenário principal, o Gerente de Projeto possui a opção de editar um membro da equipe de trabalho.

1. Gerente de Projeto acessa a opção de editar o membro da equipe de trabalho;
2. Sistema apresenta o formulário de cadastro de membro a equipe com os campos preenchidos;
3. Gerente de Projeto altera os campos que julgar necessário;
4. Gerente de Projeto salva as alterações realizadas;
5. Sistema valida os campos informados;
6. Sistema persiste os dados na base de dados;
7. Sistema apresenta a tela de visualização do projeto com foco na guia equipe.

Cenário alternativo – Apagar:

No passo 1 do cenário principal, o Gerente de Projeto possui a opção de apagar um membro da equipe de trabalho.

1. Gerente de Projeto acessa a opção de apagar o membro da equipe de trabalho;
2. Sistema exclui o membro da equipe de trabalho;
3. Sistema apresenta a tela de visualização do projeto com foco na guia equipe.

Cenário alternativo – Histórico:

No passo 1 do cenário principal, o Gerente de Projeto possui a opção de verificar o histórico de modificações de um membro da equipe de trabalho.

1. Gerente de Projeto acessa a opção de histórico de modificações da equipe de trabalho;
2. Sistema apresenta a relação de modificações da equipe de trabalho.

Cenário de exceção:

No passo 6 do cenário principal, no passo 6 do cenário alternativo responsável e no passo 5 do cenário alternativo editar, caso os campos obrigatórios não forem preenchidos corretamente,

o sistema apresentará uma mensagem informando que há erros nos dados informados que precisam ser corrigidos para prosseguir. Informará também quais os campos obrigatórios que não foram informados.

No Quadro 13 tem-se o detalhamento do caso de uso Registrar Tarefa.

Quadro 13 - Caso de uso Registrar Tarefa

Caso de uso: UC04 – Registrar Tarefa

Permitir que o Líder Técnico registre uma nova tarefa.

Ator: Líder Técnico.

Pré-condição: Estar logado no sistema e estar visualizando um projeto cadastrado.

Pós-condição: A tarefa é registrada e a tela de visualização de projeto é apresentada, com foco na guia tarefas.

Cenário principal:

1. Líder Técnico acessa a guia tarefas;
2. Líder Técnico acessa a opção de adicionar tarefa;
3. Sistema apresenta o formulário de registro de tarefa;
4. Líder Técnico informa os campos solicitados no formulário;
5. Líder Técnico salva os dados do formulário;
6. Sistema valida os campos informados;
7. Sistema persiste os dados no banco de dados;
8. Sistema apresenta a tela de visualização do projeto com foco na guia tarefa.

Cenário alternativo – Responsável:

No passo 3 do cenário principal, o Líder Técnico deve vincular um responsável para a tarefa e caso o responsável não esteja cadastrado o usuário poderá cadastrá-lo.

1. Líder Técnico acessa a opção de adicionar usuário;
2. Sistema apresenta o formulário de cadastro de usuário;
3. Líder Técnico informa os campos solicitados no formulário;
4. Líder Técnico salva os dados do formulário;
5. Sistema valida os campos informados;
6. Sistema persiste os dados no banco de dados;

7. Sistema vincula o usuário a equipe de trabalho do projeto;
8. Sistema envia *e-mail* ao usuário cadastrado informando o cadastro;
9. Sistema volta para o formulário de cadastro de tarefa preenchendo o campo usuário com o cadastrado.

Cenário alternativo – Entrega:

No passo 3 do cenário principal, o Líder Técnico deve vincular as entregas que serão atendidas com a conclusão da tarefa e caso a entrega não esteja cadastrada o usuário poderá cadastra-la.

1. Líder Técnico acessa a opção de adicionar entrega;
2. Sistema apresenta o formulário de cadastro de entrega;
3. Líder Técnico informa os campos solicitados no formulário;
4. Líder Técnico salva os dados do formulário;
5. Sistema valida os campos informados;
6. Sistema persiste os dados no banco de dados;
7. Sistema volta para o formulário de cadastro de tarefa preenchendo o campo entrega com a entrega cadastrada.

Cenário alternativo – Editar:

No passo 1 do cenário principal, o Líder Técnico possui a opção de editar uma tarefa.

1. Líder Técnico acessa a opção de editar tarefa;
2. Sistema apresenta o formulário de registro de tarefa com os campos preenchidos;
3. Líder Técnico altera os campos que julgar necessário;
4. Líder Técnico salva as alterações realizadas;
5. Sistema valida os campos informados;
6. Sistema persiste os dados na base de dados;
7. Sistema apresenta a tela de visualização do projeto com foco na guia tarefa.

Cenário alternativo – Apagar:

No passo 1 do cenário principal, o Líder Técnico possui a opção de apagar uma tarefa.

1. Líder Técnico acessa a opção de apagar tarefa;
2. Sistema exclui a tarefa;
3. Sistema apresenta a tela de visualização do projeto com foco na guia tarefa.

Cenário alternativo – Histórico:

No passo 1 cenário principal, o Líder Técnico possui a opção de verificar o histórico de modificações de uma tarefa.

1. Líder Técnico acessa a opção de histórico de modificações da tarefa;
2. Sistema apresenta a relação de modificações da tarefa.

Cenário alternativo – Finalizar:

No passo 1 do cenário principal, o Líder Técnico possui a opção de marcar uma tarefa como concluída.

1. Líder Técnico acessa a opção de finalizar tarefa;
2. Sistema muda o *status* da tarefa para finalizada;
3. Sistema recarrega a tela de visualização de projeto com foco na guia tarefas;
4. Sistema recarrega o gráfico de tarefas.

Cenário de exceção:

No passo 6 do cenário principal, no passo 5 do cenário alternativo responsável, no passo 5 do cenário alterativo entrega e no passo 5 do cenário alternativo editar, caso os campos obrigatórios não forem preenchidos corretamente, o sistema apresentará uma mensagem informando que há erros nos dados informados que precisam ser corrigidos para prosseguir. Informará também quais os campos obrigatórios que não foram informados.

No Quadro 14 tem-se o detalhamento do caso de uso Cadastrar Recurso.

Quadro 14 - Caso de uso Cadastrar Recurso

Caso de uso: UC05 – Cadastrar Recurso

Permitir que o Líder Técnico cadastre um novo recurso.

Ator: Líder Técnico.

Pré-condição: Estar logado no sistema e estar visualizando um projeto cadastrado.

Pós-condição: O recurso é cadastrado e a tela de visualização de projeto é apresentada, com foco na guia recursos.

Cenário principal:

1. Líder Técnico acessa a guia recursos;

2. Líder Técnico acessa a opção de adicionar recurso;
3. Sistema apresenta o formulário de cadastro de recurso;
4. Líder Técnico informa os campos solicitados no formulário;
5. Líder Técnico salva os dados do formulário;
6. Sistema valida os campos informados;
7. Sistema persiste os dados na base de dados;
8. Sistema apresenta a tela de visualização do projeto com foco na guia recursos.

Cenário alternativo – Editar:

No passo 1 do cenário principal, o Líder Técnico possui a opção de editar um recurso.

1. Líder Técnico acessa a opção de editar recurso;
2. Sistema apresenta o formulário de cadastro recurso com os campos preenchidos;
3. Líder Técnico altera os campos que julgar necessário;
4. Líder Técnico salva as alterações realizadas;
5. Sistema valida os campos informados;
6. Sistema persiste os dados na base de dados;
7. Sistema apresenta a tela de visualização do projeto com foco na guia recursos.

Cenário alternativo – Apagar:

No passo 1 do cenário principal, o Líder Técnico possui a opção de apagar um recurso.

1. Líder Técnico acessa a opção de apagar recurso;
2. Sistema exclui a recurso;
3. Sistema apresenta a tela de visualização do projeto com foco na guia recurso.

Cenário alternativo – Histórico:

No passo 1 do cenário principal, o Líder Técnico possui a opção de verificar o histórico de modificações de uma entrega.

1. Líder Técnico acessa a opção de histórico de modificações do recurso;
2. Sistema apresenta a relação de modificações da recurso.

Cenário de exceção:

No passo 6 do cenário principal e no passo 5 do cenário alternativo editar, caso os campos obrigatórios não forem preenchidos corretamente, o sistema apresentará uma mensagem

informando que há erros nos dados informados que precisam ser corrigidos para prosseguir. Informará também quais os campos obrigatórios que não foram informados.

No Quadro 15 tem-se o detalhamento do caso de uso Cadastrar Risco.

Quadro 15 - Caso de uso Cadastrar Risco

Caso de uso: UC06 – Cadastrar Risco

Permitir que a Equipe de Trabalho cadastre um novo risco.

Ator: Equipe de Trabalho

Pré-condição: Estar logado no sistema e estar visualizando um projeto cadastrado.

Pós-condição: O risco é cadastrado e a tela de visualização de projeto é apresentada, com foco na guia riscos.

Cenário principal:

1. Equipe de Trabalho acessa a guia riscos;
2. Equipe de Trabalho acessa a opção de adicionar risco;
3. Sistema apresenta o formulário de cadastro de risco;
4. Equipe de Trabalho informa os campos solicitados no formulário;
5. Equipe de Trabalho salva os dados do formulário;
6. Sistema valida os campos informados;
7. Sistema persiste os dados na base de dados;
8. Sistema apresenta a tela de visualização do projeto com foco na guia risco.

Cenário alternativo – Editar:

No passo 1 do cenário principal a Equipe de Trabalho técnico possui a opção de editar um risco.

1. Equipe de Trabalho acessa a opção de editar risco;
2. Sistema apresenta o formulário de cadastro de risco com os campos preenchidos;
3. Equipe de Trabalho altera os campos que julgar necessário;
4. Equipe de Trabalho salva as alterações realizadas;
5. Sistema valida os campos informados;
6. Sistema persiste os dados na base de dados;
7. Sistema apresenta a tela de visualização do projeto com foco na guia riscos.

Cenário alternativo – Apagar:

No passo 1 do cenário principal, a Equipe de Trabalho possui a opção de apagar um risco.

1. Equipe de Trabalho acessa a opção de apagar risco;
2. Sistema exclui a risco;
3. Sistema apresenta a tela de visualização do projeto com foco na guia riscos.

Cenário alternativo – Histórico:

No passo 1 do cenário principal, a Equipe de Trabalho possui a opção de verificar o histórico de modificações de um risco.

1. Equipe de Trabalho acessa a opção de histórico de modificações do risco;
2. Sistema apresenta a relação de modificações da risco.

Cenário de exceção:

No passo 6 do cenário principal e no passo 5 do cenário alternativo editar, caso os campos obrigatórios não forem preenchidos corretamente, o sistema apresentará uma mensagem informando que há erros nos dados informados que precisam ser corrigidos para prosseguir. Informará também quais os campos obrigatórios que não foram informados.

No Quadro 16 tem-se o detalhamento do caso de uso Registrar Reunião.

Quadro 16 - Caso de uso Registrar Reunião

Caso de uso: UC07 – Registrar Reunião

Permitir que o Líder Técnico registre uma nova reunião.

Ator: Líder Técnico.

Pré-condição: Estar logado no sistema e estar visualizando um projeto reunião.

Pós-condição: A reunião é registrada e a tela de visualização de projeto é apresentada, com foco na guia reuniões.

Cenário principal:

1. Líder Técnico acessa a guia reuniões;
2. Líder Técnico acessa a opção de adicionar reunião;
3. Sistema apresenta o formulário de registro de reunião;
4. Líder Técnico informa os campos solicitados no formulário;

5. Líder Técnico salva os dados do formulário;
6. Sistema valida os campos informados;
7. Sistema persiste os dados na base de dados;
8. Sistema apresenta a tela de visualização do projeto com foco na guia reuniões.

Cenário alternativo – Editar:

No passo 1 do cenário principal, o Líder Técnico possui a opção de editar uma reunião.

1. Líder Técnico acessa a opção de editar reunião;
2. Sistema apresenta o formulário de registro de reunião com os campos preenchidos;
3. Líder Técnico altera os campos que julgar necessário;
4. Líder Técnico salva as alterações realizadas;
5. Sistema valida os campos informados;
6. Sistema persiste os dados na base de dados;
7. Sistema apresenta a tela de visualização do projeto com foco na guia reuniões.

Cenário alternativo – Apagar:

No passo 1 do cenário principal, o Líder Técnico possui a opção de apagar uma reunião.

1. Líder Técnico acessa a opção de apagar reunião;
2. Sistema exclui a reunião;
3. Sistema apresenta a tela de visualização do projeto com foco na guia reuniões.

Cenário alternativo – Histórico:

No passo 1 do cenário principal, o Líder Técnico possui a opção de verificar o histórico de modificações de uma reunião.

1. Líder Técnico acessa a opção de histórico de modificações da reunião;
2. Sistema apresenta a relação de modificações da reunião.

Cenário de exceção:

No passo 6 do cenário principal e no passo 5 do cenário alternativo editar, caso os campos obrigatórios não forem preenchidos corretamente, o sistema apresentará uma mensagem informando que há erros nos dados informados que precisam ser corrigidos para prosseguir. Informará também quais os campos obrigatórios que não foram informados.

Quadro 17 - Caso de uso Registrar Mudança

Caso de uso: UC08 – Registrar Mudança.

Permitir que o Líder Técnico registre uma nova mudança.

Ator: Líder Técnico.

Pré-condição: Estar logado no sistema e estar visualizando um projeto cadastrado.

Pós-condição: A mudança é registrada e a tela de visualização de projeto é apresentada, com foco na guia mudanças.

Cenário principal:

1. Líder Técnico acessa a guia mudanças;
2. Líder Técnico acessa a opção de adicionar mudança;
3. Sistema apresenta o formulário de registro de mudança;
4. Líder Técnico informa os campos solicitados no formulário;
5. Líder Técnico salva os dados do formulário;
6. Sistema valida os campos informados;
7. Sistema persiste os dados na base de dados;
8. Sistema apresenta a tela de visualização do projeto com foco na guia mudanças.

Cenário alternativo – Editar:

No passo 1 do cenário principal, o Líder Técnico possui a opção de editar uma mudança.

1. Líder Técnico acessa a opção de editar mudança;
2. Sistema apresenta o formulário de registro de mudança com os campos preenchidos;
3. Líder Técnico altera os campos que julgar necessário;
4. Líder Técnico salva as alterações realizadas;
5. Sistema valida os campos informados;
6. Sistema persiste os dados na base de dados;
7. Sistema apresenta a tela de visualização do projeto com foco na guia mudanças.

Cenário alternativo – Apagar:

No passo 1 do cenário principal, o Líder Técnico possui a opção de apagar uma mudança.

1. Líder Técnico acessa a opção de apagar mudança;
2. Sistema exclui a mudança;

3. Sistema apresenta a tela de visualização do projeto com foco na guia mudanças.

Cenário alternativo – Histórico:

No passo 1 do cenário principal, o Líder Técnico possui a opção de verificar o histórico de modificações de uma mudança.

1. Líder Técnico acessa a opção de histórico de modificações da mudança;
2. Sistema apresenta a relação de modificações da mudança.

Cenário de exceção:

No passo 6 do cenário principal e no passo 5 do cenário alternativo editar, caso os campos obrigatórios não forem preenchidos corretamente, o sistema apresentará uma mensagem informando que há erros nos dados informados que precisam ser corrigidos para prosseguir. Informará também quais os campos obrigatórios que não foram informados.

No Quadro 18 tem-se o detalhamento do caso de uso Cadastrar Correção.

Quadro 18 - Caso de uso Cadastrar Correção

Caso de uso: UC09 – Cadastrar Correção

Permitir que a Equipe de Trabalho cadastre uma nova correção.

Ator: Equipe de Trabalho.

Pré-condição: Estar logado no sistema e estar visualizando um projeto cadastrado.

Pós-condição: A correção é cadastrada e a tela de visualização de projeto é apresentada, com foco na guia correções.

Cenário principal:

1. Equipe de Trabalho acessa a guia correções;
2. Equipe de Trabalho acessa a opção de adicionar correção;
3. Sistema apresenta o formulário de cadastro de correção;
4. Equipe de Trabalho informa os campos solicitados no formulário;
5. Equipe de Trabalho salva os dados do formulário;
6. Sistema valida os campos informados;
7. Sistema persiste os dados na base de dados;
8. Sistema apresenta a tela de visualização do projeto com foco na guia correções.

Cenário alternativo – Editar:

No passo 1 do cenário principal, a Equipe de Trabalho possui a opção de editar uma correção.

1. Equipe de Trabalho acessa a opção de editar correção;
2. Sistema apresenta o formulário de cadastro de correção com os campos preenchidos;
3. Equipe de Trabalho altera os campos que julgar necessário;
4. Equipe de Trabalho salva as alterações realizadas;
5. Sistema valida os campos informados;
6. Sistema persiste os dados na base de dados;
7. Sistema apresenta a tela de visualização do projeto com foco na guia correções.

Cenário alternativo – Apagar:

No passo 1 do cenário principal, a Equipe de Trabalho possui a opção de apagar uma correção.

1. Equipe de Trabalho a opção de apagar correção;
2. Sistema exclui a correção;
3. Sistema apresenta a tela de visualização do projeto com foco na guia correções.

Cenário alternativo – Histórico:

No passo 1 do cenário principal, a Equipe de Trabalho possui a opção de verificar o histórico de modificações de uma correção.

1. Equipe de Trabalho acessa a opção de histórico de modificações do correção;
2. Sistema apresenta a relação de modificações da correção.

Cenário de exceção:

No passo 6 do cenário principal e no passo 5 do cenário alternativo editar, caso os campos obrigatórios não forem preenchidos corretamente, o sistema apresentará uma mensagem informando que há erros nos dados informados que precisam ser corrigidos para prosseguir. Informará também quais os campos obrigatórios que não foram informados.

No Quadro 19 tem-se o detalhamento do caso de uso Incluir Documento.

Quadro 19 - Caso de uso Incluir Documento

Caso de uso: UC10 – Incluir Documento

Permitir que o Usuário inclua um novo documento.

Ator: Usuário.

Pré-condição: Estar visualizando um projeto cadastrado.

Pós-condição: O documento é incluído e a tela de visualização de projeto é apresentada, com foco na guia anexos.

Cenário principal:

1. Usuário acessa a guia anexos;
2. Usuário acessa a opção de adicionar anexo;
3. Sistema apresenta o formulário de inclusão de documento;
4. Usuário informa os campos solicitados no formulário;
5. Usuário salva os dados do formulário;
6. Sistema valida os campos informados;
7. Sistema persiste os dados na base de dados;
8. Sistema apresenta a tela de visualização do projeto com foco na guia anexos.

Cenário alternativo – Editar:

No passo 1 do cenário principal, o Usuário possui a opção de editar um anexo.

1. Usuário acessa a opção de editar anexo;
2. Sistema apresenta o formulário de anexo com os campos preenchidos;
3. Usuário altera os campos que julgar necessário;
4. Usuário salva as alterações realizadas;
5. Sistema valida os campos informados;
6. Sistema persiste os dados na base de dados;
7. Sistema apresenta a tela de visualização do projeto com foco na guia anexos.

Cenário alternativo – Remover:

No passo 1 do cenário principal, o Usuário possui a opção de remover um anexo.

1. Usuário acessa a opção de remover anexo;
2. Sistema exclui a correção;

3. Sistema apresenta a tela de visualização do projeto com foco na guia correções.

Cenário alternativo – Histórico:

No passo 1 do cenário principal, o Usuário possui a opção de verificar o histórico de modificações de um anexo.

1. Usuário acessa a opção de histórico de modificações do anexo;
2. Sistema apresenta a relação de modificações do anexo.

Cenário de exceção:

No passo 6 do cenário principal e no passo 5 do cenário alternativo editar, caso os campos obrigatórios não forem preenchidos corretamente, o sistema apresentará uma mensagem informando que há erros nos dados informados que precisam ser corrigidos para prosseguir. Informará também quais os campos obrigatórios que não foram informados.

APÊNDICE B – Descrição do Dicionário de Dados

Neste Apêndice apresenta-se o dicionário de dados das principais entidades utilizadas no sistema, com informações de atributos, tipos, chaves primárias (PK) e chaves estrangeiras (FK), campos obrigatórios e descrições. Foram utilizados nove tipos de atributos:

- a) *boolean*: para atributos de verdadeiro e falso;
- b) *date*: para atributos de data;
- c) *datetime*: para atributos de data e hora;
- d) *float*: para atributos de pontos flutuantes (campos decimais);
- e) *integer*: para atributos numéricos;
- f) *smalint*: para atributos numéricos;
- g) *text*: para atributos de texto (sem tamanho máximo);
- h) *time*: para atributos de hora;
- i) *timestamp*: para atributos de date e hora local;
- j) *varchar*: para atributos de texto (com tamanho máximo).

No Quadro 20 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo registro de usuários, sendo que a mesma é criada pelo próprio *framework* Django.

Quadro 20 - Dicionário de dados da entidade *auth_user*

Entidade: <i>auth_user</i>					
Descrição: Persiste os dados de usuários.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>first_name</i>	<i>varchar(30)</i>	Não	Não	Sim	Primeiro nome.
<i>last_name</i>	<i>varchar(30)</i>	Não	Não	Sim	Último nome (Sobrenome).
<i>email</i>	<i>varchar(75)</i>	Não	Não	Sim	Endereço de correio eletrônico.
<i>username</i>	<i>varchar(30)</i>	Não	Não	Sim	Usuário.
<i>pssword</i>	<i>varchar(128)</i>	Não	Não	Sim	Senha.
<i>last_login</i>	<i>timestamp</i>	Não	Não	Não	Data/hora do último acesso.
<i>date</i>	<i>timestamp</i>	Não	Não	Não	Data/hora de criação.
<i>is_staff</i>	<i>boolean</i>	Não	Não	Não	Membro da equipe administrativa.

<i>is_superuser</i>	<i>boolean</i>	Não	Não	Não	Super usuário.
<i>is_active</i>	<i>boolean</i>	Não	Não	Não	Ativo.

No Quadro 21 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo cadastro de projetos.

Quadro 21 - Dicionário de dados da entidade project

Entidade: project					
Descrição: Persiste os dados do projeto.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>date_open</i>	<i>datetime</i>	Não	Não	Sim	Data de criação.
<i>date_end</i>	<i>datetime</i>	Não	Não	Sim	Data de finalização.
<i>resp_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do usuário.
<i>name</i>	<i>varchar(500)</i>	Não	Não	Sim	Título.
<i>purpose</i>	<i>text</i>	Não	Não	Sim	Propósito.
<i>geral_requeriment</i>	<i>text</i>	Não	Não	Sim	Requisitos gerais.
<i>scope</i>	<i>text</i>	Não	Não	Sim	Escopo.
<i>description</i>	<i>text</i>	Não	Não	Sim	Descrição.
<i>complete</i>	<i>boolean</i>	Não	Não	Sim	Completo.

No Quadro 22 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo cadastro da equipe de trabalho.

Quadro 22 - Dicionário de dados da entidade project_workteam

Entidade: project_workteam					
Descrição: Persiste os dados da equipe de trabalho.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>project_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do projeto.
<i>user_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do usuário.
<i>paper</i>	<i>integer</i>	Não	Não	Sim	Papel.

No Quadro 23 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo cadastro de entregas.

Quadro 23 - Dicionário de dados da entidade project_delivery

Entidade: project_delivery					
Descrição: Persiste os dados das entregas.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>project_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do projeto.
<i>name</i>	<i>varchar(500)</i>	Não	Não	Sim	Título.
<i>description</i>	<i>text</i>	Não	Não	Sim	Descrição.
<i>date_end</i>	<i>date</i>	Não	Não	Sim	Data estimada de conclusão.

No Quadro 24 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo cadastro de atividades.

Quadro 24 - Dicionário de dados da entidade project_activity

Entidade: project_activity					
Descrição: Persiste os dados das tarefas.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>project_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do projeto.
<i>name</i>	<i>varchar(500)</i>	Não	Não	Sim	Título
<i>description</i>	<i>text</i>	Não	Não	Sim	Descrição.
<i>user_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do usuário.
<i>time_duration</i>	<i>time</i>	Não	Não	Sim	Tempo estimado de duração.
<i>date_start</i>	<i>date</i>	Não	Não	Sim	Data estimada de início.
<i>date_end</i>	<i>date</i>	Não	Não	Sim	Data estimada de conclusão.
<i>complete</i>	<i>boolean</i>	Não	Não	Sim	Completo.

No Quadro 25 tem-se os campos utilizados para a ligação entre as entregas e as atividades.

Quadro 25 - Dicionário de dados da entidade project_activity_delivery

Entidade: project_activity_delivery					
Descrição: Persiste os dados de ligação entre tarefas e entregas.					
Atributo	Tipo	PK	FK	Obrig.	Descrição

<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>activity_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação da tarefa.
<i>delivery_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação da entrega.

No Quadro 26 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo cadastro de recursos.

Quadro 26 - Dicionário de dados da entidade *project_resource*

Entidade: <i>project_resource</i>					
Descrição: Persiste os dados dos recursos.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>project_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do projeto.
<i>type</i>	<i>integer</i>	Não	Não	Sim	Tipo.
<i>name</i>	<i>varchar(500)</i>	Não	Não	Sim	Título.
<i>description</i>	<i>text</i>	Não	Não	Sim	Descrição.

No Quadro 27 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo cadastro de solicitações de mudança.

Quadro 27 - Dicionário de dados da entidade *project_changerequest*

Entidade: <i>project_changerequest</i>					
Descrição: Persiste os dados das solicitações de mudança.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Sim	Sim	Identificação única.
<i>project_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do projeto.
<i>name</i>	<i>varchar(500)</i>	Não	Não	Sim	Título.
<i>decription</i>	<i>text</i>	Não	Não	Sim	Descrição.
<i>status</i>	<i>integer</i>	Não	Não	Sim	Situação.
<i>level</i>	<i>integer</i>	Não	Não	Sim	Nível.
<i>requested_by</i>	<i>integer</i>	Não	Não	Sim	Solicitante.
<i>date_necessary</i>	<i>date</i>	Não	Não	Sim	Necessário para.
<i>impact</i>	<i>text</i>	Não	Não	Sim	Impacto.

No Quadro 28 tem-se os campos utilizados para a ligação entre as mudanças e as

entregas que serão impactadas.

Quadro 28 - Dicionário de dados da entidade project_changerequest_delivery

Entidade: project_changerequest_delivery					
Descrição: Persiste os dados das entregas impactadas com a mudança.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>changerequest_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação da mudança.
<i>delivery_ud</i>	<i>integer</i>	Não	Sim	Sim	Identificação da entrega.

No Quadro 29 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo cadastro de correções.

Quadro 29 - Dicionário de dados da entidade project_correct

Entidade: project_correct					
Descrição: Persiste os dados das correções.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>project_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do projeto.
<i>name</i>	<i>varchar(500)</i>	Não	Não	Sim	Título.
<i>description</i>	<i>text</i>	Não	Não	Sim	Descrição.
<i>solution</i>	<i>text</i>	Não	Não	Sim	Solução.
<i>user_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do usuário.
<i>date_open</i>	<i>date</i>	Não	Não	Sim	Data de abertura.
<i>date_end</i>	<i>date</i>	Não	Não	Sim	Data de previsão para entrega.
<i>action</i>	<i>varchar(500)</i>	Não	Não	Sim	Ação corretiva.

No Quadro 30 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo cadastro de riscos.

Quadro 30 - Dicionário de dados da entidade project_risk

Entidade: project_risk					
Descrição: Persiste os dados das riscos.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.

<i>project_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do projeto.
<i>name</i>	<i>varchar(500)</i>	Não	Não	Sim	Título.
<i>description</i>	<i>text</i>	Não	Não	Sim	Descrição.

No Quadro 31 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo cadastro de reuniões.

Quadro 31 - Dicionário de dados da entidade *project_meeting*

Entidade: <i>project_meeting</i>					
Descrição: Persiste os dados das reuniões.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>project_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do projeto.
<i>name</i>	<i>varchar(500)</i>	Não	Não	Sim	Título.
<i>description</i>	<i>text</i>	Não	Não	Sim	Descrição.
<i>local</i>	<i>text</i>	Não	Não	Sim	Local.
<i>date</i>	<i>date</i>	Não	Não	Sim	Data agendada.
<i>lift</i>	<i>text</i>	Não	Não	Sim	Levantamentos.
<i>pendency</i>	<i>text</i>	Não	Não	Sim	Pendências.
<i>agreement</i>	<i>text</i>	Não	Não	Sim	Acordos.
<i>ata</i>	<i>varchar(100)</i>	Não	Não	Sim	Arquivo da ata.

No Quadro 32 tem-se os campos utilizados para a ligação entre a reunião e os usuários convocados.

Quadro 32 - Dicionário de dados da entidade *project_meeting_team*

Entidade: <i>project_meeting_team</i>					
Descrição: Persiste os dados dos usuários convocados para a reunião.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>meeting_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação da reunião.
<i>user_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do usuário.

No Quadro 33 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo cadastro de anexos.

Quadro 33 - Dicionário de dados da entidade project_file

Entidade: project_file					
Descrição: Persiste os dados dos arquivos anexos.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>project_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do projeto.
<i>user_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do usuário.
<i>date</i>	<i>timestamp</i>	Não	Não	Sim	Data de anexo.
<i>name</i>	<i>varchar(500)</i>	Não	Não	Sim	Título do arquivo.
<i>file</i>	<i>varchar(100)</i>	Não	Não	Sim	Caminho do arquivo anexado.

No Quadro 34 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo relatório de verificação.

Quadro 34 - Dicionário de dados da entidade project_reportcheck

Entidade: project_reportcheck					
Descrição: Persiste os dados do relatório de verificação.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>project_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do projeto.
<i>date</i>	<i>date</i>	Não	Não	Sim	Data de cadastro.
<i>time_duration</i>	<i>time</i>	Não	Não	Sim	Tempo de duração.
<i>local</i>	<i>varchar(200)</i>	Não	Não	Sim	Local de realização do relatório.
<i>delivery_viable</i>	<i>boolean</i>	Não	Não	Não	Entregas são viáveis.
<i>delivery_solid</i>	<i>boolean</i>	Não	Não	Não	Entregas são consistentes.
<i>resource_vialble</i>	<i>boolean</i>	Não	Não	Não	Recursos são viáveis.
<i>resource_solid</i>	<i>boolean</i>	Não	Não	Não	Recursos são consistentes.
<i>activity_viable</i>	<i>boolean</i>	Não	Não	Não	Tarefas são viáveis.
<i>activity_solid</i>	<i>boolean</i>	Não	Não	Não	Tarefas são consistentes.
<i>changerequest_viable</i>	<i>boolean</i>	Não	Não	Não	Mudanças são viáveis.
<i>changerequest_solid</i>	<i>boolean</i>	Não	Não	Não	Mudanças são consistentes.
<i>correct_viable</i>	<i>boolean</i>	Não	Não	Não	Correções são viáveis.
<i>correct_solid</i>	<i>boolean</i>	Não	Não	Não	Correções são consistentes.
<i>Defect.</i>	<i>text</i>	Não	Não	Não	Defeitos.

No Quadro 35 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo *backup*.

Quadro 35 - Dicionário de dados da entidade *project_backup*

Entidade: <i>django_admin_log</i>					
Descrição: Persiste os dados do histórico de alteração.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>project_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do projeto.
<i>user_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do usuário.
<i>file</i>	<i>varchar(100)</i>	Não	Não	Sim	Caminho do arquivo de <i>backup</i> .

No Quadro 36 tem-se os campos utilizados para a persistência dos dados armazenados no sistema responsável pelo histórico de modificações. Esta entidade também é gerada e persistida pelo *framework* Django.

Quadro 36 - Dicionário de dados da entidade *django_admin_log*

Entidade: <i>django_admin_log</i>					
Descrição: Persiste os dados do histórico de alteração.					
Atributo	Tipo	PK	FK	Obrig.	Descrição
<i>id</i>	<i>integer</i>	Sim	Não	Sim	Identificação única.
<i>action_time</i>	<i>timestamp</i>	Não	Não	Sim	Data/Hora da alteração.
<i>user_id</i>	<i>integer</i>	Não	Sim	Sim	Identificação do usuário.
<i>content_type_id</i>	<i>integer</i>	Não	Sim	Não	Identificação do modelo.
<i>object_id</i>	<i>text</i>	Não	Não	Não	Identificação do registro.
<i>object_repr</i>	<i>varchar(200)</i>	Não	Não	Sim	Representação do registro.
<i>action_flag</i>	<i>smallint</i>	Não	Não	Sim	Ação realizada.
<i>change_message</i>	<i>text</i>	Não	Não	Sim	Texto da alteração realizada.

As demais entidades demonstradas na seção 3.2.3 deste documento são criadas pelo *framework* Django e não são utilizadas para persistência dos dados gerados pelo sistema desenvolvido. Estas entidades são criadas para controle de permissão de usuário, grupos de usuário e sessões do *framework*.